

Diplomarbeit
zur Erlangung des Grades eines
Diplom-Informatikers
im
Studiengang Informatik
an der
Hochschule Bremerhaven

Thema der Diplomarbeit:

Konzipierung und Implementierung eines Programms zur
Übernahme und Extrahierung von unterschiedlich
strukturierten Forschungsdaten in eine
Forschungsdatenbank

vorgelegt am 09.05.2005 von

Valerij Cysko

aus Schiffdorf

Referent:
Korreferent:

Prof. Dr. Günter Matthiessen
Prof. Dr. Thomas Umland

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Abbildungsverzeichnis	3
Tabellenverzeichnis	4
Diagrammverzeichnis	5
1 Einleitung.....	6
2 Aufgabenstellung	8
3 Aufgabenanalyse.....	8
3.1 Messverfahren, Datenerhebungsart, Datenbestand	8
3.1.1 Messverfahren „Ground thickness profiles“	8
3.1.2 Messverfahren „Freeboard profiles“	15
3.1.3 Messverfahren „HEM Thickness Profiles“	18
3.2 Anforderungen an die Daten für die Archivierung	20
3.2.1 Anforderungen an Event labels	21
3.2.2 Struktur von zu erzeugenden Metadatendateien im Projektbereich „PANGAEA“	21
3.2.3 Struktur von zu erzeugenden Datendateien im Projektbereich „PANGAEA“	23
3.2.4 Anforderungen an die Daten für die Archivierung in der Datenbank „SeaIceDB“	28
4 Die Entwicklung	29
4.1 Pflichtenheft SI_Thickness_DB.....	29
4.1.1 Zielbestimmung	29
4.1.2 Einsatz	29
4.1.3 Umgebung.....	29
4.1.4 Funktionalität	29
4.1.5 Daten.....	30
4.1.6 Leistungen.....	30
4.1.7 Benutzungsoberfläche.....	30
4.1.8 Qualitätsziele	30
4.2 UML-Diagramme	31
4.3 Das Datenmodell	31
4.3.1 Tabelle “scientists”	31
4.3.2 Tabelle “parameters”	31
4.3.3 Tabelle “methods”	31
4.3.4 Tabelle “measurementPositions”	31
4.3.5 Tabelle “campaigns”	32
4.3.6 Tabelle “iceThicknessMetaData”	32
4.3.7 Tabelle “iceThicknessData”	33
4.3.8 Tabelle “laserMetaData”	35
4.3.9 Tabelle “laserData”	36
4.3.10 Tabelle “iceThickLaserMetaData”	37
4.3.11 Tabelle “iceThickLaserData”	37
4.3.12 ER-Diagramm.....	39
4.3.13 Relationales Datenmodell	40
4.4 Das Klassendesign	41
4.4.1 Paket FilesProcessing	41
4.4.2 Paket Importing	48
4.4.3 Paket Presenting	50
5 Entwicklungsumgebung	53
5.1 Software.....	53
5.2 Hardware	53
5 Zusammenfassung	54
Anhang A Metadatendatei „la072508.42“ aus dem Projekt „Freeboard profiles“	LV
Anhang B Use Case Diagramme	II
Anhang C Aktivitätsdiagramme	IV
Anhang D Klassendiagramme	XV
Anhang E Sequenzdiagramme.....	XXXIX
Anhang F Paketdiagramm	XLI
Anhang G Verteilungsdiagramm	XLII
Anhang H SQL-Skript	XLIII

Abbildungsverzeichnis

Abbildung 1: Die Eisschollen mit ihren Parametern	9
Abbildung 2: Die Messung mit dem Sensor „Geonics EM31“	10
Abbildung 3: Komponenten der Messmethode “Electromagnetic sounding (EM), GeonicsEM31 conductivity sensor, Haas et al 1997”	11
Abbildung 4: Lasereinheit der Messsonde „HEM Bird“	15
Abbildung 5: Die Messsonde „HEM Bird“ von innen.....	15
Abbildung 6: HEM Bird vor dem Einsatz	16
Abbildung 7: Schematische Darstellung aller am Messverfahren beteiligten Komponenten	16
Abbildung 8: Die Messung in Aktion.....	17
Abbildung 9: Ausschnitt aus der Internetseite von PANGAEA	24
Abbildung 10: Ausschnitt aus der Datenbeschreibung für den Event label Ark12_222p1 (1)	25
Abbildung 11: Ausschnitt aus der Datenbeschreibung für den Event label Ark12_222p1 (2)	26
Abbildung 12: Startfenster des Programms	42
Abbildung 13: Dialog für die Parameterauswahl in dem Projektbereich „PANGAEA”	43
Abbildung 14: Dialog für die Parameterauswahl in dem Projektbereich „SeaIceDB”	44
Abbildung 15: Dialog für die Attribute von der ausgewählten Parameter in dem Projektbereich „SeaIceDB”	46
Abbildung 16: Dialog für die Attribute von der ausgewählten Parameter in dem Projektbereich „PANGAEA”	46
Abbildung 17: Dialog für das Erfassen des Wissenschaftlers oder der Messmethode in der Datenbank	49
Abbildung 18: Dialog für das Erfassen des Parameters in der Datenbank	49
Abbildung 19: Dialog für das Projekt-und Expeditionauswahl	50
Abbildung 20: Dialogfenster für die Auswahl der Metadatenätzen von der ausgewählte Expedition	51
Abbildung 21: Fenster für die Darstellung der Daten von den ausgewählten Metadatenätzen.....	52

Tabellenverzeichnis

Tabelle 1: Die Parameter und Messmethoden in dem Projekt „Ground thickness profiles“	10
Tabelle 2: Format der Quellenmetadatendatei in dem Projekt „Ground thickness profiles“	12
Tabelle 3: Ausschnitt aus der Datendatei von der Expedition ARK-VIII/3 (Transpolar Drift, 02.08.1991 – 08.10.1991)	12
Tabelle 4: Ausschnitt aus der Datendatei von der Expedition ARK-IX/4 (Laptev-See, 06.08.1993 – 05.10.1993).....	12
Tabelle 5: Ausschnitt aus der Datendatei von der Expedition ARK-XI/1 (Laptev-See, 07.07.1995 – 20.09.1995).....	13
Tabelle 6: Ausschnitt aus der Datendatei der Expedition ARK-XVII/2 (Reisegebiet: Arktische Ozean, die Zeitraum: 31.07.2001 – 07.10.2001)	13
Tabelle 7: Ausschnitt aus der Datendatei „07250842a.dat,, von der Expedition ARK-XII.....	17
Tabelle 8: Ausschnitt aus der Datei „200302170937_allfinal.dat“ von der Expedition IRIS2003.....	19
Tabelle 9: Ausschnitt aus Datendatei „200402051152_allfinal.dat“ von der Expedition IRIS2004	19
Tabelle 10: Struktur der zu erzeugende Metadatendatei (Projekt: „Ground thickness profiles“, Projektbereich „PANGAEA“)	21
Tabelle 11: Struktur der zu erzeugenden Metadatendatei (Projekte: „Freeboard profiles“, „HEM thickness profiles“, Projektbereich „PANGAEA“)	22
Tabelle 12: Datenbanktabelle „scientists“ mit dazugehörigen Attribute	31
Tabelle 13: Datenbanktabelle „parameters“ mit dazugehörigen Attributen	31

Diagrammverzeichnis

Diagramm 1: ER-Diagramm..... 39
Diagramm 2: Relationales Datenmodell 40

1 Einleitung

Die Diplomarbeit wurde an dem Alfred-Wegener-Institut für Polar- und Meeresforschung in Bremerhaven geschrieben. Das Institut wurde 1980 in Bremerhaven als Stiftung des öffentlichen Rechts gegründet. Die Stiftung Alfred-Wegener-Institut für Polar- und Meeresforschung umfasst das Alfred-Wegener-Institut für Polar- und Meeresforschung in Bremerhaven, die Forschungsstelle Potsdam (seit 1992), die Biologische Anstalt Helgoland und die Wattenmeerstation Sylt.

Sie ist das Zentrum der deutschen Polar- und Meeresforschung und wurde nach dem deutschen Polarforscher und Geowissenschaftler Alfred Wegener benannt. Sie ist Mitglied der Helmholtz-Gemeinschaft, die mit ihren 15 Forschungszentren und einem Jahresbudget von rund 2,1 Milliarden Euro die größte Wissenschaftsorganisation Deutschlands ist.

Die Stiftung Alfred-Wegener-Institut führt wissenschaftliche Projekte in der Arktis, Antarktis und den gemäßigten Breiten durch und koordiniert die Polarforschung in Deutschland.

Ziel der wissenschaftlichen Arbeit des AWI ist ein besseres Verständnis der Beziehungen zwischen Ozean, Eis und Atmosphäre, der Tier- und Pflanzenwelt der Arktis und Antarktis sowie der Entwicklungsgeschichte der polaren Kontinente und Meere.

Das AWI arbeitet in zahlreichen internationalen Forschungsprogrammen und steht in engem Kontakt mit zahlreichen Universitäten und Institutionen in Europa und Übersee. Es entsendet Wissenschaftler an Institute in aller Welt, auf Forschungsschiffe und Stationen und lädt die Wissenschaftler anderer Nationen auf das Forschungsschiff „Polarstern“ ein.

Die Stiftung Alfred-Wegener-Institut für Polar- und Meeresforschung umfasst vier Fachbereiche, diese sind:

- Klimasystem,
- Pelagische Ökosysteme,
- Benthische Ökosysteme,
- Geosystem.

Der Schwerpunkt im Fachbereich Klimasystem liegt in der Untersuchung des gekoppelten Systems Ozean-Eis-Atmosphäre und dessen Einfluss auf das globale Klima. Wissenschaftler führen Feldexperimente und numerische Modellstudien durch, um die ozeanische Zirkulation, den Transport von Energie und die Substanzen im polaren Ozean und der polaren Atmosphäre und den Einfluss dieser Prozesse auf das globale Klimageschehen zu untersuchen.

Während des Praxissemesters wurde ich in der Arbeitsgruppe Meereseis (Sektion Regionale Zirkulation, Fachbereich Klimasystem) eingesetzt. Das Forschungsthema dieser Arbeitsgruppe, deren Leiter Herr Dr. Christian Haas ist, heisst „Meereseis-Prozesse“. In Rahmen der Tätigkeit der Gruppe werden die systematische Eisdickenmessungen mittels verschiedenen Verfahren durchgeführt, um Klimaveränderungen zu beobachten. Ich war für die Entwicklung der Programme für die Datenverarbeitung aus diesen Messverfahren zuständig.

Bei allen diesen Verfahren werden die Untersuchungsdaten während der Expeditionen (besser nach der Meinung von Herrn Haas wäre der Begriff „Messkampagne“ - das sind die Forschungsreisen der Wissenschaftlersgruppen mit Schiffen, Flugzeugen, Hubschraubern und Schneemobilen) von den Mitarbeitern der Gruppe „Meereseis“ (größtenteils von dem Herrn Haas selbst) erhoben. Gewonnene Daten müssten noch entsprechend verarbeitet werden, damit sie nach der Auswertung in den Datenbanken von PANGAEA und „SeaIceDB“ archiviert werden könnten. PANGAEA ist ein Netzwerk für die geologischen und Umgebungsdaten des ganzen Instituts. Die im Laufe der Forschungstätigkeit der Wissenschaftler vom AWI entstehenden Daten werden in die Datenbank von PANGAEA eingefügt und stehen allen Interessierenden unter www.pangaea.de zur Verfügung.

„SeaIceDB“ ist die Datenbank auf einem Datenbankserver `edvs0.awi-bremerhaven.de` von AWI (DBMS ASE 12.5), wo ich im Auftrag von Herrn Haas 6 Tabellen (2 für jedes Projekt: 1 für die Metadaten und 1 für die Daten) zur Speicherung des Datenbestands aus den oben genannten Projekten während des Praxissemesters einrichtete.

Dafür entwickelte ich 16 Konsolenprogrammen, die diese Aufgaben bewältigten. Leider ist es mir nicht gelungen, diese Programme zum Ende des Praktikums in einem unter grafischer Oberfläche zu bedienendes Programm zu integrieren. Die entwickelten Programmen erzeugten richtige Ausgaben nur aus den Quellendateien, die für jedes Messverfahren eine bestimmte Spaltenreihenfolge haben mussten. Der Benutzer hatte keine Möglichkeit die Parameter, die Messmethoden oder die Wissenschaftler auszuwählen, weil sie nicht in dem System waren (weder in der Property-Datei noch in der Datenbank gespeichert). Stattdessen wurde der Benutzer bei der Erzeugung der Dateien für die Datenbank von PANGAEA dazu gezwungen, die sogenannte PANGAEA-Identifikationsnummer von diesen Objekten auf der Konsole einzugeben. Die Möglichkeit in der Datenbank „SeaIceDB“ gespeicherte Metadaten und Daten bei dem Bedarf zu extrahieren fehlte ebenfalls.

2 Aufgabenstellung

Aus allen diesen unerfüllten Bedürfnissen entstand der Themenvorschlag für meine Diplomarbeit. Das zu entwickelnde System musste folgende Aufgaben erfüllen:

1. die Dateien von den 3 obengenannten Messverfahren für 2 Bereiche („PANGAEA“ und „SeaIceDB“) ungeachtet auf die Reihenfolge der Parameterspalten erstellt;
2. die Daten aus der für den Bereich „SeaIceDB“ erzeugten Dateien in die Datenbank „SeaIceDB“ einfügt und bei dem Bedarf (Hinzukommen von neuen Parameter oder Messmethoden) die Struktur der Datentabellen, wo diese Daten gespeichert werden, entsprechend erweitert;
3. in der Datenbank „SeaIceDB“ gespeicherte Daten extrahiert und bei dem Bedarf in der Textdateien speichert.

Das zu entwickelnde Programm wäre unter grafischer Benutzeroberfläche (als Empfehlung: in Englisch) zu bedienen und musste sowohl die permanente Speicherung der zur Zeit existierenden Expeditionen, Parameter, Messmethoden und Wissenschaftler in dem System, das dieses Programm und die Datenbanktabellen bilden, gewährleisten, als auch das künftige Hinzufügen von oben genannten Objekten in das System ermöglichen.

3 Aufgabenanalyse

3.1 Messverfahren, Datenerhebungsart, Datenbestand

Abhängig von dem Messverfahren unterscheiden sich die Art der Datenerhebung und die daraus entstehenden Datenbestände.

Gemeinsam für die Datenerhebung bei allen Messverfahren ist die Tatsache, dass während der Untersuchungen die Werten von bestimmten Parametern mit der bestimmten Messmethoden (was die entsprechende Ausrüstung voraussetzt) registriert werden. Alle Messwerte (außer von der Eisbedeckung, die dem Parameter „Ice coverage“ entspricht) haben die gleiche Messeinheit – Meter. Für alle Messverfahren hat Herr Haas kurze Namen vorgeschlagen, die die Verfahren kennzeichnen, und sowohl in dieser Arbeit als auch in dem entwickelten Programm benutzt werden.

3.1.1 Messverfahren „Ground thickness profiles“

Bei diesem Verfahren werden die einzelnen Eisschollen untersucht. Die Messung läuft wie folgt ab. Die Wissenschaftler gelangen auf eine Eisscholle. Als Erstes notieren sie in einem Notizblock den Namen der Eisscholle und mittels GPS-Gerätes bestimmte geografischen Breite und Länge des Orts. Für den Namen der Eisscholle benutzt man den julianischen Tag der Messung. Ist die Eisscholle groß, schaffen die Wissenschaftler nur sie an diesem Tag zu untersuchen, und der Name dieser Eisscholle ist dann z.B. 247. Sind die Eisschollen klein, so dass die Wissenschaftler mehrere Eisschollen am selben Tag untersuchen, lauten deren Namen entsprechend 247.1, 247.2, 247.3 usw. (abhängig von der Anzahl der am gleichen Tag untersuchten Eisschollen). Diese 3 Angaben zu der Eisscholle stellen ihre Metadaten dar.

Danach erhebt man die Messdaten von der Eisscholle. Man wählt den räumlichen Startpunkt der Messung und rollt das Messband auf, um die Entfernung von jeden Messpunkt von dem Startpunkt (dieser Parameter heißt „Measurement point distance“) auf dieser Eisscholle zu ermitteln. In regelmäßigen Abständen (manchmal sind es jede 0.5, 1, 2, 5 oder 20 Meter) erhebt man die Nutzdaten. Dazu misst man die Werte der Parameter, die die Charakteristiken von der Eisscholle darstellen: die Eisdicke („Ice thickness“), die Freibord („Freeboard“), die Tiefe („Draft“), die Schneedicke („Snow thickness“), die Freibord und Schneedicke zusammen („Surface elevation“), die Dicke der Verwitterungsschicht (wie die Schneedicke, aber nur im Sommer, „Surface layer

thickness“), Schmelztümpeltiefe („Melt pond depth“), Schmelztümpel-Boden-Z-Koordinate („Melt pond Z coordinate“). Folgende Abbildung stellt die Eisscholle mit ihren Parametern dar.

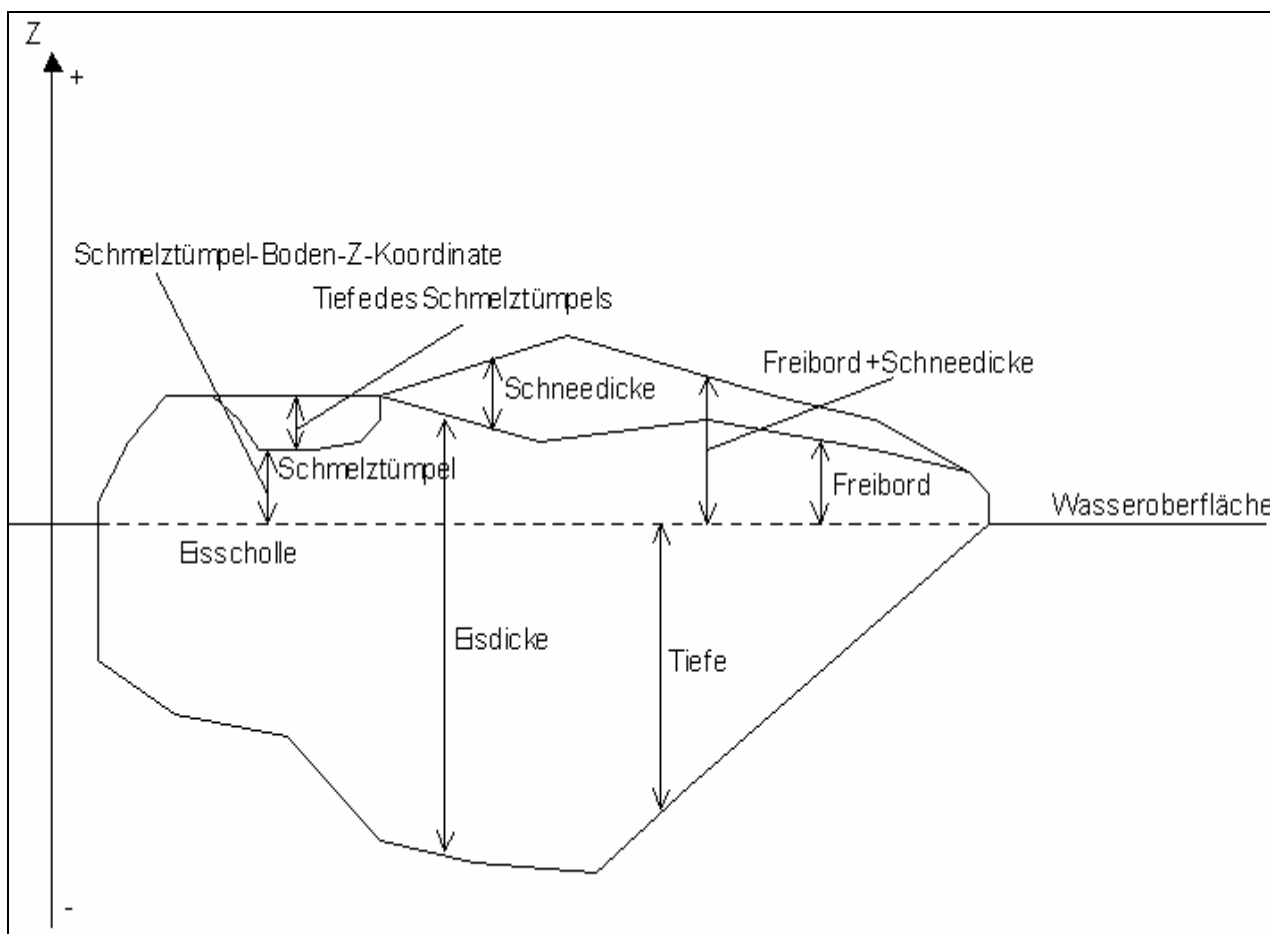


Abbildung 1: Die Eisschollen mit ihren Parametern
Quelle: eigene Darstellung

Je nach Definition können die Werte von der Parametern „Measurement point distance“, „Draft“ und „Melt pond bottom z coord“ positiv oder negativ sein. Wählt man der Rand der Eisscholle als Startpunkt der Messung, sind die Werte des Parameters „Measurement point distance“ immer positiv. Fängt man die Messung in der Mitte der Eisscholle an, werden die Werte von diesem Parameter positiv, bis man zu einem Rand der Eisscholle gelangt. Läuft man danach von der Mitte zu dem anderen Rand, werden die Werte des Parameters „Measurement point distance“ negativ. Die nachfolgende Tabelle stellt die zurzeit untersuchten Parameter und die Messmethoden, mit denen die Werte dieser Parameter bei dem Verfahren „Ground thickness profiles“ gemessen werden.

Parameter	Messmethode
Measurement point distance	Ruler tape
Ice thickness	Drilling and ruler tape; Electromagnetic sounding (EM), GeonicsEM31 conductivity sensor, Haas et al 1997
Freeboard	Drilling and ruler tape
Draft	Drilling and ruler tape; Electromagnetic sounding (EM), GeonicsEM31 conductivity sensor, Haas et al 1997
Snow thickness	Drilling and ruler tape
Surface layer thickness	Drilling and ruler tape
Surface elevation	Drilling and ruler tape
Melt pond depth	Drilling and ruler tape
Melt pond bottom Z coordinate	Drilling and ruler tape

Tabelle 1: Die Parameter und Messmethoden in dem Projekt „Ground thickness profiles“

Wie man sieht, heißt die überwiegend benutzte Messmethode „Drilling and ruler tape“. Die Messungen mit dieser Methode laufen folgendermaßen ab: zuerst wird das Loch gebohrt, danach wird der oder anderer Parameter mit dem Messband gemessen. Alle Messwerte, die mit dieser und der Messmethode „Ruler tape“ (das Messband) erhoben werden, werden auch in dem Notizblock vermerkt. Die Messmethode „Electromagnetic sounding (EM), GeonicsEM31 conductivity sensor, Haas et al 1997“ wurde von Herrn Haas mit der Forschergruppe beschrieben und in PANGAEA im Jahre 1997 als eigenständige wissenschaftliche Messmethode aufgenommen. Diese Messmethode wird mit dem elektromagnetischen Sensor Geonics EM31 von dem kanadischen Unternehmen „Geonics Limited“ durchgeführt. Der Sensor sieht äusserlich wie ein tragbarer (mit dem Griff oben) Kasten aus, an dessen Seitenflächen die Balken angebracht sind. Diese Balken haben an ihren Enden sendende und empfangende elektromagnetische Einheiten (Induktionsspulen), die voneinander 3,66 Meter entfernt sind. Die unten folgende Abbildung demonstriert diesen Sensor in dem Kajak.



Abbildung 2: Die Messung mit dem Sensor „Geonics EM31“
Quelle: www.awi-bremerhaven.de

Aus mir vorliegenden Dateien kann ich den Rückschluss machen, dass dieser Sensor während der Expedition ARK-XI/1 im Laptev-See im Jahre 1995 zum ersten Mal für die Eisdickenmessungen eingesetzt wurde.

Der Sensor wird in den Kajak gelegt und mit dem an AWI entwickelten Daten-Logger verbunden. Der Kajak wird über die Eisschollenoberfläche von einem Wissenschaftler gezogen. Betätigt man den Schalter des Sensors, wird der elektromagnetische Impuls von der Sendeeinheit ausgestrahlt und von der Empfangseinheit registriert. Dabei werden die Stärken der elektromagnetischen Felder gemessen und erst als prozentuelle Angabe (so genannte RPM-Wert) von dem Daten-Logger registriert und danach in ASCII-Datei als Mikrovolt geschrieben.

Unten folgt die schematische Darstellung der in dieser Messmethode verwendeten Komponenten.

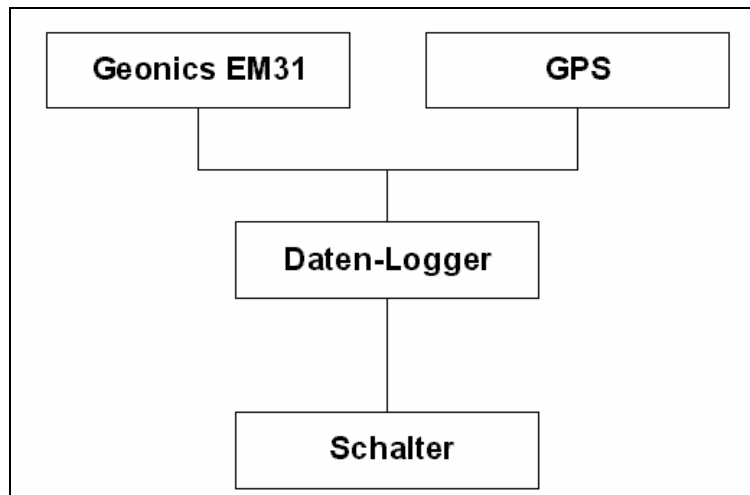


Abbildung 3: Komponenten der Messmethode "Electromagnetic sounding (EM), GeonicsEM31 conductivity sensor, Haas et al 1997"

Quelle: Christian Haas

Nach Ende der Messungen nimmt man den Daten-Logger am Bord des Schiffes (z.B. „Polarstern“), schließt an das Laptop an und führt das von dem Herrn Haas entwickelte Verfahren „Spannungs-Dicken-Transformation“ durch. Dieses Verfahren wurde mittels des Programms „IGOR Pro“ von der Firma „WaveMetrics, Inc“, das ein Werkzeug für Datenerfassung und -analyse darstellt und die Makrosprache zur Datenmanipulation hat, implementiert. Als Ergebnis bekommt man aus registrierten Mikrovolt die Eisdicken in Meter, die in einer ASCII-Datei gespeichert werden. In diese so genannte Datendatei (eine für alle Eisschollen) kommen noch die Werte von den oben erwähnten Parameter. Der Name von dieser Datei hat die Endung „.dat“.

Die Metadaten von allen untersuchten Eisschollen werden in die so genannte Metadatendatei, die eine ASCII-Datei ist und die Endung „.txt“ hat, per Hand übertragen. Die Werte in den Zeilen von diesen beiden Dateien sind durch den Tabulator voneinander getrennt.

An diese Stelle wird vermerkt, dass die Formate und Ausschnitte aus allen Dateien, die bei allen Messverfahren den Ausgangsdatenbestand darstellen, in dieser Arbeit in der tabellarischen Form dargestellt werden. In Wirklichkeit sind es ASCII-Dateien, deren einzelne Metadaten- und Datenwerte entweder durch den Tabulator oder das Leerzeichen voneinander getrennt sind. Auch werden diese Dateien als Quellendateien bezeichnet, weil sie die Quellen darstellen, deren Daten mit dem von mir zu entwickelten Programm verarbeitet werden.

Station	Latitude	Longitude
222.1	77.5674	68.7765
222.2	77.5652	68.7783
223	77.5648	68.7791
...

Tabelle 2: Format der Quellenmetadatendatei in dem Projekt „Ground thickness profiles“

In der Metadatendatei bezeichnet jede Zeile eine Eisscholle eindeutig. Die Anzahl der Zeilen in dieser Datei ist gleich der Anzahl der untersuchten Eisschollen.

Jede Datendatei (eine für eine Expedition) beinhaltet die Zeilen (größte Anzahl bisher war 22447), die die Messdaten von allen während einer Expedition untersuchten Eisschollen (maximale Anzahl bisher war 54) darstellen. Jede Zeile in dieser Datei stellt eine Messung auf irgendeiner Eisscholle und beinhaltet den Namen der Eisscholle, die Entfernung des Messpunktes von dem Messungsanfang auf dieser Eisscholle (der Wert des Parameters „Measurement point distance“) und eigentliche Nutzdaten: die Werte der in der Tabelle 1 genannten Parameter. Die Spaltenreihenfolge in diesen Dateien ist nicht fest vorgeschrieben und unterscheidet sich von Expedition zu Expedition. Nur die Spalten mit den Namen der Eisschollen (Station) und den Werten von dem Parameter „Measurement point distance“ standen bislang immer vorne, obwohl in abwechselnden Reihenfolge. Auch die Spaltennamen in den Datendateien von verschiedenen Expeditionen unterscheiden sich, was heißt: wenn es um die Eisdickenwerten geht, die mit der Messmethode „Drilling and ruler tape“ gemessen wurden, heißt die entsprechende Spalte nicht immer „Ice thickness (DH)“, sondern kann von dem Herrn Haas mit dem ihm verständlichen und nicht immer demselben Kürzel benannt werden.

Auch nicht während jeder Expedition werden alle Eisschollenparameter gemessen, was bedeutet, dass manche Spalten in der Datendatei ganz fehlen.

Die Unten folgende Ausschnitte aus den Dateien von verschiedenen Expeditionen demonstrieren die oben erläuterte Tatsache.

Station	Point distance	Ice thickness (DH)	Draft (DH)	Zsl
217	0	2.463454545	2.32243434	0.08123323
217	5	2.191243434	1.97748484	0.05867676
217	10	2.422439484	2.32847465	0.29475657
...

Tabelle 3: Ausschnitt aus der Datendatei von der Expedition ARK-VIII/3 (Transpolar Drift, 02.08.1991 – 08.10.1991)

Station	P_d	Draft (DH)	Zse	Freeboard	Zsl	I_t (DH)
246	0	-0.15546565	0.00545654	0.00534535	0	0.16453457
246	2	-0.19342444	0.02344546	0.02144535	0	0.21386855
...

Tabelle 4: Ausschnitt aus der Datendatei von der Expedition ARK-IX/4 (Laptev-See, 06.08.1993 – 05.10.1993)

Station	P_d	I_t (EM)	I_t (DH)	Fb	Draft (EM)	Draft (DH)	Zsl	Zse	Mpd
203	0	3.252434	2.579446	0.36	-2.8734455	-2.2453525	0.0212	0.38234	
203	5	
203	
205	0	1.687656	1.653535	0.44	-1.1684743	-1.3575454	0.8123	0.52145	
205	0.5							0.53665	
205	1							0.44856	
...	

Tabelle 5: Ausschnitt aus der Datendatei von der Expedition ARK-XI/1 (Laptev-See, 07.07.1995 – 20.09.1995)

Aus Platzgründen habe ich die Abkürzungen in der Kopfzeilen der oben dargestellten Tabellen verwendet, die an diese Stelle erläutert werden müssen:

- P_d und Point distance bezeichnen den Parameter „Measurement point distance,,
- I_t - „Ice thickness,,
- Fb - „Freeboard,,
- S_t - „Snow thickness,,
- Zsl - „Surface layer thickness,,
- Zse - „Surface elevation,,
- Mpd - „Melt pond depth,,

So einheitlich sehen die Spaltennamen in dem Idealfall aus. Die unten dargestellte Tabelle zeigt den Ausschnitt aus der Datendatei der Expedition ARK-XVII/2 (Reisegebiet: Arktische Ozean, die Zeitraum: 31.07.2001 – 07.10.2001) mit dem originalen Spaltennamen.

DistEM	StationEM	TTEM	TTdrillDB	DRdrillDB	ZsDB
0	216		4.299999999999999	- 3.85	0.08
5	216	6.36620560155165			
...	216	...			
20	216	3.13718243065543	2.959999999999999	-2.62	0.08
...
0	227				
...	...				
155	227				
160	227				
165	227	10.083328777467			
170	227	8.13741194808028			
...	227	...			
...	228
...

Tabelle 6: Ausschnitt aus der Datendatei der Expedition ARK-XVII/2 (Reisegebiet: Arktische Ozean, die Zeitraum: 31.07.2001 – 07.10.2001)

Dabei steht

- DistEM für den Parameter „Measurement point distance“,
- StationEM – für den Eischollennamen,
- TTEM – für die Werte des Parameters „Ice thickness“, die mit der Messmethode „Electromagnetic sounding (EM), GeonicsEM31 conductivity sensor, Haas et al 1997“ erhoben wurden,
- TtdrillDB – für die Werte des Parameters „Ice thickness“, die mit der Messmethode „Drilling and ruler tape“ erhoben wurden,
- DRdrillDB – für die Werte des Parameters „Draft“, die mit der Messmethode „Drilling and ruler tape“ erhoben wurden,
- ZsDB – für die Werte des Parameters „Surface layer thickness“, die mit der Messmethode „Drilling and ruler tape“ erhoben wurden.

So ähnlich (aber nicht gleich, und vor allem: jedes Mal anders) sahen auch die tatsächlichen Spaltennamen in allen Datendateien von diesem Projekt aus, was keine Systematik von den verwendeten Kürzel und den existierenden Parameter ableiten liess.

Es kommt nicht selten vor, dass bei etlichen Eisschollen (in obere Tabelle z.B. 227) mehrere Spalten leer sind, obwohl die Werte der entsprechenden Parameter während dieser Expedition gemessen wurden. Diese Tatsache ist darauf zurückzuführen, dass die Zeit manchmal aus irgendwelchen Gründen nicht ausreicht, um die Löcher zu bohren. Die Eisdickenmessungen mit dem elektromagnetischen Sensor Geonics EM31 wurden aber seit dem Jahre 1995 immer durchgeführt

(wobei auch da Aussetzer passieren können, was in der Tabelle oben bei der Eisscholle mit dem Namen 227 bei den Entfernungen 155 und 160 Meter von dem Startpunkt der Fall ist). Fehlt der Wert des Parameters „Measurement point distance“, ist die Datenzeile ungültig.

Weil die Schmelztümpel nur gelegentlich auf der Eisschollen vorkommen, sind die entsprechenden Spalten in Datendateien (wenn die Werte von den Parametern „Melt pond depth“ und „Melt pond bottom Z coordinate“ überhaupt in einer Expedition gemessen werden) meistens leer.

3.1.2 Messverfahren „Freeboard profiles“

Bei diesem Messverfahren werden die Messungen der großen Meeresgebiete während der Hubschrauberflügen durchgeführt. Nach meinen Erkenntnissen stammen die erste Messungen mit diesem Verfahren aus der Expedition ARK-XI (im Jahre 1995), obwohl ich bisher nur die Messdaten aus der Expedition ARK-XII (Jahr 1996) verarbeitet. Als Messinstrument wird die Messsonde namens „HEM Bird“ benutzt, die an AWI entwickelt wurde, und am Anfang nur die Lasereinrichtung hatte. Unten auf der Abbildung ist der sendende Teil der Einrichtung rechts, der Empfangsteil – links im Gerät.

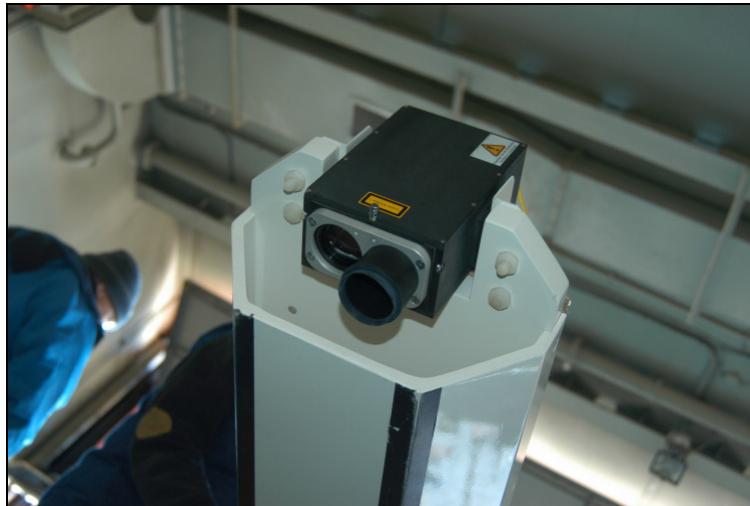


Abbildung 4: Lasereinheit der Messsonde „HEM Bird“

Quelle: www.awi-bremerhaven.de

Zusammen mit dem Rechner, der die drahtlose Netzwerk- und GPS-Karten besitzt, wurde die Lasereinrichtung in einer 3 Meter langen Rohr integriert, der noch vorne die Kappe mit dem nach unten zeigenden Glasfenster für den Laserstrahl und hinten den Flugleitwerk verpasst wurden.

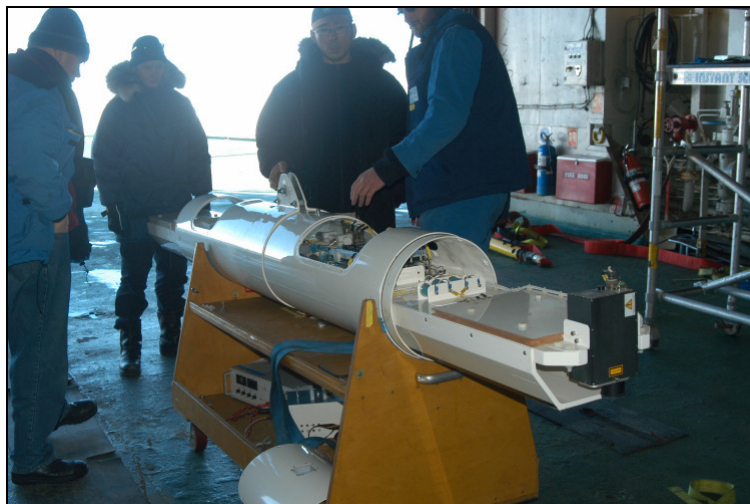


Abbildung 5: Die Messsonde „HEM Bird“ von innen

Quelle: www.awi-bremerhaven.de

Erst im Jahr 2001 kam die elektromagnetische Messeinrichtung hinzu, was aber für die Datenerhebung bei diesem Verfahren ohne Bedeutung ist.



Abbildung 6: HEM Bird vor dem Einsatz

Quelle: www.awi-bremerhaven.de

Die Messsonde wird an dem Hubschrauber mittels des 20 Meter langen Drahtseils befestigt, und der Flug beginnt. In der Kabine des Hubschraubers befindet sich außer dem Piloten der Wissenschaftler mit dem Laptop, auf dem das Datenaquisitionssystem (DAS) installiert ist und läuft. Das Laptop und der Rechner der Messsonde bilden ein drahtloses peer-to-peer Netzwerk. Die Lasereinrichtung sendet ein Lasersignal mit der Frequenz 100 Hz, was 100 Messungen pro Sekunde bedeutet. Abhängig von der Fluggeschwindigkeit (durchschnittlich 60 – 80 Knoten, was etwa 120 km/h entspricht) liegen die Messpunkte ungefähr 0,15 – 0,3 Meter voneinander getrennt. Mit dem Lasersignal werden die Höhe des Geräts über die Oberfläche und die Intensität des reflektierten Lasersignals ermittelt und die entstehenden Werte (Mikrovolt als ASCII-Wert) per Funk auf das Laptop in der Hubschrauberkabine übermittelt. Das auf dem Laptop laufende Datenaquisitionssystem berechnet aus diesen Angaben das Freibord („Freeboard“) und die Eisbedeckung („Ice coverage“) in Meter und zeichnet die Werte als ASCII auf. Unten folgt die schematische Darstellung aller Einrichtungen von diesem Datenerhebungsverfahren, wobei, wie oben schon vorgemerkt wurde, die elektromagnetische Einheit bei diesem Messverfahren keine Rolle spielt.

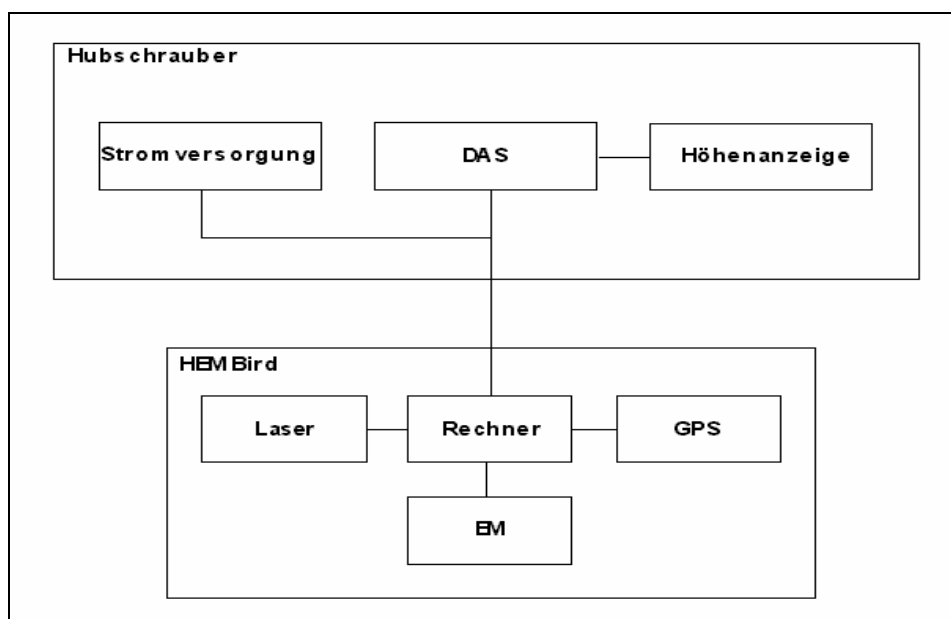


Abbildung 7: Schematische Darstellung aller am Messverfahren beteiligten Komponenten
Quelle: Christian Haas



Abbildung 8: Die Messung in Aktion
 Quelle: www.awi-bremerhaven.de

Während eines Hubschrauberfluges führt man mehrere Messungen (1 - 7 bislang) durch. Bei jeder Messung entstehen zwei Arten von Dateien: eine Metadaten- und eine Datendatei, die in einem für dieses Flug erzeugten Verzeichnis abgelegt werden. Die Metadatendatei beinhaltet die Angaben über den Zeitpunkt und die geografische Lage (geografischen Breite und Länge) des Anfangs und des Endes der Messung. Die Namen der Metadatendateien fangen mit der Zeichenkette „la,“ und beinhalten die Angaben zu dem Datum der Messung und dem Zeitpunkt des Messungsanfangs (z.B. stehen in der Zeichenkette „la072508.42,“, 3 und 4 Zeichen für Monat (der Juli), 5 und 6 Zeichen für den Tag (25), 7 und 8 Zeichen für Stunden, 10 und 11 für Minuten). Im Anhang A ist die Metadatendatei „la072508.42,“ aus diesem Projekt als Beispiel aufgeführt. Darauf folgende Aufführungen treffen jede Metadatendatei von diesem Projekt zu. In der 2. Zeile der Metadatendatei steht die Uhrzeit des Messungsanfangs, in der 3. Zeile – die Uhrzeit des Messungsendes. In der ersten Zeile (gezählt von der Dateianfang), die 7 voneinander durch das Leerzeichen getrennte Spalten beinhaltet, sind die 4. und 5. Elementen die Angaben (die Grads und Minuten) über die geografische Breite, 6. und 7. Elementen – über die geografische Länge von der Messungsanfang. In der ersten Zeile (gezählt von der Dateiende), die 7 Spalten (wie oben) beinhaltet, sind die 4. und 5. Elementen die Angaben über die geografische Breite, 6. und 7. Elementen – über die geografische Länge von dem Messungsende.

Die Messwerte werden in der Datendatei (eine für eine Messung) gespeichert. Jede Zeile der Datendatei beinhaltet die Entfernung von dem Messungsanfang bei einem ausgestrahlten Lasersignal und die dabei registrierte Werte der untersuchten Parameter.

Unten ist der Ausschnitt aus der Datendatei „07250842a.dat,“ von der Expedition ARK-XII (das Reisegebiet: Kara/Laptevsee/Transpolar Drift, die Reisezeit: 12.07.1996 – 23.09.1996) dargestellt.

Measurement point distance	Freeboard	Ice coverage
0.155401	-0.031422	1
0.310802	-0.023117	1
0.466203	-0.022617	1
0.621604	-0.045559	1
0.777005	-0.006012	1
0.932406	-0.028969	1
1.087807	-0.113609	0
...

Tabelle 7: Ausschnitt aus der Datendatei „07250842a.dat,“ von der Expedition ARK-XII

Insgesamt beinhaltet diese Datei 234 898 Zeilen mit Messwerten. Die Werte in jeder Zeile von den Datendateien sind durch das Leerzeichen voneinander getrennt. Die Spalten von den Datendateien in diesem Projekt hatten bislang keine Namen. Die Kopfzeile in der obere Tabelle wurde von mir besserer Übersichts halber eingefügt. Die erste Spalte stellt die Entfernungswerte von dem Messungsanfang (Parameter „Measurement point distance,“) dar und wird aus der geografischen Breite und Länge während des vorangehenden und aktuellen Laserausstrahlung berechnet (diese Messmethode heisst entsprechend „Calculated from GPS,“), die zweite – die Freibordwerte (Parameter „Freeboard,“, erhoben mit der Messmethode „Determined by helicopter-borne laser distance measurement,“), die dritte – die Werte der Eisbedeckung (Parameter „Ice coverage,“, gemessen mit der Methode „Determined from reflected laser signal strength,“, wobei „0,“ kein Eis (das Wasser) und „1,“ das Eis bedeuten).

Die Datendateien in diesem Messverfahren sind sehr umfangreich und können von 30 000 bis 300 000 Zeilen haben (abhängig von der Flugdauer und der Geschwindigkeit des Hubschraubers). Bisher hatte ich in diesem Projekt nur die Forschungsdaten aus der Expedition ARK-XII im Jahre 1996. Damals wurden 23 Hubschrauberflüge gestartet, wobei 86 Messungen durchgeführt wurden. Der Datenbestand stellte ein Verzeichnis mit 23 Unterverzeichnissen (1 für 1 Flug), jedes von diesen Verzeichnissen beinhaltete die Metadaten und Daten von der Messungen während des einzigen Fluges. Diese Unterverzeichnisse tragen die Namen, die einen Aufschluss auf die Expedition und den Messungstag geben (z.B. stehen in der Zeichenkette „helir12225“ erste 5 Zeichen („helir“) für „Helicopter“, 6 und 7 Zeichen („12“) – für die Expedition „ARK-XII“, den Rest („225“) bezeichnet den julianischen Tag der Messung). Bisher sahen die Datendateien von diesem Messverfahren sehr vernünftig aus. Sie hatten keinerlei Fehlwerte. Werden die Werte des Parameters „Measurement point distance,“ fehlen, sind solche Datensätze ungültig.

3.1.3 Messverfahren „HEM Thickness Profiles“

Dieses Messverfahren nutzt den Vorteil von der elektromagnetischen Technik, die seit dem Jahre 2001 aus den Hubschraubern oder Flugzeugen angewendet werden kann und die hochauflösende Eisdickenwerte von guter Qualität liefert. Mit dem „HEM-Bird“ ist es möglich geworden die systematischen großräumigen Untersuchungen der Eisdickenverteilungen durchzuführen, die die Fähigkeit verleihen die Beobachtungen und Vorhersagen von den möglichen klimatischen Veränderungen besser zu beurteilen.

Quelle: www.awi-bremerhaven.de

Dabei werden die Messungen wie bei dem Verfahren „Freeboard profiles“ während der Hubschrauberflüge durchgeführt. Als Nutzdaten werden statt Freibord und Eisbedeckung die Eisdickenwerte (Parameter „Ice thickness“) mit der elektromagnetischer Einheit des Messensors „HEM Bird“ registriert. Weil diese Einheit die elektromagnetische Impulse mit der Frequenz 10 Hz ausstrahlt (was 10 Signale pro Sekunde bedeutet), liegen die einzelne Messpunkte etwa 3-4 Meter (ausgehend aus der durchschnittliche Fluggeschwindigkeit 120 km/h) voneinander entfernt. Die Höhe des Sensors über die Oberfläche während einer Messung wird mit der Lasereinheit (laser altimeter) bestimmt. Das Datenerfassungs- und Konvertierungsverfahren funktioniert wie bei dem Messverfahren „Freeboard profiles“. In Folge einer Messung entsteht eine ASCII-Datendatei (weiter einfach die Datei genannt, weil die Metadatendateien bei diesem Verfahren fehlen) mit der Endung „.dat“. Diese Datei wird in einem für diesen Flug erzeugten Verzeichnis abgelegt. Während eines Fluges werden mehrere Messungen durchgeführt (von 1 bis 7, der Durchschnittswert liegt bei 4). So wurden z.B. während der Expedition IRIS2003 12 Flüge mit 59 Messungen, während der Expedition IRIS2004 19 Flüge mit 82 Messungen durchgeführt, entsprechend gibt es genau so viele Verzeichnisse und Dateien (bei IRIS2004 liegen 82 Dateien in 19 Verzeichnissen).

Die Struktur der Dateien bei diesem Projekt stellen die Ausschnitte aus der Dateien von der Expeditionen IRIS2003 (das Reisegebiet: Ostsee, die Reisezeit: 17.02.2003 – 23.02.2003) und IRIS2004 (das Reisegebiet: Bothische Bucht, die Reisezeit: 05.02.2004 – 17.03.2004) dar.

lati	long	dx	fid	ppm1_thick	height_dec
			2479		92.4625
60.1804	25.27155		2480		92.344
60.18037	25.27161		2481		92.184
...
60.14529	25.34342	0	3649		19.968
...
60.14221	25.34883	455.9797	3745	0.7272663	17.298
60.14217	25.34888	460.7466	3746	0.6722889	17.308
60.14214	25.34893	465.5135	3747	0.6677284	17.324
...
			10234		
			10235		
			10236		

Tabelle 8: Ausschnitt aus der Datei „200302170937_allfinal.dat“ von der Expedition IRIS2003

Latitude	longitude	distance	fid	ppm2_thick	height_dec
65.05635	24.45081	0	3092		19.974
65.05635	24.45075	2.949629	3093		20.042
65.05634	24.45068	6.104676	3094		19.982
65.05634	24.45062	9.143688	3095		19.932
65.05634	24.45055	12.1827	3096		19.938
65.05634	24.45049	15.22171	3097	0.6847787	19.888
...

Tabelle 9: Ausschnitt aus Datendatei „200402051152_allfinal.dat“ von der Expedition IRIS2004

Wie man es sieht, unterscheiden sich die Spaltennamen in diesen Dateien von beiden Expeditionen (oder anders ausgedrückt: wie bei dem Messverfahren „Ground thickness profiles“ auch, kann man sich darauf nicht verlassen, dass die Spaltennamen in der Dateien von allen Expeditionen gleich bleiben).

Die ersten 2 Spalten stellen die Werte der geografischen Breite und Länge (entsprechen den Parametern „Latitude“ und „Longitude“, sind mit der Messmethode „GPS“ bestimmt) dar, die dritte – die Entfernungswerte vom Messungsanfang (entspricht dem Parameter „Measurement point distance“, sind aus der geografischen Breite und Länge berechnet, die Messmethode heißt „Calculated from GPS“), die vierte – die Messungsnummer (entspricht dem Parameter „Fid number“, die Messmethode dafür heißt „not given“), die fünfte – die Eisdickenwerte (entspricht dem Parameter „Ice thickness“, sind mit der Messmethode „Measured from helicopter-borne EM sounding“ bestimmt), die sechste – die Höhenwerte des Messensors über die Eisoberfläche (entspricht dem Parameter „Device height above ground“, sind mit der Methode „Determined with laser altimeter on EM Bird“ gemessen). Die Spalten sind voneinander durch den Tabulator getrennt. Jede Zeile stellt die Werte der oben erwähnten Messpositionen (Parameter + Messmethode) für den einzelnen Punkt der Messung. Fehlen bei einer oder mehreren Zeilen die Angaben in den ersten zwei Spalten (wie in den ersten und letzten 3 Zeilen der Tabelle mit dem Ausschnitt aus der Datei von der Expedition IRIS2003 der Fall ist), lassen sich die Messpunkte nicht mehr geografisch zuordnen.

Solche Datenzeilen sind als ungültig zu betrachten. In der Dateien bei diesem Verfahren fehlen gelegentlich die Messwerte bei den Parametern „Measurement point distance“ und „Ice thickness“. Das wird damit erklärt, dass sich der Messsensor während der Messung höher als 20 Meter über die Eisoberfläche befand (laut der Gebrauchsanweisung muss die Maximalhöhe des Geräts über der zu untersuchende Oberfläche 15 Meter nicht übersteigen).

Jede Datendatei beinhaltet von 1500 bis 19000 Zeilen (abhängig von der Flugdauer und der Geschwindigkeit des Hubschraubers), die die Daten einer Messung darstellen.

Die Dateinamen beinhalten die Information über die Zeitpunkt der Messungen (z.B. in den Dateinamen „200402290922_allfinal.dat.“, stehen "2004" (erste 4 Zeichen) für das Jahr, "02" (5. und 6. Zeichen) – für den Monat, "29" (7. und 8. Zeichen) – für den Tag, "0922" (9. bis 12. Zeichen) – für die Uhrzeit des Messungsanfangs (9 Uhr 22 Minuten). Die geografischen Breiten und Längen des Messungsanfangs und des Messungsendes stehen in der ersten bzw. letzten gültigen Zeilen der Dateien. Insofern stehen die gleichen Angaben zu allen Messungen, wie in der Metadatendatei von der zwei anderen Messverfahren, zur Verfügung. Lediglich die Uhrzeit des Messungsendes fehlt.

3.2 Anforderungen an die Daten für die Archivierung

Wie schon oben erklärt wurde, entstehen während der Datenerhebung bei allen Verfahren 2 Arten von Daten: die Metadaten und die eigentliche Daten. Die Metadaten in diesem Fall sind das Datum, die Zeitpunkt und der geografische Lage der Messungen. Die Daten stellen die Werte von bestimmten Parametern, die mit der bestimmten Messmethoden erhoben werden, dar. Der Parameter und die Methode, mit der die Werte von diesem Parameter gemessen wurden, bilden zusammen die Messposition.

Leider sind die Metadaten in diesem Umfang nicht ausreichend, um die Archivierung in der Datenbank von PANGAEA und „SeaIceDB“ zu ermöglichen. Die Werte der Messpositionen (eigentliche Daten) haben unnötig viel Nachkommastellen, um ohne die zusätzliche Datenverarbeitung in die Datenbanken eingefügt zu werden.

Die Dateien, die den Datenbestand darstellen, werden ab dieser Stelle weiter als Quellendateien bezeichnet, weil sie die Quellen beinhalten, aus deren das Programm die Dateien im geforderten Format erzeugt.

Die Unterscheidung nach den Projektbereichen ist aus den Anforderungen von den 2 Wissenschaftlern entstanden, die mir die Aufgaben stellten und mich während des Praxissemesters und der Diplomarbeit betreuten. Der erste ist Herr Dr. Christian Haas (weiter als Auftraggeber 1 bezeichnet), der das von mir entwickelte Programm in seiner Tätigkeit benutzen wird. Unter Anderem wird er mit dem Programm die Dateien erzeugen, die er für das Einfügen der Daten in die Datenbank „SeaIceDB“ benutzen wird.

Die Anforderungen an die Daten für die Archivierung in der Datenbank von PANGAEA stammen von dem Herrn Dr. Rainer Sieger (weiter als Auftraggeber 2 bezeichnet), der mich während der Diplomarbeit betreute. Er gehört zu der Gruppe der Fachleute, die für PANGAEA verantwortlich sind.

Um die Anforderungen kürzer und klarer differenzieren zu können und später die Implementierung verständlich zu beschreiben, werden an dieser Stelle die Begriffe „Projektbereich“ und „Projekt“ eingeführt. In dem zu entwickelndes Programm werden entsprechend der Anforderungen 2 Projektbereiche - „PANGAEA“ und „SeaIceDB“ - und 3 Projekten, die die Namen der Messverfahren tragen, gebildet. Die Begriffe „Messverfahren“ und „Projekt“ werden weiter synonym verwendet.

Gemäß beiden Anforderungen müssen 2 Arten von ASCII-Dateien zu jeder Expedition bei allen Messverfahren geliefert werden: eine Metadatendatei für alle Event Labels der Expedition und

etliche Datendateien (eine für jeden Event Label). Alle Spalten in diesen Dateien sind durch den Tabulator voneinander zu trennen. Die Namen der erzeugten Dateien für den Projektbereich „PANGAEA“ müssen die Endung „.txt“ haben. Diese für Windows erzeugten Dateien stellen das Endergebnis in diesem Bereich dar. Sie werden an Herrn Sieger übermittelt, der die darin enthaltene Metadaten und Daten in die Datenbank von PANGAEA übernimmt.

Die Erzeugung der Dateien in dem Bereich „SeaIceDB“ ist für die Kontrolle gedacht, die durchgeführt wird, bevor die Daten in die Datenbank „SeaIceDB“ eingefügt werden.

Die Namen der in diesem Bereich erzeugten Metadatendateien müssen die Endung „.txt“, die Namen der Datendateien – „.dat“ aufweisen.

3.2.1 Anforderungen an Event labels

Übersetzt aus dem Englisch bedeutet „Event Label“, nichts Anderes als „Name des Ereignisses“. Bei den verschiedenen Messverfahren hat Event Label aber die unterschiedliche Bedeutung.

Bei dem Verfahren „Ground thickness profiles“ stellen die Event Labels die Namen der untersuchten Eisschollen dar. Dabei setzt sich Event label aus der Anfangszeichenkette für die Expedition (z.B. „Ark12_“ für ARK-XII oder „Ark9_4_“ für ARK-IX/4), dem julianischen Tag der Messung und dem Suffix „p“ plus die Untersuchungsnummer dieser Eisscholle während diesem Tag zusammen. Ist die Eisscholle groß, schaffen die Wissenschaftler nur sie an diesem Tag zu vermessen. Event Label ist dann z.B. „Ark9_4_247p1“. Am nächsten Tag werden 3 Eisschollen untersucht, die Event labels sind entsprechend „Ark9_4_248p1“, „Ark9_4_248p2“ und „Ark9_4_248p3“. Sie müssen von dem Programm aus der Angaben „248.1“ (oder einfach „248“), „248.2“ und „248.3“ in der Quellenmetadatendatei erzeugt werden.

Bei 2 anderen Messverfahren sind Event labels die Namen der Messungen.

Als Beispiel des Event labels bei dem Messverfahren „Freeboard profiles“ wird die Zeichenkette „Ark12_laser_207_09_08“ aufgeführt, wo „Ark12_“ für die Expedition, „207“ für den julianischen Tag der Messung (aus dem Namen des Verzeichnisses „helir207“ ermitteln), „09_08“ für die Uhrzeit (aus den letzten 5 Zeichen des Namen der entsprechende Quellenmetadatendatei „la072509.08“ konvertieren) stehen.

Bei dem Messverfahren „HEM thickness profiles“ unterscheiden sich die Anforderungen zum Format des Event labels. Für den Projektbereich „PANGAEA“ müssen sich die Event labels aus der Namen der Expedition und der Messungsnummer zusammensetzen. So bezeichnet „IRIS2003_1“ die erste Messung während der Expedition IRIS2003. Derselbe Event label in dem Projektbereich „SeaIceDB“ ist z.B. „200302290922“ und muss aus dem Namen der entsprechende Quellendatei „200302290922_allfinal.dat“ gewonnen werden.

3.2.2 Struktur von zu erzeugenden Metadatendateien im Projektbereich „PANGAEA“

Abhängig von dem Messverfahren unterscheiden sich (obwohl unwesentlich) die Spaltenstrukturen der zu erzeugenden Metadatendateien.

Campaign	Area	Label Event	Gear	Date Event	Latitude Event	Longitude Event
----------	------	-------------	------	------------	----------------	-----------------

Tabelle 10: Struktur der zu erzeugende Metadatendatei (Projekt: „Ground thickness profiles“, Projektbereich „PANGAEA“)

Dabei werden die Spaltennamen wie folgt interpretiert:

- Campaign steht für den Expeditionsnamen (z.B. ARK-IX/4),
- Area – für den Reisegebiet der Expedition (z. B. Laptev Sea),
- Label Event – für die Namen der Eisschollen (entspricht Event label; diese Spalte in der zu erzeugenden Metadatendateien muss aber tatsächlich so heissen),
- Gear – für die Kurzname des Geräts (hier ist immer ICEM),

- Date Event – für das Datum der Messung (Datumsformat: TT.MM.JJJJ),
- Latitude Event – für die geografische Breite der Eisscholle,
- Longitude Event – für die geografische Länge der Eisscholle.

Die Informationen über die Expedition (der Name, das Reisegebiet, der Anfang und das Ende), die Anfangszeichenkette des Event labels und den Kurznamen des Geräts sind nicht in der Quellenmetadatei und müssen von dem Benutzer dem Programm mitgeteilt werden. Das Datum (Tag und Monat) wird aus den julianischen Tag der Messung berechnet, die Jahresangabe wird aus dem Anfang (und bei Jahreswechsel während einer Expedition – aus dem Ende) der Expedition ermittelt.

Camp	Area	Label	Gear	Date	Time	Date2	Time2	Lat	Long	Lat2	Long2
------	------	-------	------	------	------	-------	-------	-----	------	------	-------

Tabelle 11: Struktur der zu erzeugenden Metadatei (Projekte: „Freeboard profiles“, „HEM thickness profiles“, Projektbereich „PANGAEA“)

Unten folgen die Spaltennummer als Aufzählung, der volle Spaltenname und, bei Bedarf, die Erklärung

Campaign (wie oben),

Area (wie oben),

Label Event (wie oben),

Gear (wie oben, bei dem Messverfahren „Freeboard profiles“, ist es LASER, bei dem Verfahren „HEM thickness profile“ - EMB),

Date Event (Datum des Messungsanfangs, Datumsformat: TT.MM.JJJJ, Tag und Monat werden aus dem Namen der entsprechende Quellenmeta- (bei dem Verfahren „Freeboard profiles“) oder Quellendatei (bei dem Verfahren „HEM thickness profiles“, hier auch das Jahr) ermittelt, Jahresangabe bei „Freeboard profiles“ - wie oben),

Time Event (der Zeitpunkt des Messungsanfangs, wird entweder aus dem Inhalt der entsprechende Quellenmetadatei („Freeboard profiles“) oder aus den Namen der entsprechende Quellendatei („HEM thickness profiles“) ermittelt),

Date Event2 (Datum der Messungsende, stimmt mit dem Datum des Messungsanfangs, weil es am Tag gearbeitet wird),

Time Event2 (der Zeitpunkt des Messungsende, bei „Freeboard profiles“ - wie der Zeitpunkt des Messungsanfangs ermittelt, bei dem Messverfahren „HEM thickness profiles“ muss nachgerechnet werden: ist es die letzte Messung während eines Fluges - den Zeitpunkt des Messungsanfangs nehmen, 20 Minuten hinzuaddieren; gibt es die nachfolgende Messung in diesem Flug – ihre Startzeit wird zur Endzeit der vorangehende Messung)),

Latitude Event (geografische Breite bei dem Messungsanfang, wird entweder aus dem Inhalt der entsprechende Quellenmetadatei ermittelt und dabei die Grad in das Dezimalform umwandelt („Freeboard profiles“) oder aus dem Inhalt der entsprechende Quellendatei („HEM thickness profiles“) ermittelt),

Longitude Event (geografische Länge bei dem Messungsanfang, wird wie die geografische Breite bei dem Messungsanfang ermittelt),

Latitude Event2 (geografische Breite bei dem Messungsende, wird wie die geografische Breite bei dem Messungsanfang ermittelt),

Longitude Event2 (geografische Länge bei dem Messungsende, wird wie die geografische Breite bei dem Messungsanfang ermittelt).

Aus der oben dargestellten Struktur der Quellen- und zu erzeugende Metadateien ist ersichtlich, dass nicht alle Angaben dem Programm zur Verfügung stehen. Die fehlende Informationen (den Namen der Expedition, deren Reisegebiet und Zeitraum, den Kurznamen des Geräts, die Anfangszeichenkette des Event labels bei 2 Messverfahren und die Anzahl der Nachkommastellen für die Werte von der geografischen Breite und Länge) muss der Benutzer dem Programm mitteilen können.

Die erste Zeile jeder erzeugte Metadatendateien in diesem Projektbereich muss die dem Messverfahren entsprechende Kopfzeile sein.

3.2.3 Struktur von zu erzeugenden Datendateien im Projektbereich „PANGAEA“

Jede zu erzeugende Datendatei in diesem Projektbereich muss eine Spalte für Event label und Spalten für die Werte von dem gemessenen Parameter besitzen. Weil die Datendatei einem Event label gehört, bleibt er in dieser Datei gleich. Aus dieser Überlegung hat Herr Sieger entschieden, dass nur die erste Datenzeile den Event label aufweisen muss. In restlichen Zeilen müssen die leere Zeichenketten („“) in dieser Spalte stehen. Die Anzahl der Parameterspalten in jeder Datendatei muss der Anzahl der bei diesem Event label gemessenen Parameter entsprechen. Die Anzahl der Nachkommastellen von den Werten dieser Parameter muss von dem Benutzer bestimmt werden können.

Jede zu erzeugende Datendatei in diesem Projektbereich muss an ihren Anfang einen so genannten Header besitzen. Header beinhaltet die Informationen, die nach dem Einfügen der Daten aus diesen Dateien in die Datenbank von PANGAEA für die Visualisierung dieser Daten auf der Internetseite www.pangaea.de benutzt werden.

Klickt man mit der linke Maustaste in den grünen Bereich „DATA“, gelangt man auf die Seite von der PANGAEA-Suchmaschine „PangaVista“. Als Beispiel werden jetzt die Daten von dem Event label „Ark12_222p1“ angeschaut.

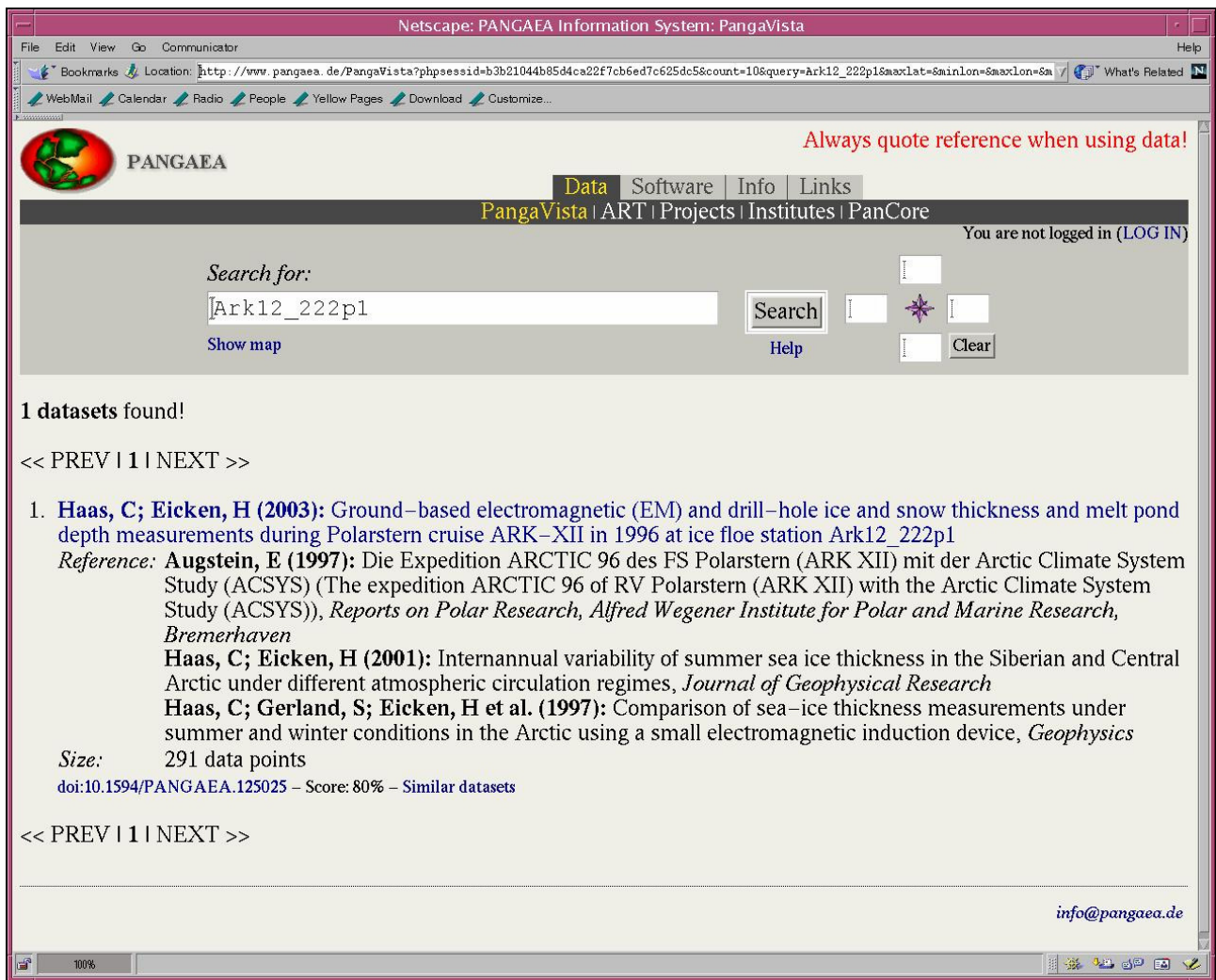


Abbildung 9: Ausschnitt aus der Internetseite von PANGAEA


Nach der Betätigung der Schaltfläche „Search“ bekommt man das Suchergebnis angezeigt. Folgt man den Link „Haas, C; Eicken, H (2003):“, gelangt man auf die Internetseite mit der Adresse <https://doi.org/10.1594/PANGAEA.125025>.

Netscape: Ground-based electromagnetic (EM) and drill-hole ice and snow thickness and melt pond depth measurements during Polarstern cru...

File Edit View Go Communicator Help

Bookmarks Location: <http://doi.pangaea.de/10.1594/PANGAEA.125025> What's Related

WebMail Calendar Radio People Yellow Pages Download Customize...

 PANGAEA Always quote reference when using data!

Data Software Info Links

PangaVista | ART | Projects | Institutes | PanCore

You are not logged in (LOG IN)

Data Description

Citation: Haas, Christian; Eicken, Hajo (2003): Ground-based electromagnetic (EM) and drill-hole ice and snow thickness and melt pond depth measurements during Polarstern cruise ARK-XII in 1996 at ice floe station Ark12_222p1, PANGAEA, doi:10.1594/PANGAEA.125025

Reference(s): Augstein, Ernst (1997): Die Expedition ARCTIC 96 des FS Polarstern (ARK XII) mit der Arctic Climate System Study (ACSYS) (The expedition ARCTIC 96 of RV Polarstern (ARK XII) with the Arctic Climate System Study (ACSYS)), *Reports on Polar Research, Alfred Wegener Institute for Polar and Marine Research, Bremerhaven*, **234**, 54 pp, <http://www.awi-bremerhaven.de/BIB/BerPolarforsch/BerPolarforsch1997234.pdf>

Haas, Christian; Eicken, Hajo (2001): Internannual variability of summer sea ice thickness in the Siberian and Central Arctic under different atmospheric circulation regimes, *Journal of Geophysical Research*, **106(C3)**, 4449–4462, doi:10.1029/1999JC000088

Haas, Christian; Gerland, Sebastian; Eicken, Hajo; Miller, Heinz (1997): Comparison of sea-ice thickness measurements under summer and winter conditions in the Arctic using a small electromagnetic induction device, *Geophysics*, **62/3**, 749–757

Project(s): Remote Sensing of Sea Ice Properties (FEME)

Spatial Coverage: West: 121.4400 * East: 121.4400 * South: 85.8600 * North: 85.8600

Event(s): Ark12_222p1 * Latitude: 85.8600 * Longitude: 121.4400 * DateTime: 1996-08-09T00:00:00 * Location: Kara/Laptev Sea/Transpolar Drift * Campaign: ARK-XII * Basis: Polarstern * Device: Ice measurement

Parameter(s):

Parameter	Short Name	Unit	Label	Principal Investigator	Method	Comment
POINT DISTANCE	Dist	m				Geocode

Abbildung 10: Ausschnitt aus der Datenbeschreibung für den Event label Ark12_222p1 (1)

Netscape: Ground-based electromagnetic (EM) and drill-hole ice and snow thickness and melt pond depth measurements during Polarstern cru...

File Edit View Go Communicator Help

Bookmarks Location: http://doi.pangaea.de/10.1594/PANGAEA.125025 What's Related

WebMail Calendar Radio People Yellow Pages Download Customize...

Parameter(s):

Parameter	Short Name	Unit	Label	Principal Investigator	Method	Comment
POINT DISTANCE from start	Dist	m				Geocode
LATITUDE	Latitude					Geocode
LONGITUDE	Longitude					Geocode
Draft	Draft	m	Ark12_222p1.13	Haas, Christian	Electromagnetic sounding (EM), Geonics EM31 conductivity sensor, Haas et al 1997	
Draft	Draft	m	Ark12_222p1.14	Haas, Christian	Drilling and ruler tape	
Freeboard	Freeboard	m	Ark12_222p1.12	Haas, Christian	Surface levelling/surveying	
Ice thickness	Ice thickness	m	Ark12_222p1.10	Haas, Christian	Electromagnetic sounding (EM), Geonics EM31 conductivity sensor, Haas et al 1997	
Ice thickness	Ice thickness	m	Ark12_222p1.11	Haas, Christian	Drilling and ruler tape	
Snow thickness	Snow thickness	m	Ark12_222p1.15	Haas, Christian	Ruler stick	
Surface elevation	Surf elev	m	Ark12_222p1.16	Haas, Christian		

Size: 291 data points

[Download dataset as tab-delimited text](#) – [View dataset as HTML](#)

Abbildung 11: Ausschnitt aus der Datenbeschreibung für den Event label Ark12_222p1 (2)

Alles, was dort unter „Data Description“ (außer den Abschnitten „Spatial Coverage“ und Event(s)) steht, wird aus der Information dargestellt, die im Header von jeder in dem Projektbereich „PANGAEA“ erzeugende Datendatei stecken muss.

Jetzt folgt den Header zu Event Label Ark12_222p1 aus der entsprechenden Datendatei.

```

/* DATA DESCRIPTION:
Author: 10730
      12345
Title: Ground-based electromagnetic (EM) and drill-hole ice and snow thickness and melt
pond depth measurements during Polarstern cruise ARK-XII in 1996 at ice floe station
Ark12_222p1
Reference: 10040
          25052
          25051
Export Filename: Ark12_222p1_ice_and_snow_thickness
Event: Ark12_222p1
PI: 10730
Parameter: 26097 * PI: 10730 * FORMAT: ###0.0
           14673 * PI: 10730 * METHOD: 4763 * FORMAT: #0.00
           14673 * PI: 10730 * METHOD: 4762 * FORMAT: #0.00
           8194 * PI: 10730 * FORMAT: #0.00
           26098 * PI: 10730 * METHOD: 4763 * FORMAT: #0.00
           26098 * PI: 10730 * METHOD: 4762 * FORMAT: #0.00
           5348 * PI: 10730 * FORMAT: #0.00
           26099 * PI: 10730 * FORMAT: #0.00
Project: 4038
URL Data details:\thttp://www.pangaea.de/documentation/Sea_ice/xxx.pdf
*/
Event label 26097 14673 14673 8194 26098 26098 5348 26099

```

Absatz „Author: 10730...” stellt die PANGAEA-IDs (kurz PIDs) von der Autoren des Zitats (ein wissenschaftliches Dokument, keine Referenz oder Methodenbeschreibung!) zu diesem Datensatz. In der Zeile „Title: Ground-based electromagnetic (EM)...” steht der Name des Datensatzes, der für alle Event Labels (Eisschollen in diesem Projekt) fast gleich ist. Einziger Unterschied ist Event Label selbst am Ende des Namens (hier Ark12_222p1).

Absatz „Reference: 10040...”, beinhaltet die PIDs von 3 wissenschaftlichen Referenzen (auch „papers” genannt) für diese Expedition.

Zu der Zeile „Export Filename:...”: Auftraggeber hat entschieden, dass jede von diesem Programm erzeugte Datendatei den Namen EventLabel.txt (in diesem Fall Ark12_222p1.txt) tragen muss. Sein Importer-Programm (das Programm, mit dem er die Daten aus erzeugten Datendateien in die Datenbank von PANGAEA einfügt) baut aus den Dateinamen Ark12_222p1.txt die Zeichenkette „Ark12_222p1_ice_and_snow_thickness” zusammen.

In der Zeile „Event: Ark12_222p1” steht der Event Label selbst.

In der Zeile „PI: 10730” steht der PID des Leiters der Projektgruppe, die diesen Daten erhoben hat, in diesem Fall ist es PID von Herrn Dr. Christian Haas.

In dem Absatz „Parameter: 26097...” hat jede Zeile folgenden Format und wird auf dem Beispiel der 2. Zeile dieses Absatzes erklärt. In der Zeile „14673 * PI: 10730 * METHOD: 4763 * FORMAT: #0.00” steht PID des Parameters (14673), PID des “Principal Investigators” (PI: 10730), wenn der Parameter kein Geocode (zu Geocode gehören die Parameter „Measurement point distance” (ist in der 1. Zeile dieses Absatzes), „Latitude” und „Longitude”) ist, dann folgt die Messmethoden-PID, mit der die Werte von diesem Parameter erhoben wurden („METHOD: 4763”), danach folgt „FORMAT: #0.00” und bedeutet: die Werte von diesem Parameter haben 2 Vorkomma- und 2 Nachkommastellen. Zwischen des Messmethoden-PID und der Formatangabe muss noch der Kommentar zu der Messmethode (z.B. „COMMENT: ice + snow thickness”) vorkommen, falls ein solcher Kommentar zu dieser Messmethode gibt (in unserem Beispiel ist es nicht der Fall).

Alle diese Angaben sind durch die Zeichenkette " * " voneinander getrennt. Der Absatz „Parameter” beinhaltet 8 solchen Zeilen, was bedeutet, dass der Benutzer 8 Parameter ausgewählt hat. Wenn aber in der Quelledatendatei die Werte von einem oder mehreren ausgewählten Parameter für diesen Event Label (Ark12_222p1) komplett fehlen, dürfen die entsprechenden Zeilen von solchen Parameter hier nicht auftauchen. Hat bei einem Event Label nur der Parameter „Measurement point distance” (sein PID ist 26097) die Werten und die andere ausgewählte Parameter – dagegen nicht, darf die Datendatei (und der Header entsprechend auch) für diesen Event Label gar nicht erzeugt werden.

In der Zeile „Project: 4038” steht der PID des Projektes „Remote Sensing of Sea Ice Properties”, in dessen Rahmen die Messungen mit der Verfahren „Ground thickness profiles” und „Freeboard profiles”. PID des Projektes, bei dem die Messungen von dem Messverfahren „HEM thickness profiles” durchgeführt wird, ist 4028.

Die Zeile „URL Data details:...” soll nur dann auftauchen, wenn eine Methodenbeschreibung zu diesem Projekt existiert (dann statt „xxx” am Ende des Links steht der Dateiname mit dieser Beschreibung). Im oberen Bild referenziert der als Link hinterlegte Methodenname „Electromagnetic sounding (EM), GeonicsEM31 conductivity sensor, Haas et al 1997” diesen Dokument, der auf dem Server www.pangaea.de/documentation/Sea_ice/ abgelegt wird.

In der Zeile „Event label 26097 14673 14673 8194 26098 26098 5348 26099” stehen die PIDs von der während dieser Expedition gemessenen Parameter; fehlen die Werte von einem oder mehreren untersuchten Parameter für diesen Event Label komplett, muss PIDs von diesen Parameter in dieser Zeile gar nicht auftauchen. Das trifft die Datendateien von dem Projekt „Ground thickness profiles” und hängt mit der Spezifik der Datenerhebung zusammen. Wie schon erwähnt wurde, werden bei diesem Messverfahren nicht auf allen Eisschollen die Werte von allen während dieser Expedition gemessenen Parameter erhoben, weil die Zeit manchmal nicht ausreicht, um die Löcher zu bohren oder anderen nicht elektromagnetische Messmethoden anzuwenden. Bei der Hubschrauberflügen entstehen solche Situationen nicht.

Wie man es sieht, ist hier die Menge der Information (PIDs von den Wissenschaftler, Referenzen, Parameter, Messmethoden, Namen des Datensatzes und des Dokumenten mit der Methodenbeschreibung, mögliche Kommentare zu der Messmethoden, Anzahl der Vor- und Nachkommastellen der Werte der gemessenen Parameter), die der Benutzer dem Programm übermitteln muss, bevor die Datendateien gemäß der Anforderungen in diesem Projektbereich erzeugt werden.

3.2.4 Anforderungen an die Daten für die Archivierung in der Datenbank „SeaIceDB“

Die in dem Projektbereich „SeaIceDB“ erzeugten Dateien müssen ähnliche wie für „PANGAEA“, aber nicht gleiche Struktur aufweisen. Die Struktur von diesen Dateien muss der Struktur dem Projekt entsprechenden Datenbanktabellen gerecht werden. Die Event labels müssen als 1 Wert in jeder Zeile der Metadatendatei stehen. Statt Spalten „campaign“ und „area“ müssen die Metadatendateien die Spalte „campaign_id“ mit entsprechenden Werten haben. Die Datendateien müssen keinen Header besitzen. Fehlen gelegentlich die Werte der untersuchten Parameter, müssen an diesen Stellen die Zeichenketten „NULL“ hinzugefügt werden, um die Dateneinführung zu gewährleisten (die entsprechende Spalten in der Datentabellen, wo die Messwerte gespeichert werden, sind als NULL deklariert).

4 Die Entwicklung

4.1 Pflichtenheft SI_Thickness_DB

4.1.1 Zielbestimmung

4.1.1.1 Muß-Kriterien

- Erstellung der Dateien von den 3 unterschiedlichen Messverfahren für 2 Projektbereiche (“PANGAEA” und “SeaIceDB”);
- Übernahme der Forschungsdaten in die Datenbank “SeaIceDB”;
- Extrahierung der Forschungsdaten aus der Datenbank “SeaIceDB”.

4.1.1.2 Kann-Kriterien

Keine.

4.1.1.3 Abgrenzungskriterium

- keine Interpolation der extrahierten Daten (grafische Darstellung in Form eines Diagramms oder „Plotten“).

4.1.2 Einsatz

4.1.2.1 Anwendungsbereiche

- Forschungsgruppe „Meereseis“ (Sektion Regionale Zirkulation, Fachbereich Klimasystem, Alfred-Wegener-Institut für Polar- und Meeresforschung).

4.1.2.2 Zielgruppen

Der Leiter der Forschungsgruppe „Meereseis“ (oder einfach: Auftraggeber 1).

4.1.2.3 Betriebsbedingungen

Physikalische Umgebung des Softwaresystems entspricht einer gewöhnliche Büroumgebung.

Für die Dateierzeugung und die Übernahme der Forschungsdaten in die Datenbank „SeaIceDB“ wird die Software nach jeder Expedition, zur Extrahieren der Daten aus der Datenbank – nach Bedarf benutzt.

Nach dem Start und bis zum Schluss der Datenübername in die Forschungsdatenbank läuft die Software in einem unbeaufsichtigten Betrieb. Während der Bewältigung anderen Aufgaben muss das Programm durch die Benutzerinteraktion gesteuert werden.

4.1.3 Umgebung

4.1.3.1 Software

Solaris 8.0 oder höher oder anderes UNIX-Derivate;
J2SE v 1.4.2 SDK.

4.1.3.2 Hardware

SUN Ultra5 Workstation (im Fall von der Betriebssystem Solaris 8.0) oder andere Rechner mit Pentium-II und höher oder AMD Duron, Athlon und 256MB RAM.

4.1.3.3 Orgware

Keine.

4.1.4 Funktionalität

Es lassen sich folgende Arbeitsabläufe identifizieren:

- die Dateien von einer Expedition für den Projekt „Ground thickness profiles“ in dem Projektbereich „PANGAEA“ erzeugen;
- die Dateien von einer Expedition für den Projekt „Ground thickness profiles“ in dem Projektbereich „SeaIceDB“ erzeugen;

- die Dateien von einer Expedition für den Projekt „Freeboard profiles“ in dem Projektbereich „PANGAEA“ erzeugen;
- die Dateien von einer Expedition für den Projekt „Freeboard profiles“ in dem Projektbereich „SeaIceDB“ erzeugen;
- die Dateien von einer Expedition für den Projekt „HEM thickness profiles“ in dem Projektbereich „PANGAEA“ erzeugen;
- die Dateien von einer Expedition für den Projekt „HEM thickness profiles“ in dem Projektbereich „SeaIceDB“ erzeugen;
- die Daten von einer Expedition aus der in dem Projektbereich „SeaIceDB“ erzeugten Dateien in die Datenbank „SeaIceDB“ übernehmen;
- in der Datenbank „SeaIceDB“ gespeicherte Daten von der Expeditionen extrahieren und beim Bedarf in der Textdateien speichern.

Das Erfassen, Ändern und Löschen von Wissenschaftler, Parameter und Messmethoden, deren Attribute während der Dateierzeugung benutzt werden, erfolgt mit Hilfe der entsprechenden Funktionen, die in diesem Punkt nicht aufgeführt werden.

4.1.5 Daten

Es sollen die Messdaten von 3 Projekten (Messverfahren) aus der verschiedenen Expeditionen gespeichert werden,
d.h.

ca. $25\,000 * 15 = 375\,000$ Datensätze von dem Projekt „Ground thickness profiles“,
ca. $500\,000 * 15 = 7\,500\,000$ Datensätze von dem Projekt „Freeboard profiles“,
ca. $10\,000\,000 * 15 = 150\,000\,000$ Datensätze von dem Projekt „HEM thickness profiles“
bei einer fünfzehnjährigen Nutzungszeit der Software gespeichert werden.

4.1.6 Leistungen

Leistungen sind hier nicht relevant.

4.1.7 Benutzungsoberfläche

Es wird eine überwiegend funktionsorientierte grafische Oberfläche entsprechend der Gestaltungsvorschriften nach der Europäische Norm ISO 9241-10 : 1996 erstellt.

4.1.8 Qualitätsziele

- hohe Benutzungsfreundlichkeit;
- hohe Änderbarkeit.

4.2 UML-Diagramme

Gemäß allen Anforderungen wurden die Use Case und Aktivitätsdiagramme gezeichnet. Sie befinden sich in dem Anhang B und C.

4.3 Das Datenmodell

4.3.1 Tabelle “scientists”

Attribut	Datentyp	Optionen	Beschreibung
scientist_id	int	primary key, not null	identifiziert den Wissenschaftler eindeutig
scientist	varchar(70)	not null, unique	Name des Wissenschaftlers

Tabelle 12: Datenbanktabelle „scientists“ mit dazugehörigen Attribute

4.3.2 Tabelle “parameters”

Attribut	Datentyp	Optionen	Beschreibung
parameter_id	int	primary key, not null	identifiziert den Parameter eindeutig und ist seine PANGAEA-ID
parameter	varchar(70)	not null, unique	Name des Parameters
short_name	varchar(20)	not null, unique	Parameterkürzel
data_type	varchar(20)	not null	Datentyp des Parameters

Tabelle 13: Datenbanktabelle „parameters“ mit dazugehörigen Attributen

4.3.3 Tabelle “methods”

Attribut	Datentyp	Optionen	Beschreibung
method_id	int	primary key, not null	identifiziert die Messmethode eindeutig und ist ihre PANGAEA-ID
method	varchar(80)	not null, unique	Name der Messmethode
short_name	varchar(20)	not null, unique	Kürzel der Messmethode

Tabelle 14: Datenbanktabelle „methods“ mit dazugehörigen Attributen

4.3.4 Tabelle “measurementPositions”

Attribut	Datentyp	Optionen	Beschreibung
parameter_id	int	primary key, foreign key, not null	zusammengesetzte Primärschlüssel, Fremdschlüssel zur Tabelle „parameters“
method_id	int	primary key, foreign key, not null	zusammengesetzte Primärschlüssel, Fremdschlüssel zur Tabelle „methods“
position	varchar(41)	not null, unique	Name der Messposition, setzt sich aus der Parameterkürzel und der Kürzel der Messmethode (verbunden mit „_“)

Tabelle 15: Datenbanktabelle „measurementPositions“ mit dazugehörigen Attributen

4.3.5 Tabelle “campaigns”

Attribut	Datentyp	Optionen	Beschreibung
campaign_id	int	primary key, not null	identifiziert die Expedition eindeutig
campaign	varchar(20)	not null, unique	Name der Expedition
campaign_begin	datetime	not null	Datum des Expeditionsanfangs
campaign_end	datetime	not null	Datum des Expeditionsendes

Tabelle 16: Datenbanktabelle „campaigns“ mit dazugehörigen Attributen

4.3.6 Tabelle “iceThicknessMetaData”

Attribut	Datentyp	Optionen	Beschreibung
meta_id	int	primary key, not null	identifiziert den Metadatensatz mit diesem Event label eindeutig
event_label	varchar(30)	not null	Name der Eisscholle
campaign_id	int	foreign key, not null	Fremdschlüssel zur Tabelle „campaigns“
gear	varchar(10)	not null	Kurzname des Geräts
date_time	datetime	not null	Datum der Messung
latitude	float	not null	geografische Breite der Eisscholle
longitude	float	not null	geografische Länge der Eisscholle
remark	varchar(1024)	null	Bemerkung zu diesem Event label

Tabelle 17: Datenbanktabelle „iceThicknessMetaData“ mit dazugehörigen Attributen

In der Datenbanktabelle „iceThicknessMetaData“ stellt jeder Datensatz die Metadaten von einer während irgendeiner Expedition vermessene Eisscholle (Messverfahren „Ground thickness profiles“) dar.

4.3.7 Tabelle „iceThicknessData“

Attribut	Datentyp	Optionen	Beschreibung
data_id	int	primary key, not null	identifiziert den Datensatz eindeutig
meta_id	int	foreign key, not null	Fremdschlüssel zur Tabelle „iceThicknessMetaData“
Distance_RT	float	null	Name der Messposition #1
Ice_thickness_EM	float	null	Name der Messposition #2
Ice_thickness_DH	float	null	Name der Messposition #3
Freeboard_DH	float	null	Name der Messposition #4
Draft_EM	float	null	Name der Messposition #5
Draft_DH	float	null	Name der Messposition #6
Snow_thickness_DH	float	null	Name der Messposition #7
Zsl_DH	float	null	Name der Messposition #8
Zse_DH	float	null	Name der Messposition #9
Zmeltpond_DH	float	null	Name der Messposition #10
Zmeltpondbot_DH	float	null	Name der Messposition #11

Tabelle 18: Datenbanktabelle „iceThicknessData“ mit dazugehörigen Attributen

In der Datenbanktabelle „iceThicknessData“ stellt jeder Datensatz die Messwerte von einer während irgendeiner Expedition vermessene Eisscholle (Messverfahren „Ground thickness profiles“) dar. Der Attribut *meta_id* referenziert die Metadaten dieser Eisscholle in der Metadatentabelle für dieses Projekt („iceThicknessMetaData“). Alle Attribute, die die Messpositionen darstellen, sind als *null* deklariert, weil die entsprechende Werte in einer oder anderer Expedition gar nicht erhoben oder mit der andere Messmethode (sehr wahrscheinlich) erhoben werden. Das trifft die Datentabellen in allen Projekten.

Um die Bedeutungen der Messpositionsnummer in dieser Tabelle zu erklären, werden alle zur Zeit in der Datenbanktabellen „parameters“ und „methods“ gespeicherte Parameter und Messmethoden dargestellt.

Die Spaltennamen in dieser Tabelle entsprechen der Attributnamen in der entsprechende Datenbanktabelle.

parameter_id	parameter	short_name	data_type
27313	Device height above ground	HEIGHT	float
26098	Draft	Draft	float
8194	Freeboard	Freeboard	float
14672	Ice coverage	Ice coverage	tinyint
14673	Ice thickness	Ice thickness	float
1600	Latitude	Latitude	float
1601	Longitude	Longitude	float
26174	Melt pond bottom Z coordinate	Zmeltpondbot	float
26173	Melt pond depth	Zmeltpond	float
26097	Measurement point distance	Distance	float
16065	Fid number	Fid	int
5348	Snow thickness	Snow thickness	float
26099	Surface elevation	Zse	float
26228	Surface layer thickness	Surface layer thickness	float

Tabelle 19: Alle Parameter in der Tabelle „parameters“ der Datenbank „SeaIceDB“

method_id	method	short_name
50	Calculated	Calc
4776	Calculated from ice-thickness based on the electromagnetic sounding	Calc from EM
4777	Calculated from ice-thickness measured with ruler tape	Calc from DH
4793	Determined by helicopter-borne laser distance measurement	Heli laser
4792	Determined from reflected laser signal strength	Laser echo
4826	Determined with laser altimeter on EM Bird	Laser
4762	Drilling and ruler tape	DH
4763	Electromagnetic sounding (EM), Geonics EM31 conductivity sensor, Haas et al 1997	EM
4821	Measured from helicopter-borne EM sounding	HEM
43	Not given	Ng
4760	Ruler stick	RS
4761	Surface levelling/surveying	Surveying
5024	Ruler tape	RT
202	GPS	GPS
5025	Calculated from GPS	Calc from GPS

Tabelle 20: Alle Messmethoden in der Tabelle „methods“ der Datenbank „SeaIceDB“

Die nachfolgende Tabelle stellt die Namen der Messpositionen in der Tabelle „iceThicknessData“, in dieser Messpositionen untersuchte Parameter und die Messmethode, mit deren die Werte von diesem Parameter in dieser Messposition erhoben wurden, dar.

Name der Messposition	Name des Parameters	Name der Messmethode
Distance_RT	Measurement point distance	Ruler tape
Ice_thickness_EM	Ice thickness	Electromagnetic sounding (EM), Geonics EM31 conductivity sensor, Haas et al 1997
Ice_thickness_DH	Ice thickness	Drilling and ruler tape
Freeboard_DH	Freeboard	Drilling and ruler tape
Draft_EM	Draft	Electromagnetic sounding (EM), Geonics EM31 conductivity sensor, Haas et al 1997
Draft_DH	Draft	Drilling and ruler tape
Snow_thickness_DH	Snow thickness	Drilling and ruler tape
Zsl_DH	Surface layer thickness	Drilling and ruler tape
Zse_DH	Surface elevation	Drilling and ruler tape
Zmeltpond_DH	Melt pond depth	Drilling and ruler tape
Zmeltpondbot_DH	Melt pond bottom Z coordinate	Drilling and ruler tape

Tabelle 21: Messpositionen in der Datenbanktabelle „iceThicknessData“

4.3.8 Tabelle „laserMetaData“

Attribut	Datentyp	Optionen	Beschreibung
meta_id	int	primary key, not null	identifiziert den Metadatensatz mit diesem Event label eindeutig
event_label	varchar(30)	not null	Name der Messung
campaign_id	int	foreign key, not null	Fremdschlüssel zur Tabelle „campaigns“
gear	varchar(10)	not null	Kurzname des Geräts
start_date_time	datetime	not null	Datum und Uhrzeit des Messungsanfangs
start_latitude	float	not null	geografische Breite des Messungsanfangs
start_longitude	float	not null	geografische Länge des Messungsanfangs
end_date_time	datetime	not null	Datum und Uhrzeit des Messungsendes
end_latitude	float	not null	geografische Breite des Messungsendes
end_longitude	float	not null	geografische Länge des Messungsendes
remark	varchar(1024)	null	Bemerkung zu dieser Messung

Tabelle 22: Datenbanktabelle „laserMetaData“ mit dazugehörigen Attributen

In der Datenbanktabelle „laserMetaData“ stellt jeder Datensatz die Metadaten von einer während irgendeiner Expedition durchgeführte Messung (Messverfahren „Freeboard profiles“) dar. Gemäß Online-Dokumentation zu der DBMS ASE 12.5 auf der Internetseite <http://sybooks.sybase.com> darf maximal 1 Spalte in der Datenbanktabelle als Datentyp „timestamp“ deklariert werden. In diesem Fall haben wir 2 Spalten mit der Datums- und Uhrzeitangaben und benutzen deswegen den Datentyp „datetime“.

4.3.9 Tabelle „laserData“

Attribut	Datentyp	Optionen	Beschreibung
data_id	int	primary key, not null	identifiziert den Datensatz eindeutig
meta_id	int	foreign key, not null	Fremdschlüssel zur Tabelle „laserMetaData“
Distance_Calc from GPS	float	null	Name der Messposition #1
Freeboard_Heli_laser	float	null	Name der Messposition #2
Ice_coverage_Laser_echo	tinyint	null	Name der Messposition #3

Tabelle 23: Datenbanktabelle „laserData“ mit dazugehörigen Attributen

In der Datenbanktabelle „laserData“ stellt jeder Datensatz die Messwerte von einer während irgendeiner Expedition durchgeführte Messung (Messverfahren „Freeboard profiles“) dar. Der Attribut *meta_id* referenziert die Metadaten dieser Messung in der Metadatentabelle für dieses Projekt („laserMetaData“). Obwohl die Werte des Parameters „Ice coverage“ keine Einheit besitzen und nur „0“ oder „1“ sein können, wird deren Datentyp nicht als zu definierendes Domain *boolean* deklariert. Diese Tatsache hat 2 Gründe:

1. nach der Aussage des Auftraggebers 1 können die Werte der Messposition mit diesem Parameter in Zukunft bei ganzen Expeditionen fehlen (wäre dabei der Datentyp von der boolische Domain und stünde „0“ für fehlende Werte, könnte es als „kein Eis, sondern Wasser“ interpretieren, was sicherlich falsch wäre);
2. in der oben genannte Online-Dokumentation ist es empfohlen statt selbst deklarierte Domains die Standard-Datentypen zu verwenden.

Die nachfolgende Tabelle stellt die Namen der Messpositionen in der Tabelle „laserData“, in dieser Messpositionen untersuchte Parameter und die Messmethode, mit deren die Werte von diesem Parameter in dieser Messposition erhoben wurden, dar.

Name der Messposition	Name des Parameters	Name der Messmethode
Distance_Calc_from_GPS	Measurement point distance	Calculated from GPS
Freeboard_Heli_laser	Freeboard	Determined by helicopter-borne laser distance measurement
Ice_coverage_Laser_echo	Ice coverage	Determined from reflected laser signal strength

Tabelle 24: Messpositionen in der Datenbanktabelle „laserData“

4.3.10 Tabelle "iceThickLaserMetaData"

Attribut	Datentyp	Optionen	Beschreibung
meta_id	int	primary key, not null	identifiziert den Metadatensatz mit diesem Event label eindeutig
event_label	varchar(30)	not null	Name der Messung
campaign_id	int	foreign key, not null	Fremdschlüssel zur Tabelle „campaigns“
gear	varchar(10)	not null	Kurzname des Geräts
date_time	datetime	not null	Datum und Uhrzeit des Messungsanfangs
start_latitude	float	not null	geografische Breite des Messungsanfangs
start_longitude	float	not null	geografische Länge des Messungsanfangs
end_latitude	float	not null	geografische Breite des Messungsendes
end_longitude	float	not null	geografische Länge des Messungsendes
remark	varchar(1024)	null	Bemerkung zu dieser Messung

Tabelle 25: Datenbanktabelle „iceThickLaserMetaData“ mit dazugehörigen Attributen

In der Datenbanktabelle „iceThickLaserMetaData“ stellt jeder Datensatz die Metadaten von einer während irgendeiner Expedition durchgeführte Messung (Messverfahren „HEM thickness profiles“) dar.

4.3.11 Tabelle “iceThickLaserData”

Attribut	Datentyp	Optionen	Beschreibung
data_id	int	primary key, not null	identifiziert den Datensatz eindeutig
meta_id	int	foreign key, not null	Fremdschlüssel zur Tabelle „iceThickLaserMetaData“
Latitude_GPS	float	null	Name der Messposition #1
Longitude_GPS	float	null	Name der Messposition #2
Distance_Calc_from_GPS	float	null	Name der Messposition #3
Fid_Ng	int	null	Name der Messposition #4
Ice_thickness_HEM	float	null	Name der Messposition #5
HEIGHT_Laser	float	null	Name der Messposition #6

Tabelle 26: Datenbanktabelle „iceThickLaserData“ mit dazugehörigen Attributen

Die nachfolgende Tabelle stellt die Namen der Messpositionen in der Tabelle „iceThickLaserData“, in dieser Messpositionen untersuchte Parameter und die Messmethode, mit deren die Werte von diesem Parameter in dieser Messposition erhoben wurden, dar.

Name der Messposition	Name des Parameters	Name der Messmethode
Latitude_GPS	Latitude	GPS
Longitude_GPS	Longitude	GPS
Distance_Calc_from_GPS	Measurement point distance	Calculated from GPS
Fid_Ng	Fid number	Not given
Ice_thickness_HEM	Ice thickness	Measured from helicopter-borne EM sounding
HEIGHT_Laser	Device height above ground	Determined with laser altimeter on EM Bird

Tabelle 27: Messpositionen in der Datenbanktabelle „iceThickLaserData“

Weil gemäß der Aussage des Auftraggebers 1 die Werte von allen Parametern (mit der Ausnahme von "Ice coverage") in Meter gemessen werden, wurde auf die Entität und die Datenbanktabelle "measurementUnits" verzichtet.

4.3.12 ER-Diagramm

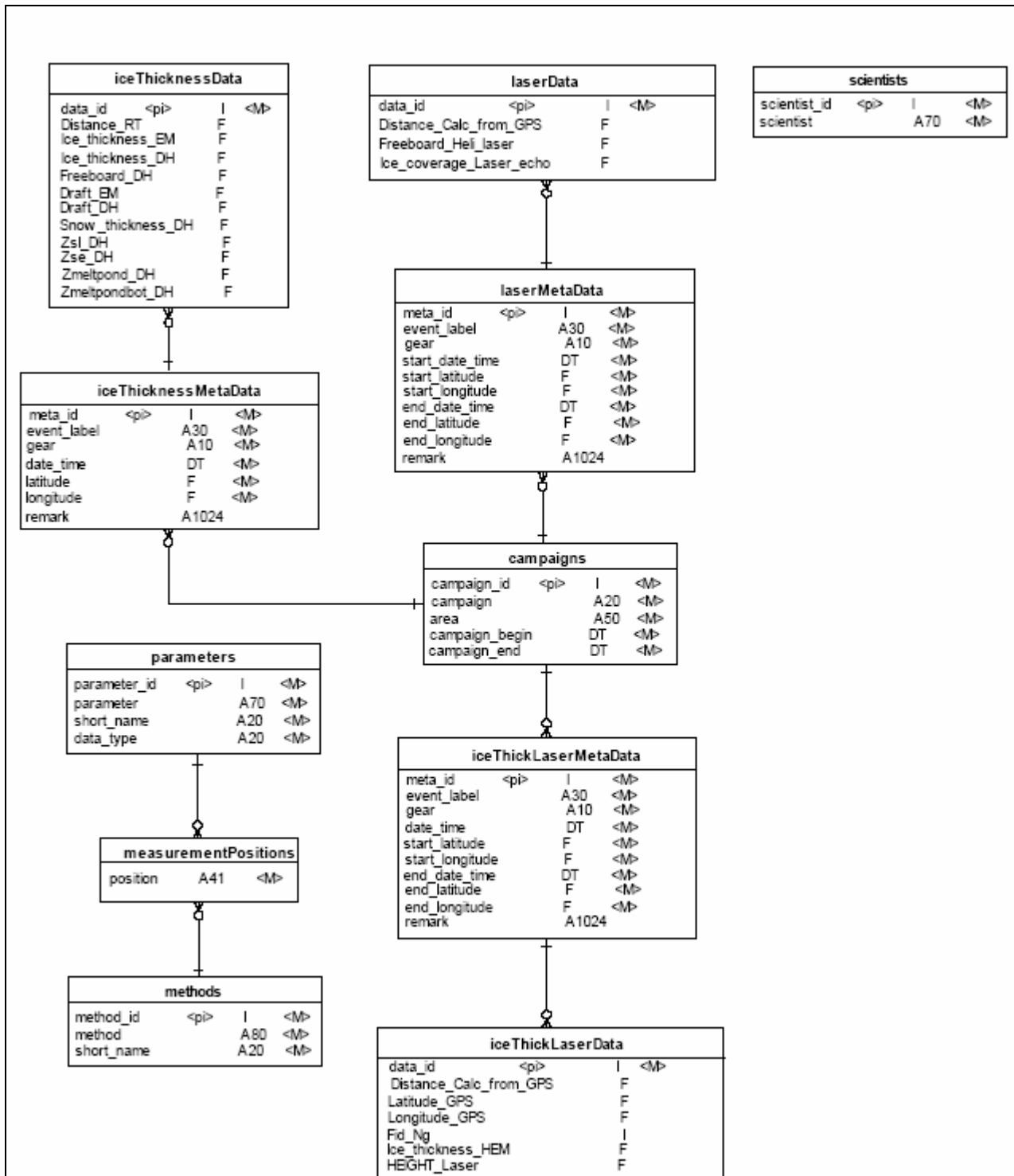


Diagramm 1: ER-Diagramm

4.3.13 Relationales Datenmodell

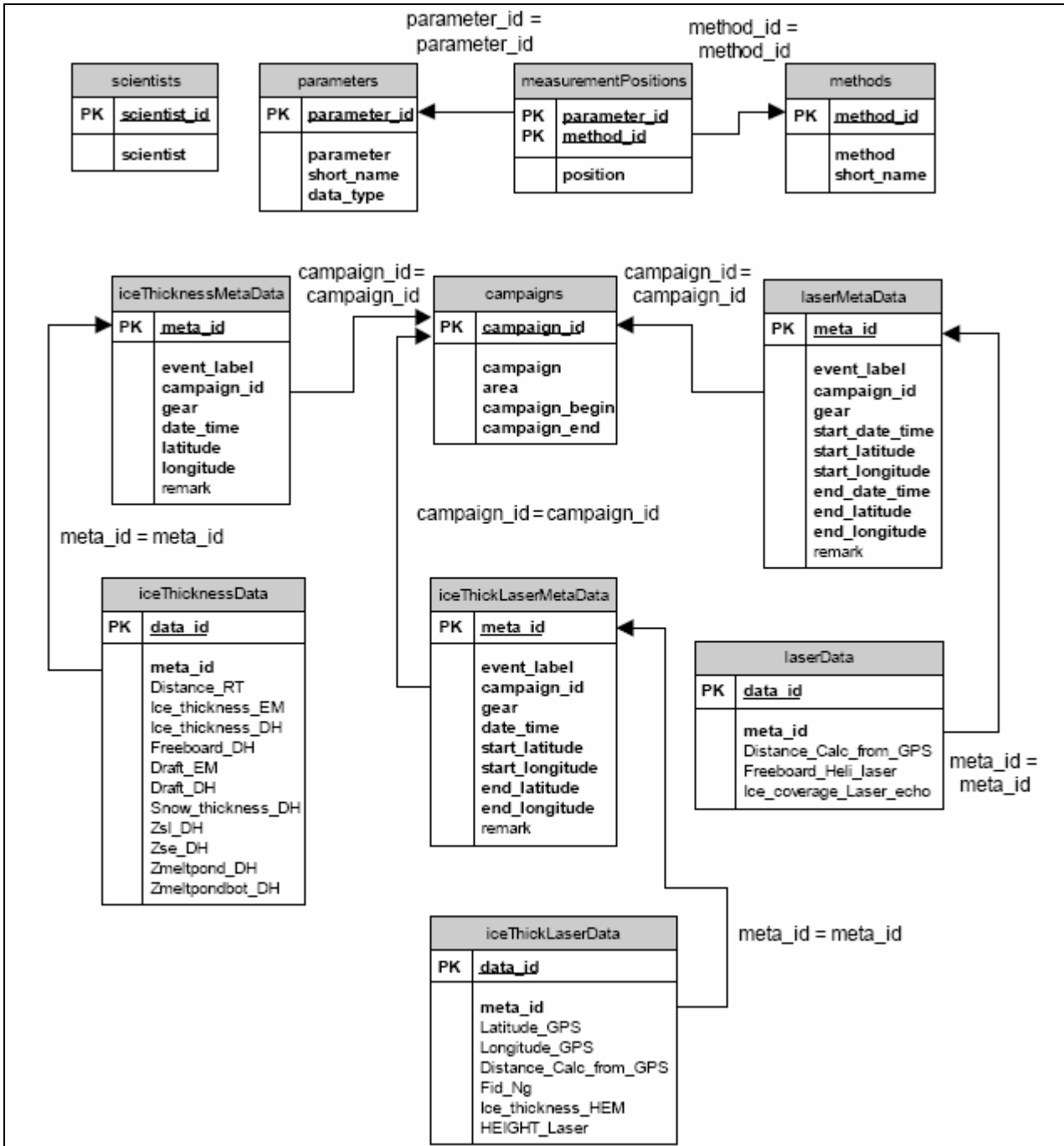


Diagramm 2: Relationales Datenmodell

Die während des Praxissemesters erstellte Datenmodell, die nur 6 Tabellen für 3 Projekte beinhaltet (3 Metadaten- und 3 Datentabellen, die keine Informationen von der benutzten Messmethoden hatten), stellte sich als überholt heraus und wurde aus der Datenbank entfernt. Das SQL-Skript zur Erzeugung der neuen Tabellen befindet sich in dem Anhang H.

4.4 Das Klassendesign

Die Entwicklung wurde in der Programmiersprache Java durchgeführt. Alle Elemente der grafische Benutzungsoberfläche sind die Instanzen der Klassen aus dem Paket `javax.swing` oder deren Unterklassen. Die während der Benutzung der Software ausgelöste Aktivitäten sind die Ereignisse aus der Paketen `java.awt.event` und `javax.swing.event`.

Das Programm besteht aus 25 Klassen, die nach dem logischen Zusammenhang den 3 Paketen zugeordnet sind. In dem Anhang D befinden sich die Klassendiagramme, die die Struktur und die Beziehungen zwischen allen Klassen im System darstellen.

4.4.1 Paket `FilesProcessing`

Das Paket `FilesProcessing` besteht aus folgenden Klassen:

- `SI_Thickness_DB`,
- `HeaderDialog`
- `AttributsOfChosenParameters`,
- `Profiles`,
- `Ground_Thickness_Profiles`,
- `Freeboard_Profiles`,
- `HEM_Thickness_Profiles`,
- `PANGAEA_Ground_Thickness_Profiles`,
- `SeaIceDB_Ground_Thickness_Profiles`,
- `PANGAEA_Freeboard_Profiles`,
- `SeaIceDB_Freeboard_Profiles`,
- `PANGAEA_HEM_Thickness_Profiles`,
- `SeaIceDB_HEM_Thickness_Profiles`,
- `TextFieldDocument`.

Die Klasse `SI_Thickness_DB` ist der Startpunkt und das Rahmen des Programms. Sie wurde aus der Klasse `JFrame` abgeleitet. Die Instanz dieser Klasse stellt das Anwendungsfenster dar, in dem der Benutzer für eine aus der Aktivitäten "Dateien erzeugen", "Daten in "SeaIceDB" einfüegen", "Daten in "SeaIceDB" anzeigen" entscheidet. Die Aktionen werden entweder durch die Auswahl des entsprechenden Menüeintrags in oberen linken Bereich des Fensters oder durch die Betätigung der Schaltfläche in der Leiste im unteren Fensterbereich ausgelöst. Die Menüeinträge besitzen die Tastatur- und mnemonische Kürzel.

Die Interaktionselemente für die Sprach-, Projektbereich- und Projektauswahl sind die Auswahlfelder, die durch die Instanzen der Klasse `JComboBox` realisiert und in dem oberen Bereich des Fensters platziert sind. Entscheidet sich der Benutzer die Dateien für eine Expedition aus vorliegenden Quellendateien zu erzeugen, muss er diese Expedition erfassen (Tabelle „campaigns“ in der Datenbank „SeaIceDB“). Falls die Dateien erzeugt werden sollen, gibt der Benutzer die Angaben für die zu erzeugende Metadatendatei ein und wählt die Verzeichnisse von Quellen- und für die zu erzeugende Dateien mittels der Schltflächen aus. Die Verzeichnisauswahl wird in einem modalen Dialog, der von der Instanz der Klasse `JFileChooser` dargestellt wird, durchgeführt. Die vollständige Pfade der ausgewählten Verzeichnissen werden in der entsprechenden einzeiligen Textbereichen dargestellt und können unter der Benutzung der Schieberegler kontrolliert werden.

Zu allen Eingabefelder in diesem Fenster gibt es die Hilfetexte, die in einem Textbereich (nur für die Anzeige konzipiert und durch die Instanz der Klasse `JTextArea` implementiert) rechts oben dargestellt wird, soweit der Benutzer mit der linke Maustaste den Cursor in den entsprechende Eingabefeld setzt.

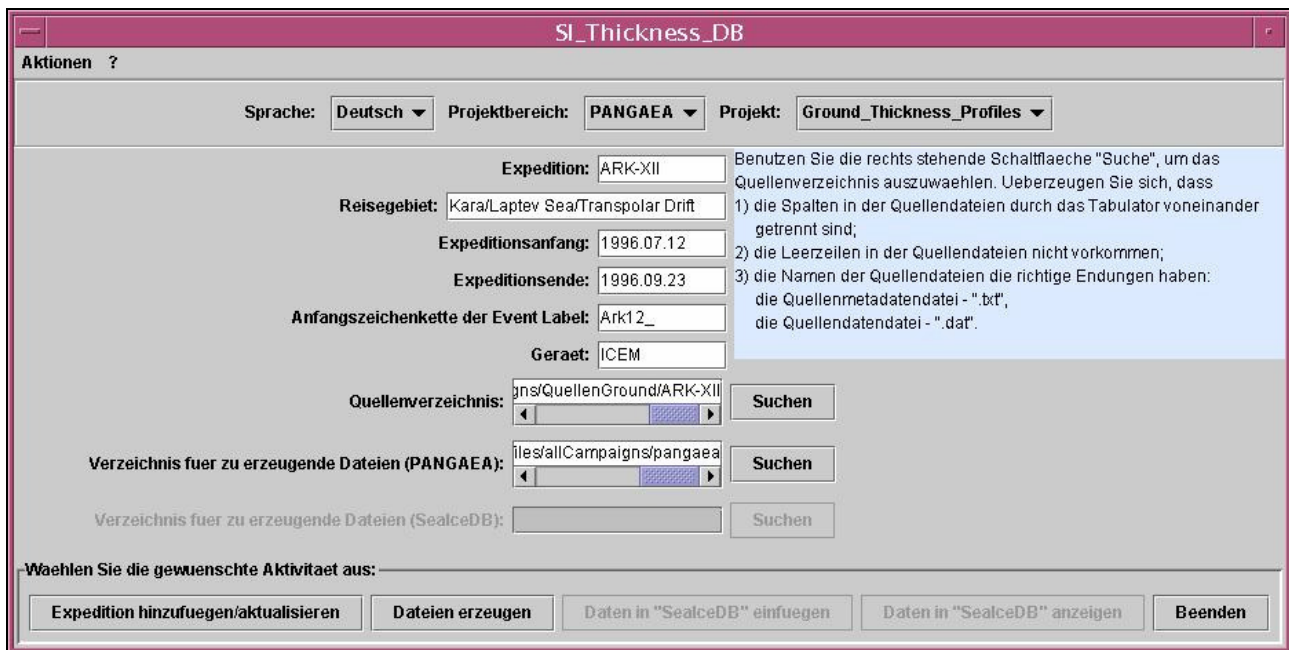


Abbildung 12: Startfenster des Programms

Der Benutzer wird die Expeditionsdaten in der Datenbank erfassen müssen, falls er die Dateien für den Projektbereich „SeaIceDB“ erzeugen möchte. Ausserdem werden alle unten dargestellten Schaltflächen in diesem Projektbereich aktiv.

Die Klasse *HeaderDialog* erzeugt den modalen Dialog für die Parameterauswahl und ist die Unterklasse von *JDialog*. Das von der Instanz dieser Klasse dargestellte Dialogfenster hängt in seinem Layout von dem Zustand des Objektes der Klasse *SI_Thickness_DB*. Falls der Benutzer in dem von dem Objekt der Klasse *SI_Thickness_DB* dargestellten Anwendungsfenster den Projektbereich PANGAEA auswählt, werden die Interaktionselemente für die Informationseingabe zum Aufbau des Headers von der zu erzeugenden Datendateien im oberen Bereich des Dialogfensters dargestellt. Das sind die Auswahlfelder (Instanzen der Klasse *JComboBox*) für die Auswahl von dem Projektleiter und der Autoren des Zitats zu dem jeweiligen Datensatz (sehen Sie die Erklärung zum Header auf der Seite 27), der einzeilige Textbereich (Instanz der Klasse *JTextArea*) für den Titel des Datensatzes (wird bei der Initialisierung mit dem voreingestellten Wert belegt, kann aber jederzeit editiert werden), die Textfelder für Eingabe des Namens des Dokumentes mit der Methodebeschreibung (falls ein solche Dokument für die Expedition nicht existiert, bleibt das entsprechende Eingabefeld leer) und der PANGAEA-IDs der wissenschaftlichen Referenzen zu der Expedition und das Drehfeld (engl. spin box, Instanz der Klasse *JSpinner*, mit der Einträgen von „1“ bis „8“) für die Auswahl von der Anzahl der Zitatsautoren. Die Einträge in der Auswahlfelder stellen die Namen aller Wissenschaftler in der Datenbanktabelle „scientists“ und werden bei der Initialisierung des Dialogfensters aus der Datenbank extrahiert. Alle Auswahlfelder haben den voreingestellten Eintrag „Christian Haas“, weil meistens seinen Namen ausgewählt werden muss. Ist der gesuchte Name des Wissenschaftlers nicht in der Liste, dann sind die Daten von diesem Wissenschaftler nicht in der Tabelle „scientists“ und müssen erst erfasst werden. Der ausgewählte Eintrag in dem Drehfeld bestimmt die Anzahl der angezeigten Auswahlfelder für die Auswahl der Zitatsautoren.

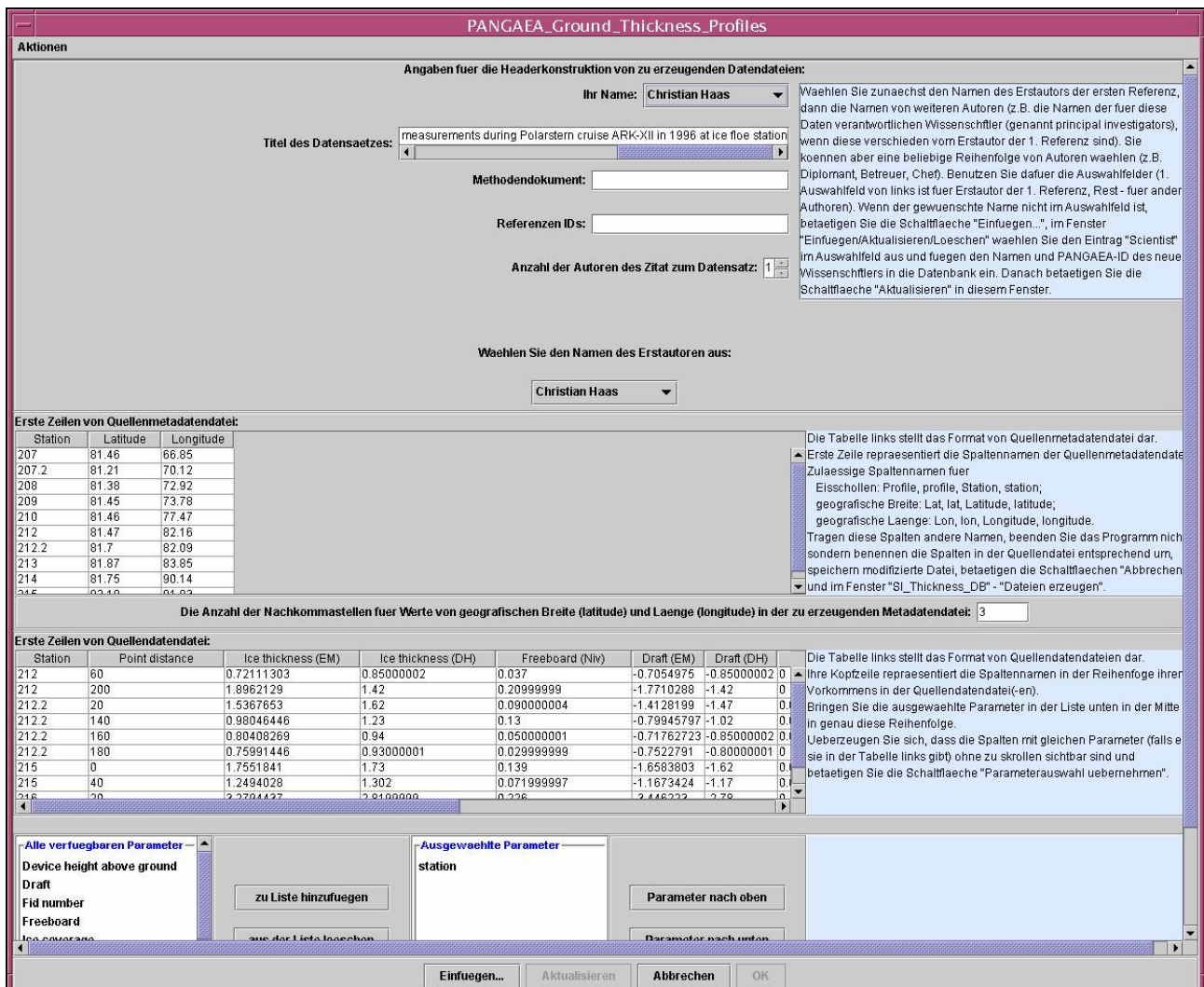


Abbildung 13: Dialog für die Parameterauswahl in dem Projektbereich „PANGAEA“

Wählt der Benutzer in dem Startfenster des Programms den Projektbereich „SeaIceDB“, wird den oben beschriebene Bereich gar nicht dargestellt, weil die für diesen Bereich erzeugte Dateien keinen Header besitzen müssen.

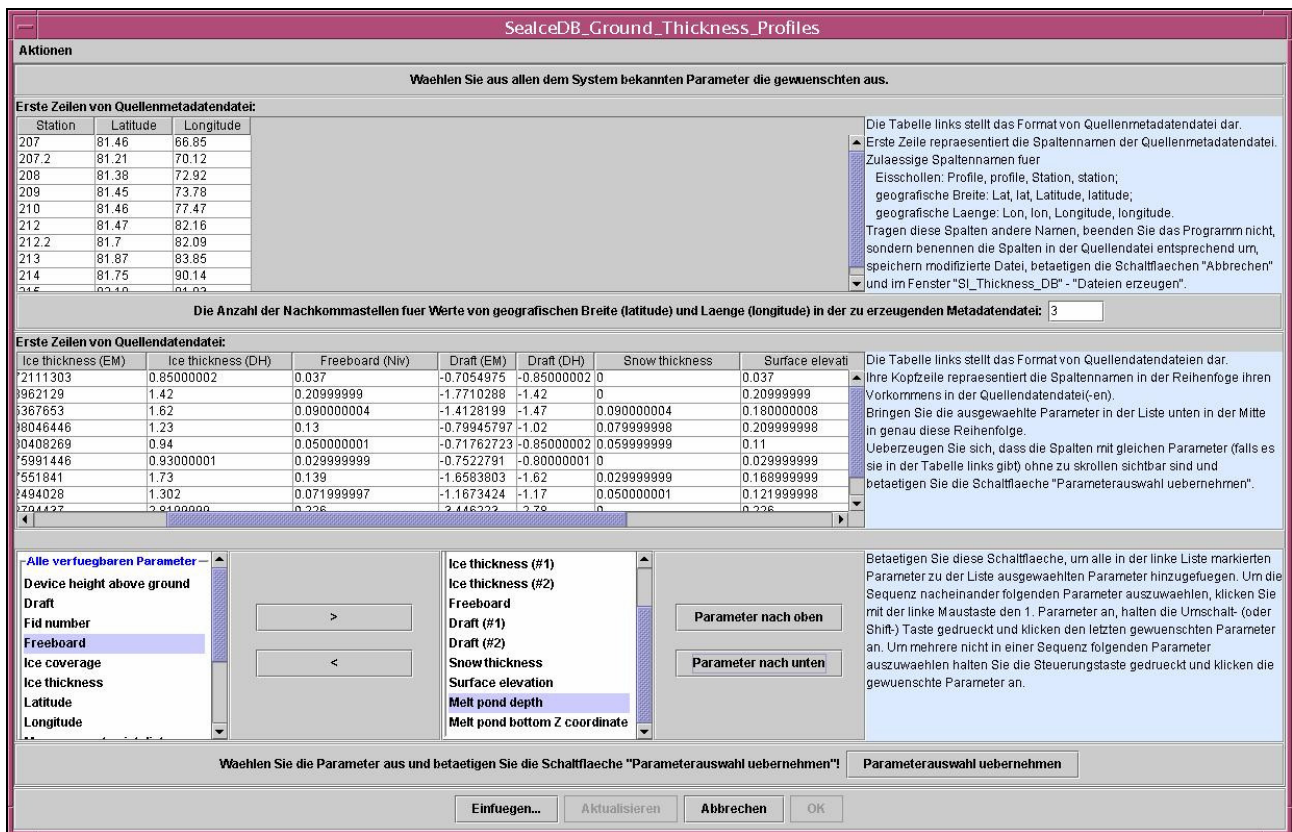


Abbildung 14: Dialog für die Parameterauswahl in dem Projektbereich „SeaIceDB”

Nachfolgende Komponenten des Dialogfensters sind für beide Projektbereiche gleich. Bei der ausgewählten Projekt “Ground thickness profiles” wird ein Bereich mit der Hilfstabelle zu dem Spaltenformat der Quellenmetadatendatei, die dafür während der Erzeugung von diesem Dialog eingelesen wird, und der dazugehörige Beschreibung angezeigt. Der Benutzer muss die angezeigten Spaltennamen in dieser Tabelle mit der in der Beschreibung ausgewiesenen Musternamen (3 möglichen Muster für jeden Namen) vergleichen. Stimmt mindestens ein Spaltennamen in der Hilfstabelle mit keinem aus 3 vorgegebenen Musternamen überein, muss der Benutzer die Namen der Spalten in der Quellenmetadatendatei berichtigen, die modifizierte Quellendatei speichern, dieses Dialogfenster durch die Betätigung der Schaltfläche “Abbrechen” schliessen und in dem Startfenster die Schaltfläche “Dateien erzeugen” wieder betätigen. Dabei werden die Quellendateien wieder eingelesen und der Abschnitt aus der Quellenmetadatendatei in dieser Tabelle erneut angezeigt. Oben geschilderten Anweisungen sind auch in der Beschreibung zu der Hilfstabelle dargestellt. Bei 2 anderen Projekten fehlt dieser Bereich, weil

- die Quellenmetadatendateien aus dem Projekt “Freeboard profiles” keinen tabellarischen Aufbau aufweisen,
- es keine Quellenmetadatendateien in dem Projekt “HEM thickness profiles” gibt.

Danach folgt das Eingabefeld für die Anzahl der Nachkommastellen für Werten von geografischen Breite und Länge in der zu erzeugenden Metadatendatei, das gemäss dem ausgewählten Projekt mit voreingestellten Wert gefüllt wird.

In der Mitte des Dialogfensters wird ein Bereich mit der Hilfstabelle zu dem Spaltenformat der Quellendatei(en) und der dazugehörige Beschreibung angezeigt. Aus der Tatsache, dass die Spaltennamen in der Quellendateien jedes Mal anders benannt werden, liess sich keinen Spaltennamenmuster für die Quellendateien erstellen. Das 1. Zweck dieser Hilfstabelle ist dem Benutzer die Spaltennamen zu veranschaulichen und den Parameterauswahl zu erleichtern, ohne dass man das Dialogfenster verlassen und die Quellendatei aufmachen muss.

Die Hilfstabellen (die Instanzen der Klasse *JTable*) werden nur dann angezeigt, wenn der Format der Quellendateien mit dem für diesen Projekt vorgegebenen Format übereinstimmt (z. B. sollen die Werte in der Zeilen von Quellendateien in dem Projekt "Freeboard profiles" durch das Leerzeichen voneinander getrennt sein, in den anderen Projekten aber durch den Tabulator). Werden die Quellendateien aus dem Projekt "Freeboard profiles" für die Dateierzeugung in dem Projekt "Ground thickness profiles" oder "HEM thickness profiles" ausgewählt, werden statt Hilfstabellen die Textbereiche mit der Fehlermeldungen angezeigt und somit die Fehlbedienung des Programms auf dem Frühstadium zu erkennen gibt.

Unten folgt den Bereich für die Parameterauswahl. Bei der Initialisierung des Dialogsfensters werden alle Parameternamen aus der Datenbanktabelle "parameters" extrahiert und in der Auswahlliste, die Auswahl von mehreren Einträgen erlaubt, unter den Überschrift "Alle verfügbaren Parameter" angezeigt. In einer Reihe mit dieser Liste befindet sich die Liste für ausgewählten Parameter. Mittels der zwischen beiden Listen platzierten Schaltflächen wählt man einen oder mehreren ausgewählten Parameter oder löscht die Parameter aus der Liste "Ausgewählte Parameter", falls der Parameter aus Versehen ausgewählt wurde. Rechts von der Liste der ausgewählten Parameter befinden sich die Schaltflächen, mit deren man die Reihenfolge der ausgewählten Parameter in der entsprechende Liste gemäss der Reihenfolge der Spaltennamen in der Hilfstabelle zu Quellendatendateien anpassen kann.

Ist der Name eines Parameters nicht in der Liste von allen Parameter, ist er nicht in der Datenbanktabelle "parameters" und muss erst erfasst werden. Ist der Name des Parameters in der Liste falsch, muss den Namen berichtigt werden. Dasselbe trifft auch die Wissenschaftlersdaten. Um es zu erreichen, betätigt man entweder die Schaltfläche "Einfügen..." oder wählt man den entsprechenden Menüeintrag aus. Durch diesen Ereignis wird die Instanz der Klasse *PSMImporter* initialisiert und das modale Dialogfenster angezeigt. Nähere Beschreibung zu dieser Klasse wird unten in dem Abschnitt für das Paket Importing dargestellt.

Betätigt man nach dem Parameterauswahl die Schaltfläche "Parameterauswahl uebernehmen" oder wählt man den entsprechenden Menüeintrag aus, wird erst die Überprüfung der Anzahl der ausgewählten Parameter gestartet. Stimmt sie mit der Anzahl der Parameterspalten in der Hilfstabelle zur Quellendatendateien überein (2. Zweck dieser Hilfstabelle), wird die Instanz der Klasse *AttributsOfChosenParameters* initialisiert. Bei dem Projekt "Ground thickness profiles" beinhalten die Quelledatendateien zusätzlich die Spalte "station" (oder wie auch immer der Benutzer diese Spalte bei der Erzeugung der Quellendatendateien während der Expedition nennen wird), die die Namen der untersuchten Eisschollen (z.B. 221, 222, 222.2 usw.) beinhalten. Es ist keine Spalte mit der Parameterwerten (wird sie aber in der Zeile aus der Quellendatendatei leer, so ist es ungültige Datenzeile, was auch in der Algorithmen während Datendateienerzeugung in den Klassen *PANGAEA_Ground_Thickness_Profiles* und *SeaIceDB_Ground_Thickness_Profiles* überprüft wird), und diese Tatsache muss bei der Prüfung berücksichtigt werden. Ausserdem muss in diesem Projekt das String-Array mit der Spaltennamen von der Quellenmetadatendatei erzeugt werden. Es passiert nur dann, wenn diese Spaltennamen mit dem vorgegebenen Muster bereinstimmen.

Nach der richtig ausgewählten Parameteranzahl und der Betätigung der Schaltfläche „Parameterauswahl uebernehmen“ wird die Instanz der Klasse *AttributesOfChosenParameters* initialisiert und von ihr erzeugtes Fenster angezeigt.

Instanz der Klasse *AttributsOfChosenParameters* stellt den modalen Dialog für die Benutzereingaben zur Attributen von der ausgewählten Parameter dar. Der Dialog-Aufrufer (das Fenster für die Parmeterauswahl) bleibt sichtbar, aber unaktiv. Das Layout des neuen Dialogfensters hängt von dem im Startfenster ausgewählten Projektbereich und der Anzahl der ausgewählten Parameter ab.

In dem Projektbereich “SeaIceDB” gibt es 2 Attribute zu jedem ausgewählten Parameter:

- die Anzahl der Nachkommastellen von dessen Werten in der zu erzeugenden Datendateien,
- der Messmethoden, mit deren Werten während der Expedition erhoben wurde.

Abbildung 15: Dialog für die Attribute von der ausgewählten Parameter in dem Projektbereich „SeaIceDB”

In dem Projektbereich “PANGAEA” außer oben genannten Attributen für jeden ausgewählten Parameter kommen noch zusätzliche hinzu:

- die Anzahl der Vorkommastellen von dessen Werten (bilden zusammen mit der Nachkommastellenanzahl das Format der Wertanzeige);
- der Namen des verantwortlichen Wissenschaftlers;
- der Kommentar (optional).

Abbildung 16: Dialog für die Attribute von der ausgewählten Parameter in dem Projektbereich „PANGAEA”

Die Interaktionselemente zur Wissenschaftler- und Messmethodenauswahl stellen die Auswahlfelder dar, deren Einträge während der Initialisieren der Instanz dieser Klasse aus der Datenbank extrahiert werden. Für jeden ausgewählten Parameter werden der verantwortliche Wissenschaftler und die Messmethode voreingestellt, wobei der Benutzer die Auswahl auch ändern kann. Fehlen auch hier die Wissenschaftler oder die Messmethoden, erzeugt man durch die Betätigung der Schaltfläche “Einfügen...” die Instanz der Klasse *PSMImporter* und erfasst die gewünschte Einträge in der entsprechenden Datenbanktabellen. Danach durch die Betätigung der Schaltfläche “Aktualisieren” werden die Einträge in der entsprechenden Auswahlfelder aktualisiert.

Restliche Eingabeelementen sind die Eingabefelder.

Eingaben zu allen Attributen (ausser der Nachkommastellenanzahl) in dem Projektbereich “PANGAEA” werden nur zwecks den Aufbau des Headers von zu erzeugenden Datendateien betätigt. Gemäss der eingegebene Nachkommastellenanzahl werden entsprechende Parameterwerte während der Dateierzeugung abgerundet.

In dem Projektbereich “SeaIceDB” bedeutet die Auswahl der Messmethoden von der ausgewählten Parametern viel mehr. Wie schon erwähnt wurde, bilden die Kurznamen des Parameters und der entsprechenden Messmethode zusammen den Namen der Messposition. Diesen Namen hat die Spalte in der DB-Datentabelle, wo die Werte von diesem Parameter gespeichert werden. Wählt der Benutzer z.B. für den Parameter „Freeboard” (Kurzname auch „Freeboard”) nicht die gewöhnliche Messmethode „Drilling and ruler tape”, sondern die Methode „Ruler tape” (die den Kurznamen „RT”) besitzt, wird an dieser Stelle nach der Bestätigung des Benutzers die neue Messposition gebildet („Freeboard_RT”), die erst in der Datenbanktabelle „measurementPositions” als neuer Datensatz gespeichert wird. Danach wird die neue Spalte mit dem Namen von dieser Messposition und dem Datentyp von dem Parameter „Freeboard” in der diesem Projekt entsprechende Datentabelle angelegt. Ab jetzt werden die Werte des Parameters „Freeboard” in dieser Spalte gespeichert, falls sie mit der Messmethode „Ruler tape” erhoben wurden. Werden die gleiche Parameter ausgewählt (wie in dem Fall auf der Abbildung 16, z.B. „Draft”), müssen deren Messmethoden in dem Projektbereich „SeaIceDB” unterscheiden; in dem Projektbereich „PANGAEA” können die Messmethoden auch gleich sein, dazu müssen sich zu jedem mehrfach ausgewählten Parameter ein Kommentar in dem entsprechenden Eingabefeld stehen. Dabei dürfen die Kommentare nicht gleich sein.

Nach der Eingabe und Auswahl von der Parameterattributen werden sie von der Instanz der Klasse *AttributsOfChosenParameters* auf die Gültigkeit in der Methode `actionPerformed(ActionEvent event)` überprüft, nach dem der Benutzer die Schaltfläche “OK” betätigt. Der Quellcodeabschnitt aus der entsprechende Methode ist über 9 Seiten lang. Aus diesem Grund wurde auf die Bereitstellung der Quellcodeabschnitten in dem Anhang verzichtet.

Werden die Benutzereingaben in Ordnung, gelangt man zu dem Dialogfenster auf der Abbildung 13 oder 14 (abhängig von dem ausgewählten Projektbereich) . Hier betätigt man auch die Schaltfläche “OK”. In der Methode `actionPerformed(ActionEvent event)` der Klasse *HeaderDialog* werden im Fall des Projektbereichs “PANGAEA” die PANGAEA-Ids von allen ausgewählten Parameter, Wissenschaftler und Messmethoden ermittelt und an die Instanz der dem Projekt entsprechende Profiles-Klasse weitergereicht. Im Fall des Projektbereichs “SeaIceDB” werden anhand der ausgewählten Parameter und deren Messmethoden überprüft, ob die neue Messpositionsspalten in der entsprechende Datentabelle entstehen sollen; auch die nicht ausgewählte Messpositionen aus dieser Tabelle werden ermittelt. In der entsprechende Profiles-Klasse werden diese Werte mit “NULL”s gefüllt.

Erzeugt man die Dateien in dem Bereich “PANGAEA”, und bleiben die Anforderungen der Auftraggebers gleich (z.B. keine Unstimmigkeiten über der Anzahl der Nachkommastellen oder keine gleichen Messmethoden bei der mehrfach ausgewählten Parameter), werden die Dateien in dem Projektbereich “SeaIceDB” durch die Betätigung von 2 Schaltflächen durchgeführt, ohne die Parameter und deren Attribute noch ein Mal auswählen zu müssen.

Die Klasse *Profiles* ist die Superklasse fuer alle ...-*Profiles*-Klassen.

Die Klasse *Ground_Thickness_Profiles* ist die Superklasse von der Klassen *PANGAEA_Ground_Thickness_Profiles* und *SeaIceDB_Ground_Thickness_Profiles*.

Die Klasse *Freeboard_Profiles* ist die Superklasse von der Klassen *PANGAEA_Freeboard_Profiles* und *SeaIceDB_Freeboard_Profiles*.

Diese Klasse *HEM_Thickness_Profiles* ist die Superklasse von der Klassen *PANGAEA_HEM_Thickness_Profiles* und *SeaIceDB_HEM_Thickness_Profiles*.

Die Klasse *PANGAEA_Ground_Thickness_Profiles* erzeugt die Metadaten- und Datendateien für das Projekt “Ground_Thickness_Profiles” in dem Projektbereich “PANGAEA”.

Die Klasse *SeaIceDB_Ground_Thickness_Profiles* erzeugt die Metadaten- und Datendateien für das Projekt “Ground_Thickness_Profiles” in dem Projektbereich “SeaIceDB”.

Die Klasse *PANGAEA_Freeboard_Profiles* erzeugt die Metadaten- und Datendateien für das Projekt “Freeboard_Profiles” in dem Projektbereich “PANGAEA”.

Die Klasse *SeaIceDB_Freeboard_Profiles* erzeugt die Metadaten- und Datendateien für das Projekt “Freeboard_Profiles” in dem Projektbereich “SeaIceDB”.

Die Klasse *PANGAEA_HEM_Thickness_Profiles* erzeugt die Metadaten- und Datendateien für das Projekt “HEM_Thickness_Profiles” in dem Projektbereich “PANGAEA”.

Die Klasse *SeaIceDB_HEM_Thickness_Profiles* erzeugt die Metadaten- und Datendateien für das Projekt “HEM_Thickness_Profiles” in dem Projektbereich “SeaIceDB”.

Die Klasse *TextFieldDocument* erzeugt einen Dokument für das Eingabefeld, der die Zeicheneingabe nur bis als Parameter bei der Objekterzeugung übergebene Zeichenanzahl erlaubt. Alle Eingabefelder in dem Programm sind durch die Instanzen der Klasse *JTextField* implementiert, deren nach der Erzeugung ein Dokument in Form des Objektes der Klasse *TextFieldDocument* zugewiesen wird.

4.4.2 Paket Importing

Das Paket Importing beinhaltet die Schnittstelle *ImporterInterface* und folgende 6 Klassen:

- *DBConnector*,
- *Importer*,
- *Ground_Thickness_Importer*,
- *Freeboard_Profiles_Importer*,
- *HEM_Thickness_Importer*,
- *PSMImporter*.

Die Klasse *DBConnector* bildet die Schnittstelle zur Datenbank. Sie stellt die Methoden zur Verfügung, um SQL-Ausdrücke abzarbeiten. Um eine gesicherte Übertragung und Änderung der Daten auf der Datenbank zu garantieren, beinhaltet diese Klasse ein Transaktionsmanagement. In der Schnittstelle *ImporterInterface* werden alle Methoden deklariert, die in der Klasse *Importer* und allen seinen Unterklassen benutzt werden.

Die Klasse *Importer* implementiert die Methoden der Schnittstelle *ImporterInterface* und ist die Superklasse für die Klassen *Ground_Thickness_Importer*, *Freeboard_Profiles_Importer* und *HEM_Thickness_Importer*. In diesen Unterklassen werden folgende Methoden für die Übername der Daten in die Datenbank gemäß der Struktur der entsprechenden Metadaten- und Datentabellen überschrieben:


```

insertMetaData(Vector vWithMetaData, int metaId)
insertData(Vector vWithData, int m_Id, int dataId)
insertDataInAlteredTable(Vector data, int m_Id, int dataId, String[]
allPositions)
getIncrementedDataID().

```

Die Instanzen von diesen Klassen erzeugen keine Benutzungsoberfläche, weil sie keine Benutzerinteraktivität erwarten. Sie fügen stumm die Daten in die Datenbank ein. Der Fortschritt des Verfahrens wird in dem Standardausgabefenster (Konsole) angezeigt.

Die Instanz der Klasse *PSMImporter* erzeugt ein Dialogfenster, um neue Einträge (Wissenschaftler, Parameter und Messmethoden) in der Datenbank zu erfassen bzw. bestehende Einträge anzusehen, zu aktualisieren oder zu löschen.



Abbildung 17: Dialog für das Erfassen des Wissenschaftlers oder der Messmethode in der Datenbank

In der Tabelle rechts (Instanz der Klasse *JTable*) sind alle Einträge aus der entsprechende Datenbanktabelle dargestellt.



Abbildung 18: Dialog für das Erfassen des Parameters in der Datenbank

4.4.3 Paket Presenting

Das Paket Presenting beinhaltet 4 Klassen:

- CampaignViewer,
- MetaDataViewer,
- DataView,
- MyTableModel.
-

Drei ersten Klassen sind aus der Klasse *JDialog* abgeleitet.

Die Instanz der Klasse *CampaignViewer* stellt den modalen Dialog für die Projekt- und Expeditionsauswahl.



Abbildung 19: Dialog für das Projekt- und Expeditionsauswahl

Die entsprechenden Interaktionselemente sind die Auswahlfelder. In der Mitte des Fensters befindet sich die Tabelle mit den Daten von allen Expeditionen, während deren Verlauf die Messungen mit diesem Messverfahren (oder von diesem Projekt) durchgeführt wurden. Die Tabelle ist durch die Instanz der Klasse *JTable* realisiert. Das Modell der Tabelle ist die Instanz der Klasse *MyTableModel*, das einerseits einfache Auswahl der in der Tabelle dargestellten Sätzen (Expeditionsdaten) erlaubt, und andererseits das Schreiben in den Tabellenzellen verhindert. Wählt der Benutzer eine Expedition aus und betätigt die Schaltfläche „Metadaten von ausgewählten Expedition anzeigen“ wird die Klasse *MetaDataViewer* instantiiert.

SealceDB_Ground_Thickness_Profiles

Datenbanktabelle "iceThicknessMetaData" beinhaltet 43 Metadatenansätze, die zur von Ihnen ausgewählte Expedition gehören.
 Wählen Sie die Metadatenansätze aus, in dem Sie in der Tabelle auf entsprechende Zeile ein Mal mit der linken Maustaste klicken. Die Metadatenansätze werden dabei farbig gekennzeichnet.
 Um Daten zur ausgewählten Metadatenansätzen anzusehen, betätigen Sie die Schaltfläche "Daten anzeigen". Um Auswahl aufzuheben, betätigen Sie die Schaltfläche "Auswahl aufheben".

meta_id	event_label	campaign	area	gear	date_time	latitude	longitude	remark
1	Ark12_207p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.07.25	81.46	66.85	
2	Ark12_207p2	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.07.25	81.21	70.12	
3	Ark12_208p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.07.26	81.38	72.92	
4	Ark12_209p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.07.27	81.45	73.78	
5	Ark12_210p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.07.28	81.46	77.47	
6	Ark12_212p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.07.30	81.47	82.16	
7	Ark12_212p2	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.07.30	81.7	82.09	
8	Ark12_213p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.07.31	81.87	83.85	
9	Ark12_214p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.01	81.75	90.14	
10	Ark12_215p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.02	82.19	91.93	Neue Bemerkung
11	Ark12_216p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.03	82.89	92.38	
12	Ark12_218p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.05	84.19	100.55	
13	Ark12_219p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.06	84.55	103.6	
14	Ark12_220p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.07	84.78	105.82	
15	Ark12_221p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.08	85.41	113.04	
16	Ark12_222p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.09	85.86	121.44	
17	Ark12_223p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.10	86.39	134.25	
18	Ark12_226p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.13	86.41	144.56	
19	Ark12_227p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.14	86.17	155.0	
20	Ark12_229p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.16	83.01	150.3	
21	Ark12_230p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.17	82.52	143.56	
22	Ark12_231p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.18	82.5	140.05	
23	Ark12_232p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.19	82.5	136.44	
24	Ark12_232p2	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.19	82.5	134.61	
25	Ark12_232p3	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.19	82.5	134.41	
26	Ark12_233p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.20	82.52	132.94	
27	Ark12_234p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.21	81.98	131.63	
28	Ark12_236p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.23	81.07	138.93	
29	Ark12_236p2	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.23	81.07	138.93	
30	Ark12_238p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.25	80.99	141.78	
31	Ark12_239p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.26	80.92	144.3	
32	Ark12_240p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.27	80.54	148.51	
33	Ark12_240p2	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.27	80.54	148.51	
34	Ark12_240p3	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.27	80.54	148.51	
35	Ark12_240p4	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.27	80.54	148.51	
36	Ark12_240p5	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.27	80.54	148.51	
37	Ark12_240p6	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.27	80.54	148.51	
38	Ark12_240p7	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.27	80.54	148.51	
39	Ark12_242p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.29	80.16	150.08	
40	Ark12_243p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.08.30	79.66	148.66	
41	Ark12_246p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.09.02	78.83	132.8	
42	Ark12_247p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.09.03	78.45	132.6	
43	Ark12_249p1	ARK-XII	Kara/Laptev Sea/Transpolar Drift	ICEM	1996.09.05	77.45	125.19	

Abbildung 20: Dialogfenster für die Auswahl der Metadatenansätze von der ausgewählte Expedition

Die Instanz der Klasse *MetaDataViewer* ermittelt die Metadaten zu der von dem Benutzer in dem Dialogfenster der Klasse *CampaignViewer* ausgewählte Expedition, erzeugt das Fenster, zeigt dort die Tabelle mit diesen Metadaten an, gibt die Möglichkeit, die Metadaten in der Textdatei zu speichern. Betätigt der Benutzer die Schaltfläche „Daten anzeigen“, werden die Daten zu der ausgewählten Metadatenansätzen ermittelt und der Objekt der Klasse *DataViewer* erzeugt.

SealceDB_Ground_Thickness_Profiles						
Parameter	Kuerzel	Messposition		Messmethode		Kuerzel
Measurement point distance	Distance	Distance (RT)		Ruler tape		RT
Ice thickness	Ice thickness	Ice thickness (EM)		Electromagnetic sounding (EM), Geonics EM31 conductivity sensor, Haas et al 1997		EM
Ice thickness	Ice thickness	Ice thickness (DH)		Drilling and ruler tape		DH
Freeboard	Freeboard	Freeboard (DH)		Drilling and ruler tape		DH
Draft	Draft	Draft (EM)		Electromagnetic sounding (EM), Geonics EM31 conductivity sensor, Haas et al 1997		EM
Draft	Draft	Draft (DH)		Drilling and ruler tape		DH
Snow thickness	Snow thickness	Snow thickness (DH)		Drilling and ruler tape		DH
Surface layer thickness	Zsl	Zsl (DH)		Drilling and ruler tape		DH
Surface elevation	Zse	Zse (DH)		Drilling and ruler tape		DH
Melt pond depth	Zmeltpond	Zmeltpond (DH)		Drilling and ruler tape		DH
Melt pond bottom Z coordinate	Zmeltpondbot	Zmeltpondbot (DH)		Drilling and ruler tape		DH

Event label	Distance (RT)	Ice thickness (EM)	Ice thickness (DH)	Freeboard (DH)	Draft (EM)	Draft (DH)	Snow thickness (DH)	Zsl (DH)	Zse (DH)	Zmeltpond (DH)
Ark12_207p1	0.0	0.98	0.9	0.0	-1.01	-0.84	0.03		0.03	
Ark12_207p1	1.0			-0.01						
Ark12_207p1	2.0			0.05						
Ark12_207p1	3.0			0.08						
Ark12_207p1	4.0	1.06		0.07	-1.02					
Ark12_207p1	5.0			0.14						
Ark12_207p1	6.0			0.17						
Ark12_207p1	7.0			0.01						
Ark12_207p1	8.0	1.35		0.15	-1.25					
Ark12_207p1	9.0			0.16						
Ark12_207p1	10.0			0.06						
Ark12_207p1	11.0			0.06						
Ark12_207p1	12.0	1.47		0.05	-1.47					
Ark12_207p1	13.0			0.07						
Ark12_207p1	14.0			0.21						
Ark12_207p1	15.0			0.18						
Ark12_207p1	16.0	1.6		0.2	-1.46					
Ark12_207p1	17.0			0.24						
Ark12_207p1	18.0			0.14						
Ark12_207p1	19.0			0.13						
Ark12_207p1	20.0	1.46	1.59	0.11	-1.41	-1.46	0.0		0.11	
Ark12_207p1	21.0			0.15						
Ark12_207p1	22.0			0.04						
Ark12_207p1	23.0			0.05						
Ark12_207p1	24.0	1.26		0.11	-1.19					
Ark12_207p1	25.0			0.1						
Ark12_207p1	26.0			0.04						
Ark12_207p1	27.0			0.07						
Ark12_207p1	28.0	0.99		0.01	-1.01					
Ark12_207p1	28.0			0.06						
Ark12_207p1	30.0			0.08						
Ark12_207p1	31.0			0.08						
Ark12_207p1	32.0	0.96		0.07	-0.92					
Ark12_207p1	33.0			0.08						
Ark12_207p1	34.0			0.09						
Ark12_207p1	35.0			0.07						
Ark12_207p1	36.0	1.0		0.07	-0.96					
Ark12_207p1	37.0			0.06						
Ark12_207p1	38.0			0.07						
Ark12_207p1	39.0			0.07						
Ark12_207p1	40.0	1.2	1.27	0.08	-1.16	-1.1	0.09		0.17	
Ark12_207p1	41.0			0.11						
Ark12_207p1	42.0			0.08						

Abbildung 21: Fenster für die Darstellung der Daten von den ausgewählten Metadatenätzen

Die Instanz der Klasse *DataViewer* erzeugt das Fenster, stellt die Hilfstabelle zur Messpositionen aus der dem ausgewählten Projekt entsprechende Datenbanktabelle in seinem oberen Bereich dar. Sie präsentiert die aus der Datenbank extrahierte Daten von den Metadatenätzen, die der Benutzer in der Klasse *MetaDataViewer* auswählte und bietet die Möglichkeit, die angezeigte Daten in der Textdateien zu speichern. Nach der Anforderungen des Auftraggebers 1 muss die erste Spalte der Tabelle den Event label darstellen.

5 Entwicklungsumgebung

5.1 Software

Betriebssystem: Solaris 8.0;

J2SE v 1.4.2_06 SDK, Eclipse 2.1 (die Programmentwicklung);

javadoc (Dokumentation);

Eclipse 3.0 mit UML-Plug-In Free Edition Version 2.0.0.beta und Studio Edition Version 2.0.0.beta

Evaluation von der Firma Omondo (UML-Diagramme);

MS Visio 2003 (UML-Diagramme, Zeichnungen).

5.2 Hardware

SUN Ultra5 Workstation.

Während der Entwicklung benutzte ich die folgende Referenzen und Informationsquellen:

- <http://java.sun.com/>

- HTML-Ausgabe von dem Buch "Handbuch der Java-Programmierung",

4. Auflage Addison-Wesley, 2004, ISBN 3-8273-2201-4, steht auf dem <http://www.javabuch.de/> zum freien Download zur Verfügung.

Weil der Quellcode von dem Programm sehr umfangreich ist und nach dem Ausdrucken über 200 Seiten lang wäre, ist er auf der beigelegten Kompaktdisk zusammen mit der mittels des Werkzeugs „javadoc“ erstellten Dokumentation zu finden.

5 Zusammenfassung

Obwohl das Thema für meine Diplomarbeit aus dem Praktikum entstanden ist, war es auf keinen Fall leichte Aufgabe die Entwicklung durchzuführen, alleine aus der Tatsache, dass während der Entwicklungszeit immer neue Anforderungen entstanden. Wie ich vor ein Paar Monaten erfahren habe, haben sich der Datenbestand bei einem Messverfahren und die Anforderungen zu dem Event label geändert, das die Anpassung des von mir entwickelten Programms bedeutet.

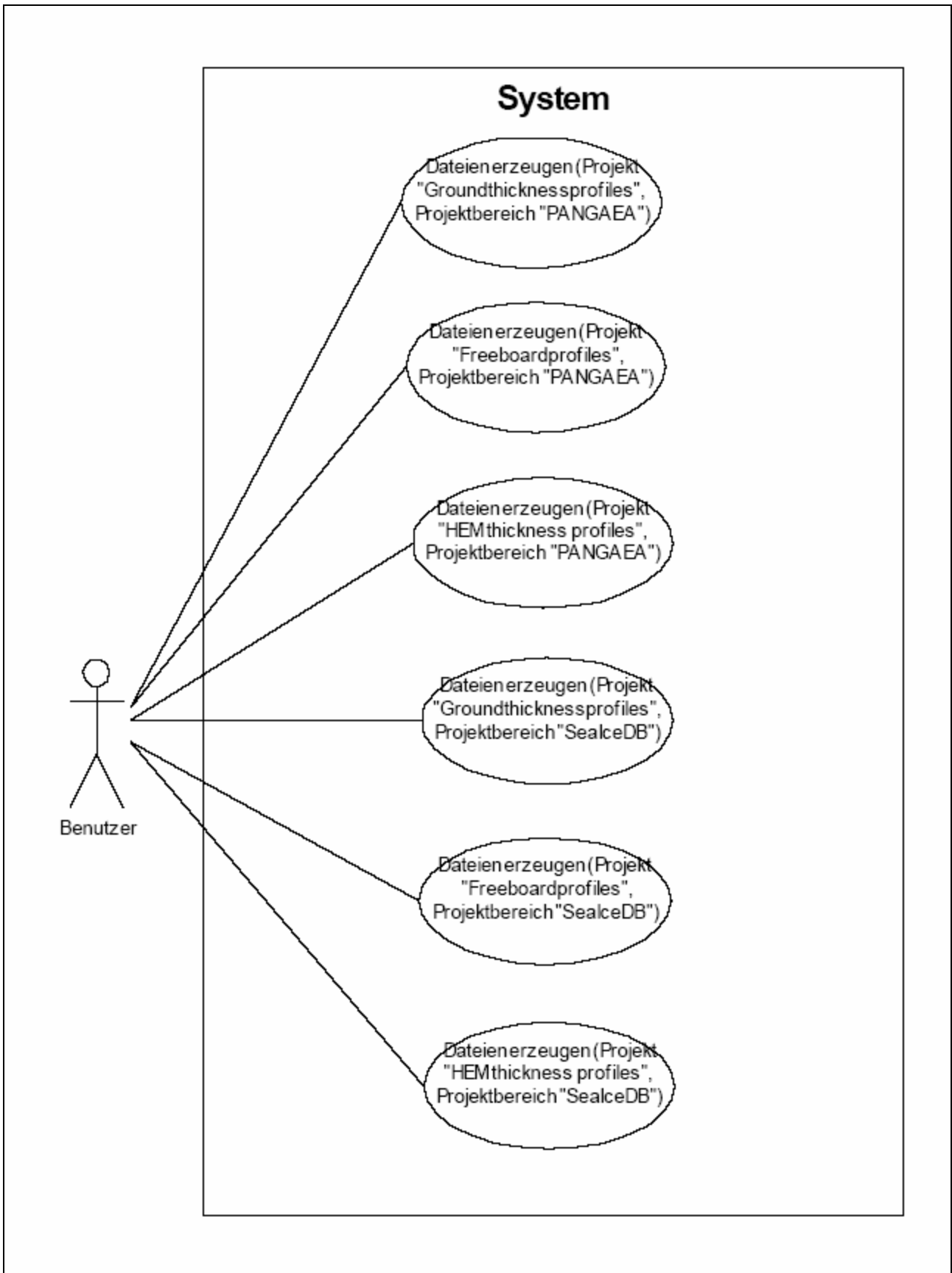
Leider ist die Zeit der Diplomarbeit nicht in einen künftigen Arbeitsverhältnis ans AWI übergegangen.

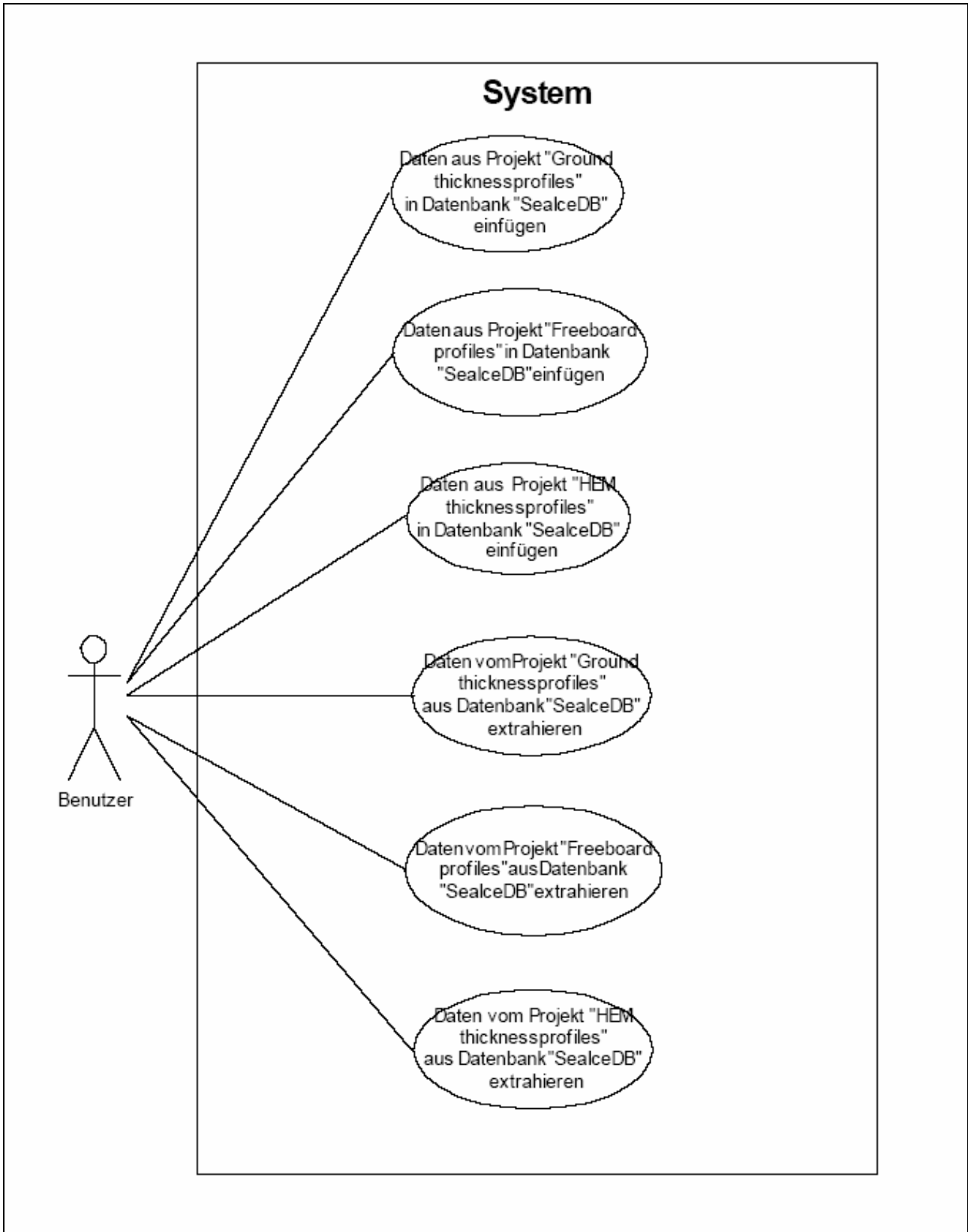
Im Ganzen hat die Entwicklung grossen Spass gemacht und die Menge praktischer Erfahrung gebracht.

Anhang A Metadatendatei „la072508.42“ aus dem Projekt „Freeboard profiles“

1/f : 0.0005
Start: 08:42:47
Ende : 09:02:56
Event 1: 37042 81 21.31 68 19.45
Event 2: 383363 81 21.30 67 59.72
Event 3: 441585 81 21.31 67 56.41
Event 4: 532443
Event 5: 671771 81 21.29 67 43.59
Event 6: 808304
Event 7: 808759
Event 8: 808826
Event 9: 934981 81 21.23 67 28.90
Event 10: 1155815 81 21.28 67 16.78
Event 11: 1318519
Event 12: 1331876
Event 13: 1409068 81 21.28 67 02.96
Event 14: 1430476
Event 15: 1460257
Event 16: 1596814
Event 17: 1643170 81 21.29 66 49.45
Event 18: 1767179
Event 19: 1892524 81 21.27 66 35.48
Event 20: 1925734
Event 21: 1993237
Event 22: 2127219 81 21.14 66 22.79
Event 23: 2405335
Sample: 2412000

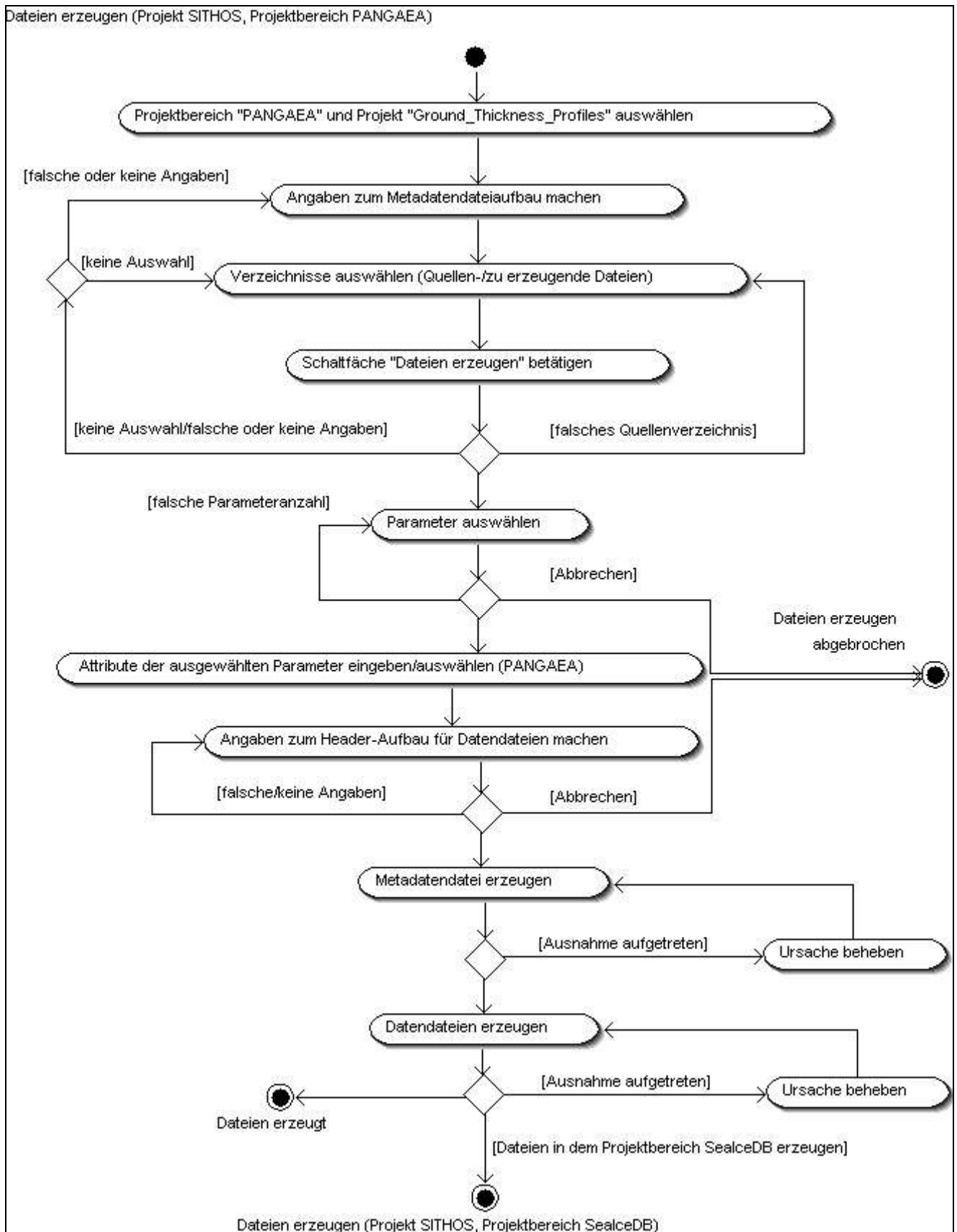
Anhang B Use Case Diagramme

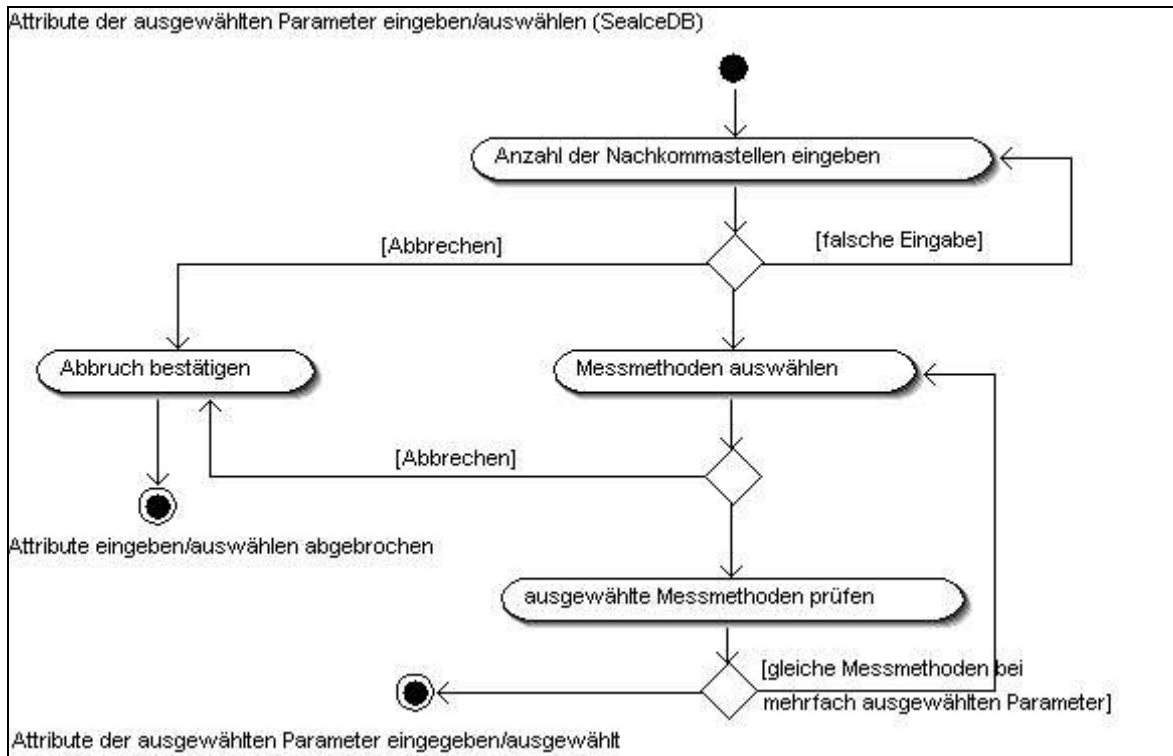


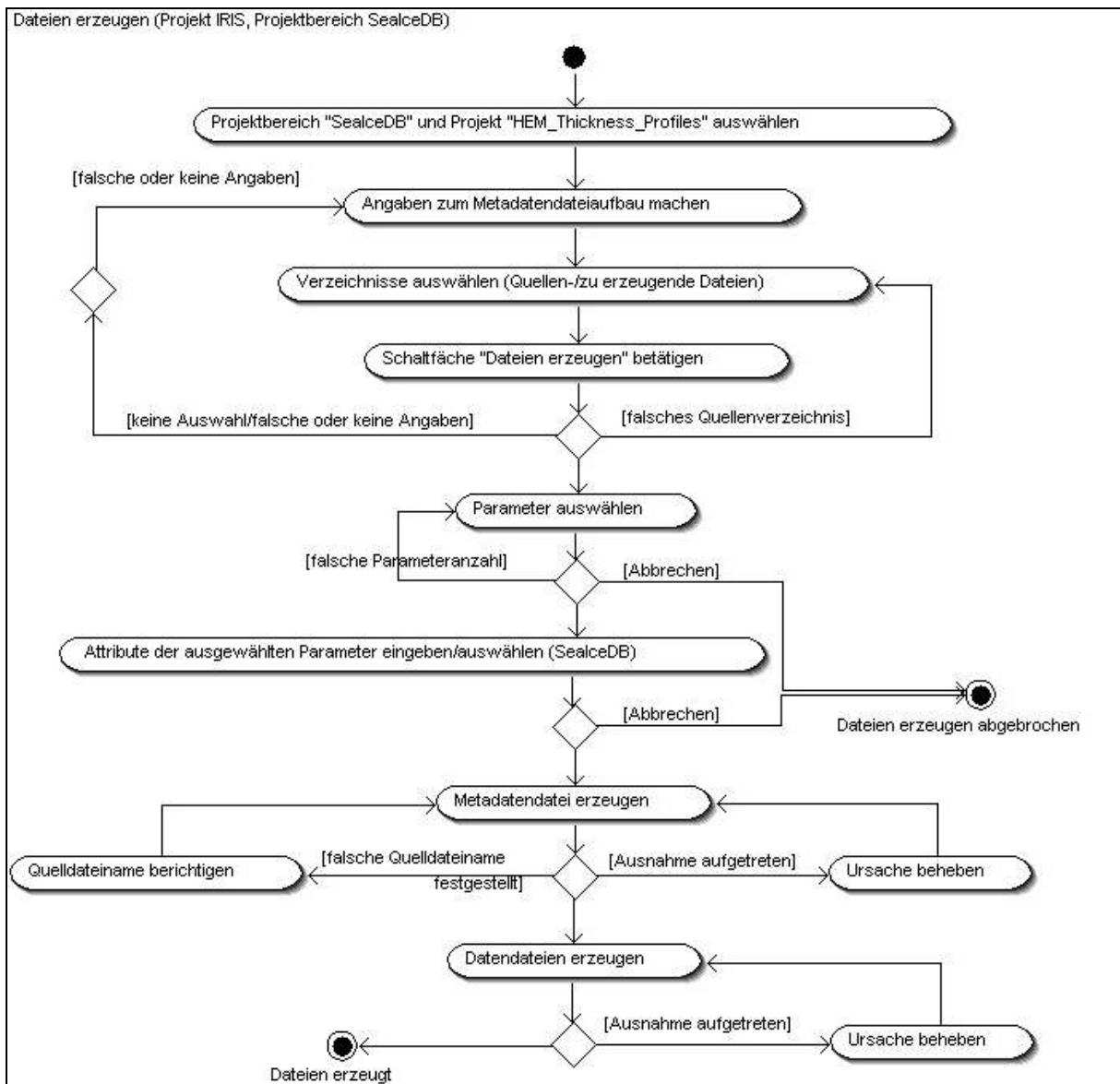


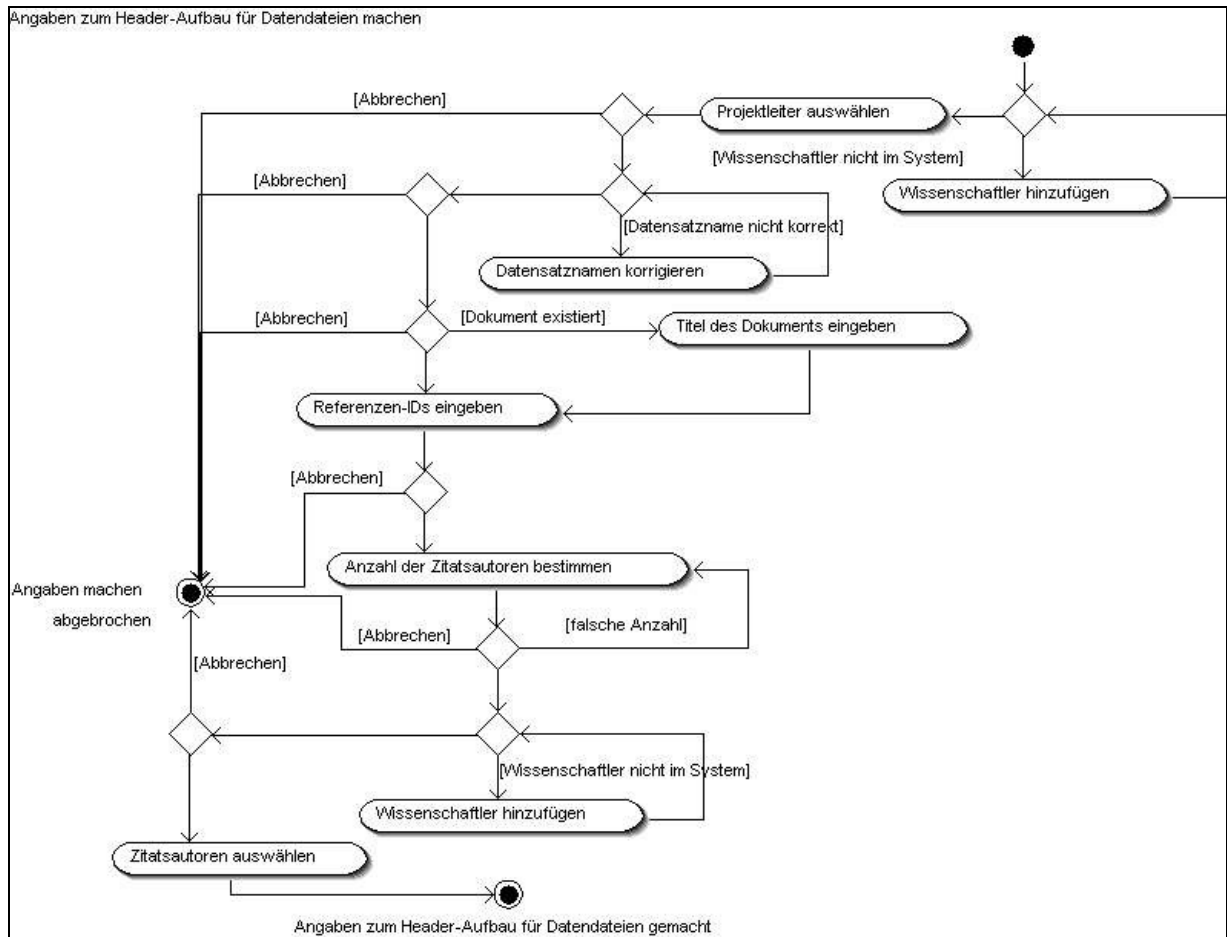
Anhang C Aktivitätsdiagramme

Bei allen diesen Diagrammen steht SITHOS für den Projekt „Ground thickness profiles“,
 GreenICE – für den Projekt „Freeboard profiles“,
 IRIS – für den Projekt „HEM thickness profiles“.

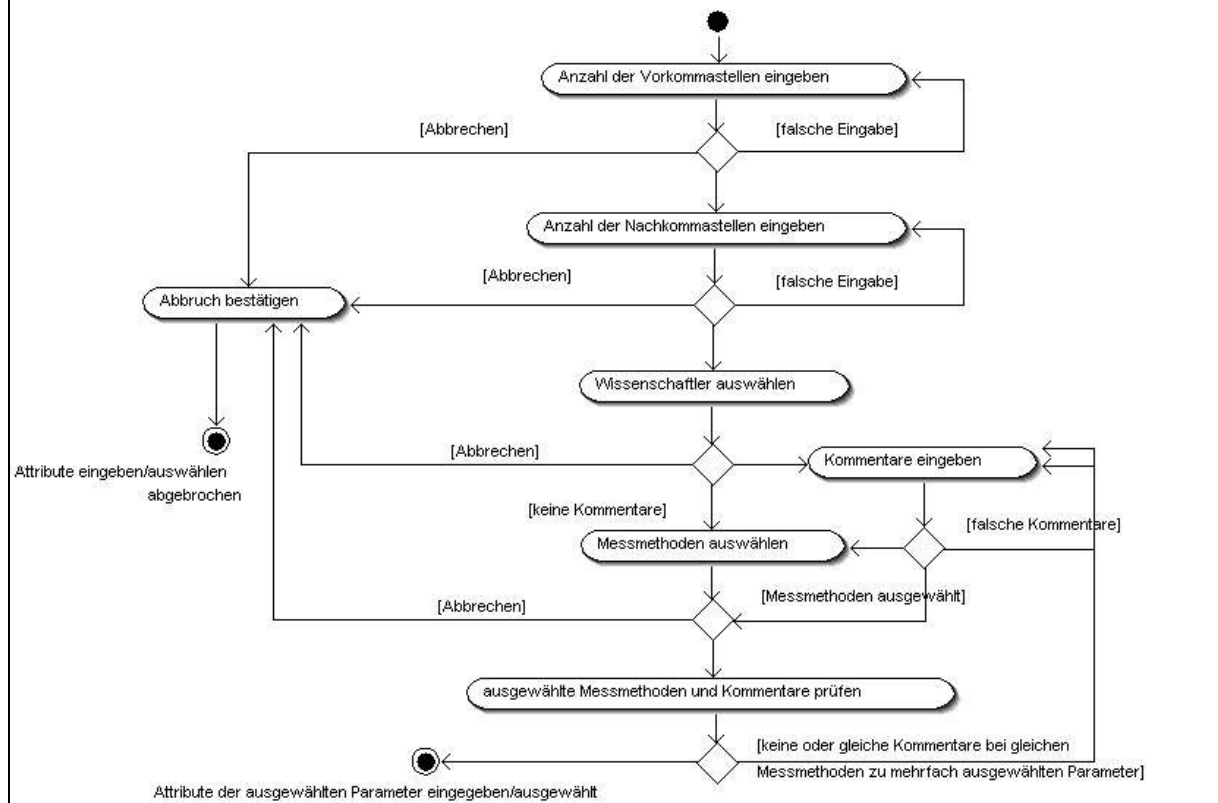






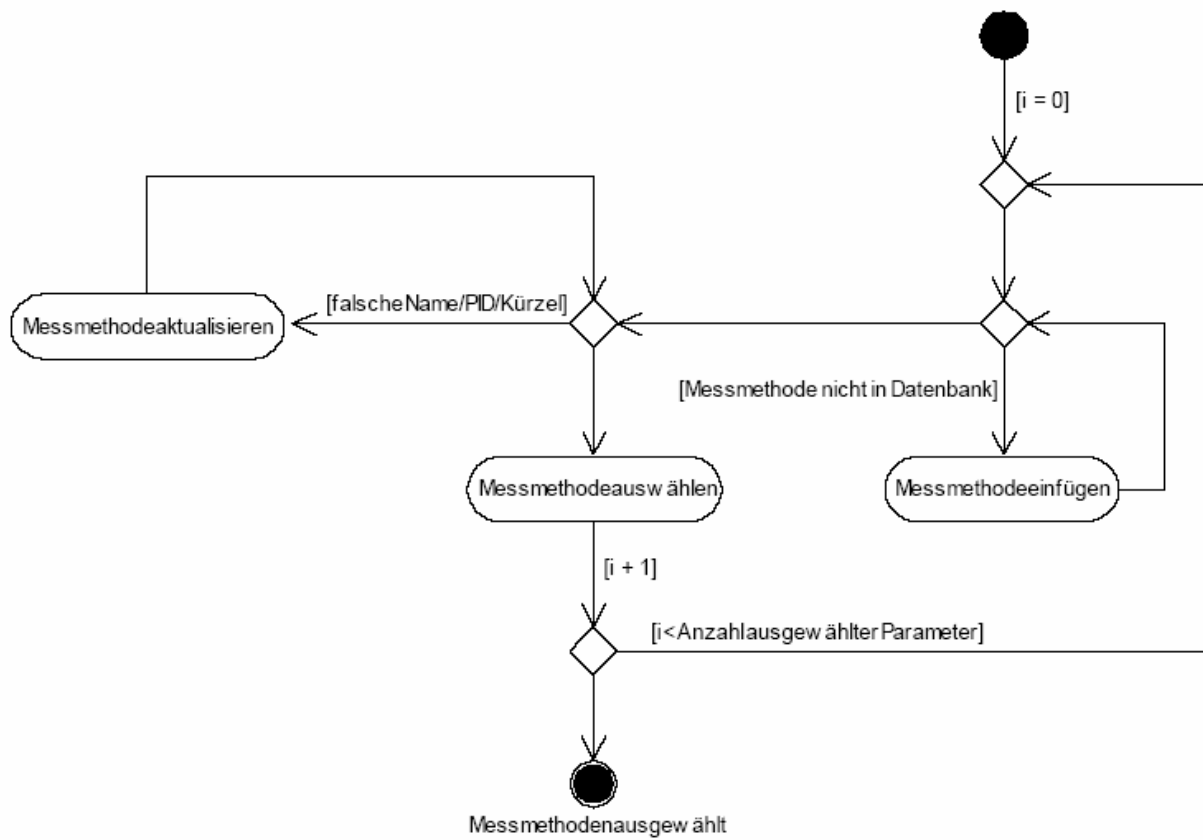


Attribute der ausgewählten Parameter eingeben/auswählen (PANGAEA)



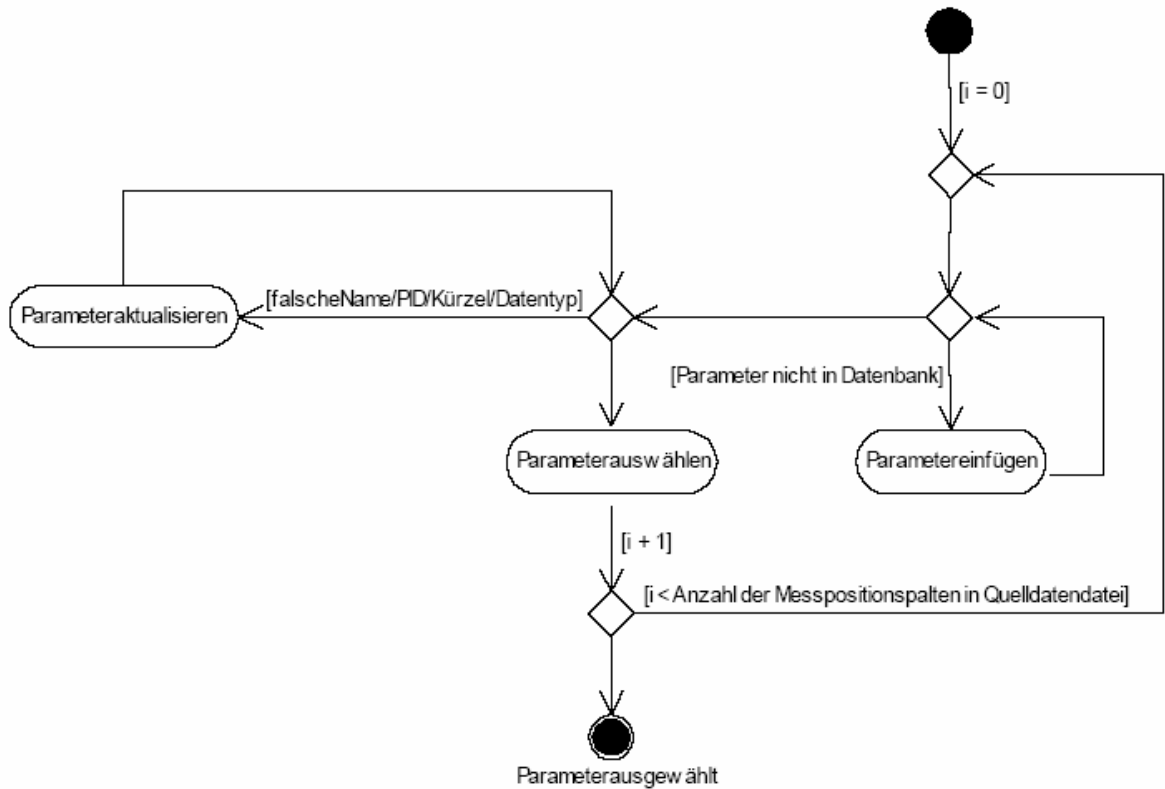
Messmethodenauswählen

i: Integer
Anzahl ausgewählter Parameter: Integer

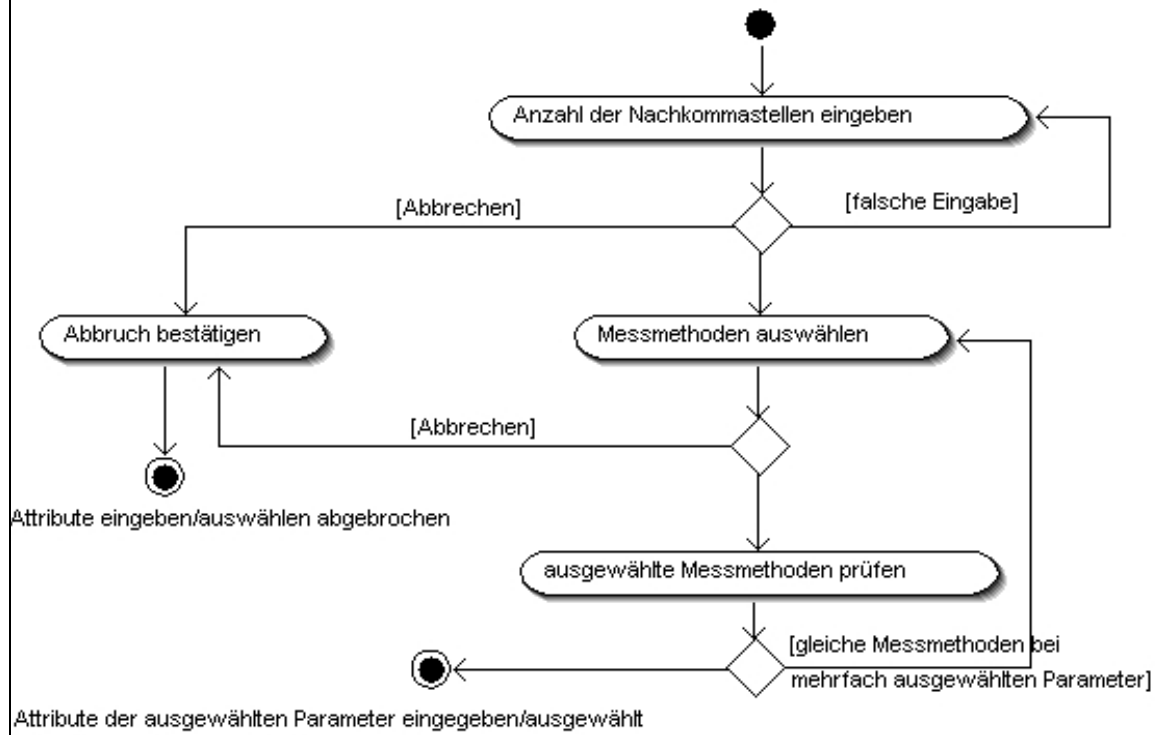


Parameterauswählen

i: Integer
Anzahl der Messpositionspalten in Quelldatendatei: Integer

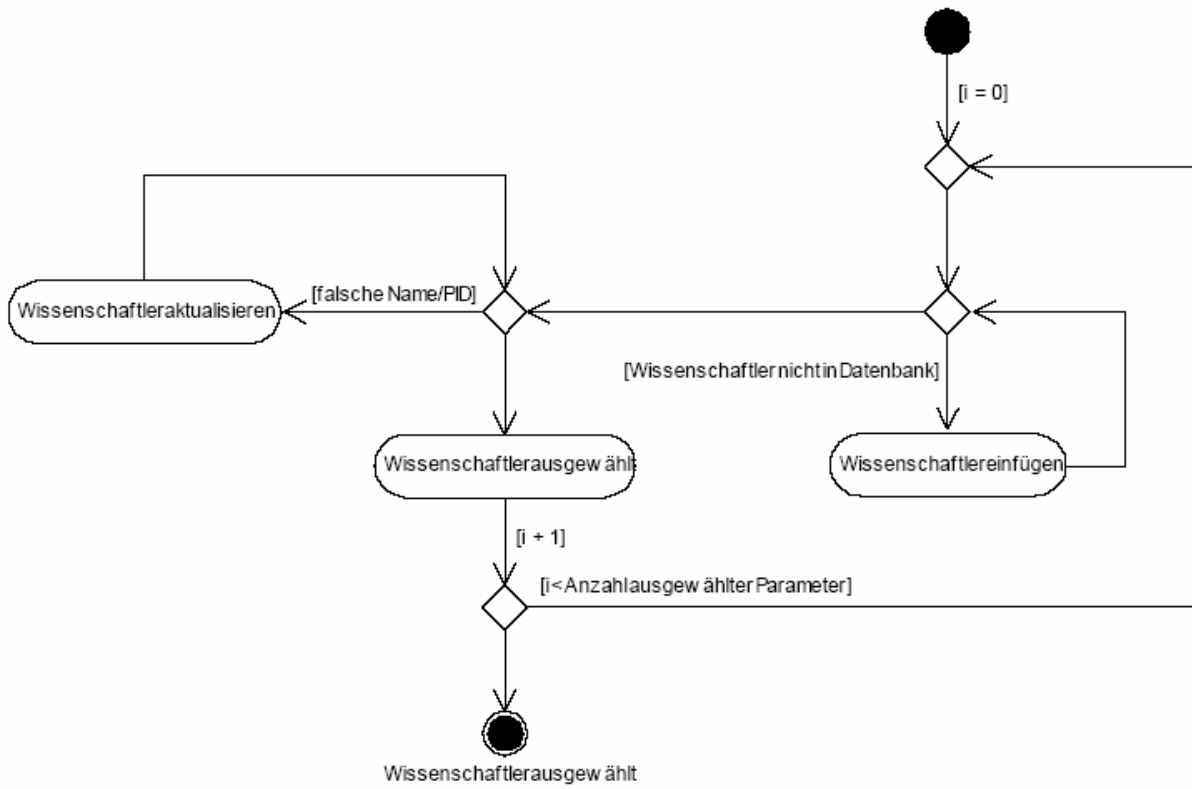


Attribute der ausgewählten Parameter eingeben/auswählen (SealceDB)

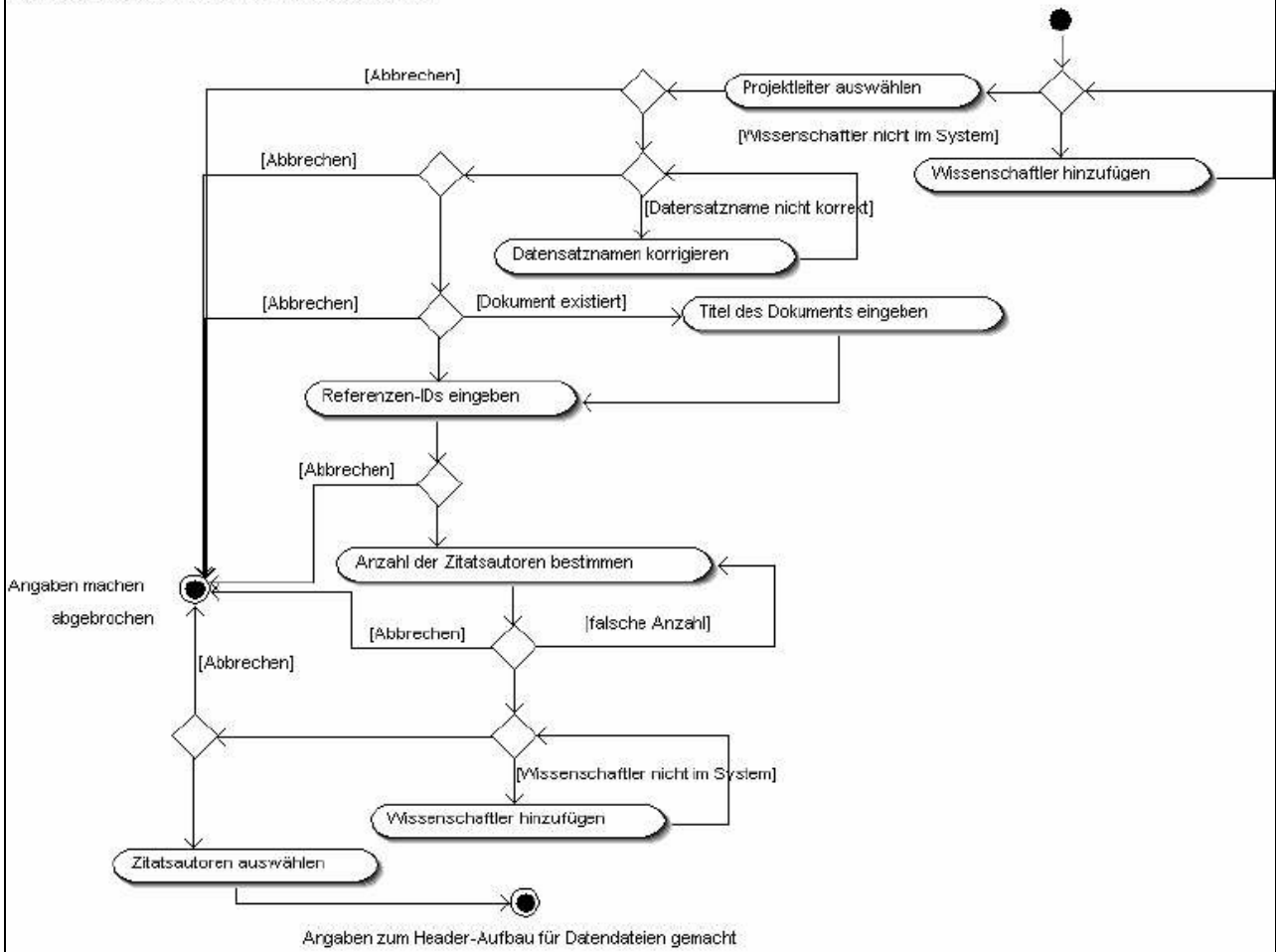


Wissenschaftlerauswählen

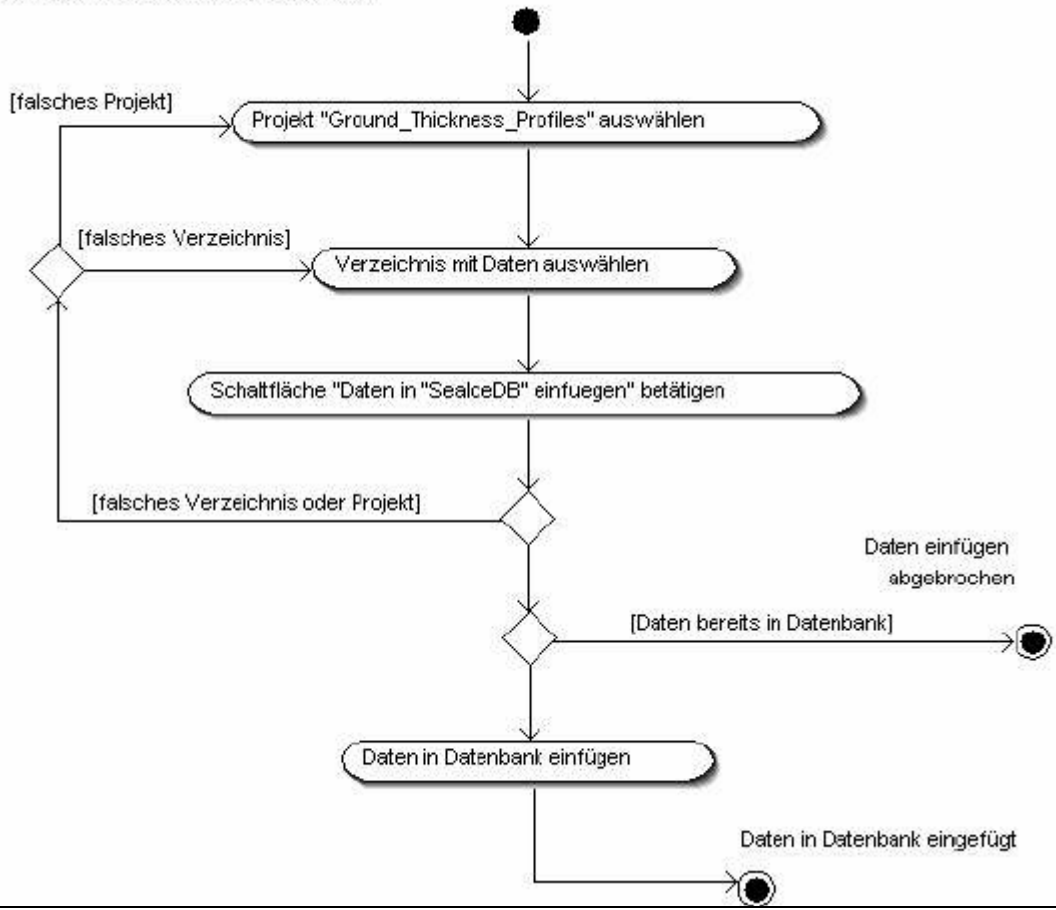
i: Integer
AnzahlAusgewählterParameter: Integer



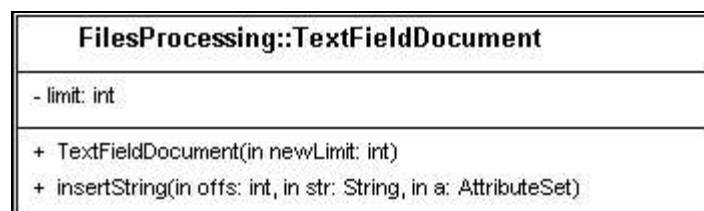
Angaben zum Header-Aufbau für Datendateien machen



Daten aus Projekt SITHOS in Datenbank SealceDB einfügen



Anhang D Klassendiagramme



FilesProcessing::HeaderDialog

```
+ HeaderDialog(owner: JFrame, dialogName: String, modal: boolean)
+ actionPerformed(event: ActionEvent)
+ stateChanged(e: ChangeEvent)
+ setBRefreshEnabled()
+ setMIRefreshEnabled()
+ setDbTableScientistsChanged()
+ setDbTableParametersChanged()
# getHashMapOfMultiChosenParameters(): HashMap
# setHashMapOfMultiChosenParametersToNull()
# getProjectName(): String
# getPIDOfProjectChief(): int
# getPIDsOfCitationAuthors(): String
# getTitleOfDataSet(): String
# getReferenceIDs(): String
# getURLOfDocumentForMethodDescription(): String
# getChosenParamsWithAttribs(): String[][]
# getChosenParamsWithAttribsOrNulls(): String[][]
# getColumnNamesOfSourceMDFile(): String[]
# getParametersPartOfHeader(): String
# getLineOfParameterIDs(lastParameter: int): String
# getCountOfChosenParameters(): int
# getPrecisionOfLatLongInOutputMDFile(): int
# getColNumberofStationInSourceDFFile(): int
# getAllScientists(): String[]
# getProjectPID(): String
# getErrorMessageTitle(): String
# getSelectedParametersWithDataTypes(): String[][]
# getLabelForBEntryToAdd(): String
# getLabelForBRefresh(): String
# getLabelForBCancel(): String
# notValidContentOfTfPrecision(tfContent: String): boolean
# setBSubmitEnabled()
# setMenuSubmitEnabled()
# setBSubmitChoiceOfParamsDisabled()
# setTheSameMethodForMultiChosenParam()
# setProcessFilesForSealceDBAfterPANGAEA()
# setProject(project: String)
# updateFirstPanelInDialog()
```

FilesProcessing::Profiles
<pre># fileSeparator: String # language: String # lineSeparator: String</pre>
<pre>+ Profiles() # getDFilesProcessing(): String # getMDFileProcessing(): String # getMessageEndOfDataProcessingFromFile(fileName: String): String # getMessageEndOfMDFileProcessing(projectSection: String, project: String, mDFileWithPath: String): String # getMessageEndOfProcessData(totalLineCounter: int, fileNumber: int, mDataFileLineCounter: int, className: String): String # getMessageFollowingEventLabel(eventLabel: String): String # getMessageReadyWithEventLabel(eventLabel: String, fileLineCounter: int): String # getMessageReadyWithSourceFile(fileName: String, fileNumber: int, lineCounter: int): String # getMessageStartOfDataProcessing(fileName: String): String # roundUpToFormatPANGAEA(value: String, precision: int): String # roundUpToFormatSealceDB(value: String, precision: int): String # roundUpValueDoubleEmptyNotPossible(value: double, precision: int): double # roundUpValueStringEmptyNotPossible(value: String, precision: int): String</pre>

FilesProcessing::Ground_Thickness_Profiles
<pre># getRestOfEventLabel(s: String): String # getDate(s: String): String # getMeasurementPointDistance(s: String, precision: int): String</pre>

FilesProcessing::HEM_Thickness_Profiles
<pre># roundUpValueOfPDistance(thisPointDistance: double, precision: int): String # calculateDistance(latitudeOfThis: double, longitudeOfThis: double, latitudeOfPreceder: double, longitudeOfPreceder: double): double # falseDayAndMonthInFileName(fileName: String): String # falseHourOrMinutesInFileName(fileName: String): String # getMessageEndOfProcessData(totalLineCounter: int, fileNumber: int, className: String): String</pre>

FilesProcessing::Freeboard_Profiles
<pre># START_TIMEPOINT: int # END_TIMEPOINT: int # LATITUDE_DEGREE: int # LATITUDE_MINUTE: int # LONGITUDE_DEGREE: int # LONGITUDE_MINUTE: int</pre>
<pre># getStartTime(puffer: Vector): String # getEndTime(puffer: Vector): String # getLatitudeStart(puffer: Vector, sizeOfPuffer: int, precision: int): double # getLongitudeStart(puffer: Vector, sizeOfPuffer: int, precision: int): double # getLatitudeEnd(puffer: Vector, lastElementOfVector: int, precision: int): double # getLongitudeEnd(puffer: Vector, lastElementOfVector: int, precision: int): double</pre>

FilesProcessing::PANGAEA_Ground_Thickness_Profiles

- firstPartOfEventLabel: String
- sourceDirectory: String
- outputDirectory: String
- outputMDataFile: String
- ENDINGS_OF_FILE: String
- METADATADATA_VECTOR_SIZE: int
- lineCountOfMDFile: int
- pidOfProjectChief: int
- pIDsOfCitationAuthors: String
- titleOfDataSet: String
- referenceIDs: String
- projectPID: String
- urlOfDocForMethodDescription: String
- chosenParamsWithAttribs: String
- lastChosenParameter: int
- columnNumbersAndPrecisionsOfChosenParams: int
- colNumberofPointDistanceInSourceDFile: int
- colNumberofStationInSourceDFile: int
- bufLengthOfOutput: int
- DATA_VECTOR_SIZE: int
- eventLabel: String
- newParametersPartOfHeader: String
- newLineOfParametersIDs: String
- fileNumber: int

+ PANGAEA_Ground_Thickness_Profiles()

processMetadata()

processData()

getOutputMDataFile(): String

getMessageEndOfDataFilesProcessing(): String

getValuesOfChosenParamWithNoPrecisionPossible(in s: String): String

- julianToGregorianForPANGAEA(in julianDay: String, in year: int, in days: int): String

- noValuesExceptPointDistance(in vColsWithoutValues: Vector): boolean

- createHeader(in sb: StringBuffer)

- roundUpAllParamsWithPrecisions(in inputLine: String): String

- roundUpAllParamsWithPrecisions(in inputLine: String, in columnNumbersAndPrecisions: int, in newLastChosenParam: int): String

- roundUpNotAllParamsWithPrecisions(in inputLine: String): String

- roundUpNotAllParamsWithPrecisions(in inputLine: String, in columnNumbersAndPrecisions: int, in newLastChosenParam: int): String

- getMessageUnvalidDataSet(in lineOfFile: int, in elementsOfLine: String): String

- getMessageUnvalidDataSet(in lineOfFile: int, in elementsOfLine: String, in z: int): String

- getMessageUnvalidDataSetJustPointDistance(in lineOfFile: int, in elementsOfLine: String): String

- getMessageNoValuesExceptPointDistance(in eventLabel: String): String

- getMessageNoValuesOfParametersAtAll(in eventLabel: String): String

FilesProcessing::SealceDB_Ground_Thickness_Profiles

```
- firstPartOfEventLabel: String
- sourceDirectory: String
- outputDirectory: String
- ENDINGS_OF_FILE: String
- lineCountOfMDFile: int
- outputMDataFile: String
- METADATA_VECTOR_SIZE: int
- countOfChosenParameters: int
- chosenParamsWithAttribsOrNulls: String
- colNumbOfLastPosInDataSet: int
- columnNumbersAndPrecisionsOfChosenParams: int
- colNumberOfStationInSourceDFile: int
- bufLengthOfOutput: int
- fileNumber: int
- DATA_VECTOR_SIZE: int

+ SealceDB_Ground_Thickness_Profiles()
# processMetadata()
# processData()
# getOutputMDataFile(): String
# getMessageEndOfDataFilesProcessing(): String
# getValuesOfChosenParamWithNoPrecisionPossible(in s: String): String
- julianToGregorianForSealceDB(in julianDay: String, in year: int, in days: int): String
- roundUpAllParamWithPrecisions(in inputLine: String): String
- roundUpNotAllParamWithPrecisions(in inputLine: String): String
```

FilesProcessing::PANGAEA_Freeboard_Profiles

```
- firstPartOfEventLabel: String
- sourceDirectory: String
- outputDirectory: String
- outputMDataFile: String
- precisionOfLatLongInOutputMDataFile: int
- lineCountOfMDFFile: int
- PART_OF_FILE_NAME: String
- METADATA_VECTOR_SIZE: int
- fileNumber: int
- lastChosenParameter: int
- chosenParamsWithAttribs: String
- columnNumbersAndPrecisionsOfChosenParams: int
- bufLengthForOutputLineWithoutEventLabel: int
- colNumbOfPDistanceInSourceDataFile: int
- DATA_VECTOR_SIZE: int

+ PANGAEA_Freeboard_Profiles()
# processMetadata()
# processData()
# getOutputMDataFile(): String
# getMessageEndOfDataFilesProcessing(): String
- roundUpNotAllParamWithPrecisions(in inputLine: String): String
```

FilesProcessing::SealceDB_Freeboard_Profiles

```
- outputDirectory: String
- outputMDataFile: String
- firstPartOfEventLabel: String
- sourceDirectory: String
- lineCountOfMDFFile: int
- fileNumber: int
- PART_OF_FILE_NAME: String
- METADATA_VECTOR_SIZE: int
- countOfChosenParameters: int
- chosenParamsWithAttribsOrNulls: String
- colNumbOfLastParInDataSet: int
- colNumbersAndPrecisionsOfChosenParams: int
- bufLengthForOutputLineWithoutEventLabel: int
- colNumbOfPDistanceInSourceDataFile: int
- DATA_VECTOR_SIZE: int

+ SealceDB_Freeboard_Profiles()
# processMetadata()
# processData()
# getOutputMDataFile(): String
# getMessageEndOfDataFilesProcessing(): String
- roundUpNotAllParamWithPrecisions(in inputLine: String): String
```

FilesProcessing::PANGAEA_HEM_Thickness_Profiles

- campaign: String
- sourceDirectory: String
- outputDirectory: String
- outputMDDataFile: String
- chosenParamsWithAttribs: String
- columnNumbersAndPrecisionsOfChosenParams: int
- colNumberofLatitudeInSourceFile: int
- colNumberofLongitudeInSourceFile: int
- colNumberofPDistanceInSourceFile: int
- precisionOfPointDistance: int
- colNumberofIceThicknessInSourceFile: int
- falseNameOfSourceFileFound: boolean
- ENDINGS_OF_FILE: String
- VECTOR_SIZE: int
- lastChosenParameter: int
- bufLengthForOutputLineWithoutEventLabel: int
- fileCounter: int

- + PANGAEA_HEM_Thickness_Profiles()
- # processMetadata()
- # processData()
- # getMessageEndOfMDFileProcessing(): String
- # getMessageEndOfDataFilesProcessing(): String
- getStartLatitude(in puffer: Vector, in sizeOfPuffer: int, in precision: int): String
- getStartLongitude(in puffer: Vector, in sizeOfPuffer: int, in precision: int): String
- getEndLatitude(in puffer: Vector, in lastElementOfPuffer: int, in precision: int): String
- getEndLongitude(in puffer: Vector, in lastElementOfPuffer: int, in precision: int): String
- getTimeEvent2(in timeEvent1: String): String
- getFileNumber(in fileNumber: int): String
- roundUpFirstLine(in inputLine: String, in thisLatitude: double, in thisLongitude: double): String
- roundUpOtherButFirstLines(in inputLine: String, in thisLatitude: double, in thisLongitude: double, in thisPointDistance: String): String

FilesProcessing::SealceDB_HEM_Thickness_Profiles

- sourceDirectory: String
- outputDirectory: String
- chosenParamsWithAttribsOrNulls: String
- colNumbOfLastParInDataSet: int
- countOfChosenParameters: int
- columnNumbersAndPrecisionsOfChosenParams: int
- colNumberofLatitudeInSourceFile: int
- colNumberofLongitudeInSourceFile: int
- colNumberofPDistanceInSourceFile: int
- precisionOfPointDistance: int
- headerLine: String
- falseNameOfSourceFileFound: boolean
- outputMDataFile: String
- ENDINGS_OF_FILE: String
- VECTOR_SIZE: int
- bufLengthForOutputLineWithoutEventLabel: int
- fileCounter: int

- + SealceDB_HEM_Thickness_Profiles()
- # processMetadata()
- # processData()
- # getMessageEndOfMDFileProcessing(): String
- # getMessageEndOfDataFilesProcessing(): String
- getStartLatitude(in puffer: Vector, in sizeOfPuffer: int, in precision: int): String
- getStartLongitude(in puffer: Vector, in sizeOfPuffer: int, in precision: int): String
- getEndLatitude(in puffer: Vector, in lastElementOfPuffer: int, in precision: int): String
- getEndLongitude(in puffer: Vector, in lastElementOfPuffer: int, in precision: int): String
- roundUpFirstLine(in inputLine: String, in thisLatitude: double, in thisLongitude: double): String
- roundUpOtherButFirstLines(in inputLine: String, in thisLatitude: double, in thisLongitude: double, in thisPointDistance: String): String

FilesProcessing::AttributsOfChosenParameters

```
- title: String
- language: String
- lineSeparator: String
- parent: HeaderDialog
- project: String
- countOfChosenParams: int
- selectParamWithDataType: String
- vOfPanelsForChosenParameters: Vector
- vOfPanelsChosenParamWithPrecAndMethod: Vector
- vOfTFsDigitsBeforeDot: Vector
- vOfTFsPrecision: Vector
- vOfCBsScientists: Vector
- vOfTFsCommentForChosenParameter: Vector
- vOfCBsMethods: Vector
- hMapOfMultiChosenParameters: HashMap
- bRefresh: JButton
- bCancel: JButton
- bSubmit: JButton
- psmlImporter: PSMLImporter
- dbTableScientistsChanged: boolean
- dbTableMethodsChanged: boolean
- dbConnector: DBConnector
- contentOfTFsForDigitsBeforeDots: String
- contentOfTFsForPrecision: String
- contentOfTFsForComment: String
- methodsOfChosenParams: String

+ AttributsOfChosenParameters(in owner: JDialog, in dialogName: String, in modal: boolean)
+ actionPerformed(in event: ActionEvent)
+ setDbTableScientistsChanged()
+ setDbTableMethodsChanged()
+ setBRefreshEnabled()
# getContentOfTFsDigitsBeforeDots(): String[]
# getContentOfTFsPrecision(): String[]
# getVectorOfCBsScientists(): Vector
# getContentOfTFsComments(): String[]
# getMethodsOfChosenParameters(): String[]
- createTaHelp(in cp: Container)
- createPAllChosenParametersWithAttributes(): JPanel
- setDefaultMeasurementMethod(in chosenParameter: String, in cBAllMethods: JComboBox)
- getMessageAttributesOfChosenParameters(): String
- getToolTipTextForBEntryToAdd(): String
- getToolTipTextForBRefresh(): String
- getToolTipTextForBSubmit(): String
- getAllScientists(): String[]
- getAllMethodsFromDB(): String[]
- getMessageEmptyMethodsTable(): String
- getTitleEmptyMethodsTable(): String
- getErrorMessageTitle(): String
- getMessageEmptyTfForDigitsBeforeDot(): String
- getMessageEmptyTfForDigitsBeforeDot(in numberOfChosenParam: int): String
- notValidContentOfTfForDigitsBeforeDot(in digitsBeforeDot: String): boolean
- getMessageNotValidContentOfTfForDigitsBeforeDot(): String
- getMessageNotValidContentOfTfForDigitsBeforeDot(in numberOfChosenParam: int): String
- getMessageEmptyTfPrecision(): String
- getMessageEmptyTfPrecision(in numberOfChosenParam: int): String
- getMessageNotValidContentOfTfPrecision(): String
- getMessageNotValidContentOfTfPrecision(in numberOfChosenParam: int): String
- getTitleForToLongComment(): String
- getMessageToLongCommentForParameter(): String
- getMessageToLongCommentForParameter(in numberOfChosenParam: int): String
- getMessageNoCommentForMultiChosenParam(in multiChosenParameter: String): String
- getMessageEqualCommentsForMultiChosenParam(in multiChosenParameter: String): String
- getMessageTheSameMethodForMultiChosenParam(in parameter: String): String
- getMessageCancelDialog(): String
- getTitleCancelDialog(): String
```

Importing::DBConnector
- dbConnection: Connection - dbStatement: Statement
+ DBConnector() + open(in host: String, in port: String, in user: String, in pwd: String, in db: String) + executeQuery(in query: String): ResultSet + executeUpdate(in query: String) + close()

«interface» ImporterInterface
+ <i>fetchMetaId(in mDTable: String): int</i> + <i>getMetaData(in file: File): Vector</i> + <i>insertMetaData(in mData: Vector, in mid: int): int</i> + <i>getIncrementedMetaID(): int</i> + <i>fetchDataId(in dTable: String): int</i> + <i>getData(in file: File, in size: int): Vector</i> + <i>getMetaId(in data: Vector, in mDTable: String): int</i> + <i>getMessageDataSetsWithMetaIDAlreadyInDB(in dTable: String, in count: int, in metaId: int): String</i> + <i>insertData(in data: Vector, in m_id: int, in dId: int): int</i> + <i>insertDataInAlteredTable(in data: Vector, in m_id: int, in dId: int, in positions: String): int</i> + <i>getIncrementedDataID(): int</i>

Importing::Importer
<pre># dbConnector: DBConnector # language: String</pre>
<pre>+ Importer() + fetchMetald(in mDTable: String): int + getMetaData(in file: File): Vector + getMessageCountOfMetaDataSetsInFile(in fileName: String, in lineCount: int): String + fetchDataId(in dTable: String): int + getData(in file: File, in size: int): Vector + getMessageCountOfDataSetsInFile(in fileName: String, in lineCount: int): String + getMetald(in data: Vector, in mDTable: String): int + getMessageMDSetAlreadyInDB(in eventLabel: String, in mDTable: String, in metald: int): String + getMessageStartInsertionOfMDSet(in eventLabel: String, in mDTable: String): String + getMessageMDDataSetInserted(in eventLabel: String, in metald: int): String + getMessageDataSetsWithMetalDAAlreadyInDB(in dTable: String, in count: int, in metald: int): String + insertMetaData(in mData: Vector, in mId: int): int + getIncrementedMetalD(): int + insertData(in data: Vector, in m_Id: int, in dId: int): int + insertDataInAlteredTable(in data: Vector, in m_Id: int, in dId: int, in positions: String): int + getIncrementedDataID(): int</pre>

Importing::Ground_Thickness_Importer
<pre>- mID: int - EVENT_LABEL: int - CAMPAIGN_ID: int - GEAR: int - DATE_TIME: int - LATITUDE: int - LONGITUDE: int - REMARK: int - dID: int - DISTANCE_RT: int - ICE_THICKNESS_EM: int - ICE_THICKNESS_DH: int - FREEBOARD_DH: int - DRAFT_EM: int - DRAFT_DH: int - SNOW_THICKNESS_DH: int - ZSL_DH: int - ZSE_DH: int - ZMELTPOND_DH: int - ZMELTPONDBOT_DH: int</pre>
<pre>+ insertMetaData(vWithMetaData: Vector, metald: int): int + getIncrementedMetalD(): int + insertData(vWithData: Vector, m_Id: int, dataId: int): int + insertDataInAlteredTable(data: Vector, m_Id: int, dataId: int, allPositions: String): int + getIncrementedDataID(): int</pre>

Importing::Freeboard_Profiles_Importer

- mID: int
- EVENT_LABEL: int
- CAMPAIGN_ID: int
- GEAR: int
- START_DATE_TIME: int
- START_LATITUDE: int
- START_LONGITUDE: int
- END_DATE_TIME: int
- END_LATITUDE: int
- END_LONGITUDE: int
- REMARK: int
- DISTANCE_CALC_FROM_GPS: int
- FREEBOARD_HELI_LASER: int
- ICE_COVERAGE_LASER_ECHO: int
- dID: int

- + insertMetaData(vvWithMetaData: Vector, metaId: int): int
- + getIncrementedMetalD(): int
- + insertData(vvWithData: Vector, m_Id: int, dataId: int): int
- + insertDataInAlteredTable(data: Vector, m_Id: int, dataId: int, allPositions: String): int
- + getIncrementedDataID(): int

Importing::HEM_Thickness_Importer
<ul style="list-style-type: none"> - mID: int - EVENT_LABEL: int - CAMPAIGN_ID: int - GEAR: int - DATE_TIME: int - START_LATITUDE: int - START_LONGITUDE: int - END_LATITUDE: int - END_LONGITUDE: int - REMARK: int - dID: int - LATITUDE_GPS: int - LONGITUDE_GPS: int - DISTANCE_CALC_FROM_GPS: int - FID_NG: int - ICE_THICKNESS_HEM: int - HEIGHT_LASER: int
<ul style="list-style-type: none"> + insertMetaData(vWithMetaData: Vector, metalId: int): int + getIncrementedMetalID(): int + insertData(vWithData: Vector, m_id: int, dataId: int): int + insertDataInAlteredTable(data: Vector, m_id: int, dataId: int, allPositions: String): int + getIncrementedDataID(): int

Importing::PSMImporter
<ul style="list-style-type: none"> + PSMImporter(owner: JDialog, project: String, modal: boolean) + actionPerformed(event: ActionEvent) + itemStateChanged(e: ItemEvent)

Presenting::MyTableModel
<ul style="list-style-type: none"> - columnNames: String - data: Object
<ul style="list-style-type: none"> + getColumnCount(): int + getData(): Object[][] + getRowCount(): int + getValueAt(row: int, column: int): Object + setColumnNames(columnNames: String) + setData(data: Object)

Presenting::CampaignViewer

```
- language: String
- lineSeparator: String
- PROJECTS: String
- cbProject: JComboBox
- dbConnector: DBConnector
- columnNames: String
- campaignID: int
- campaignNameArea: String
- tableModel: MyTableModel
- table: JTable
- bShowMetadata: JButton
- mDataViewer: MetadataViewer

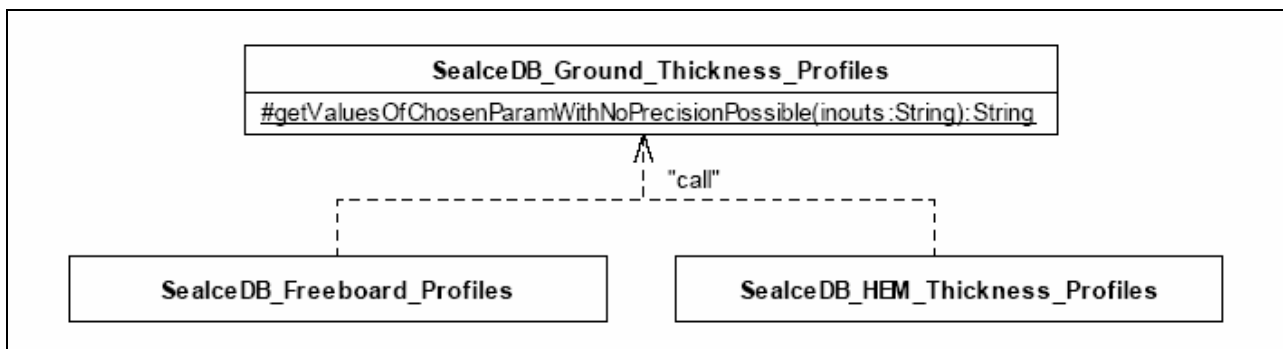
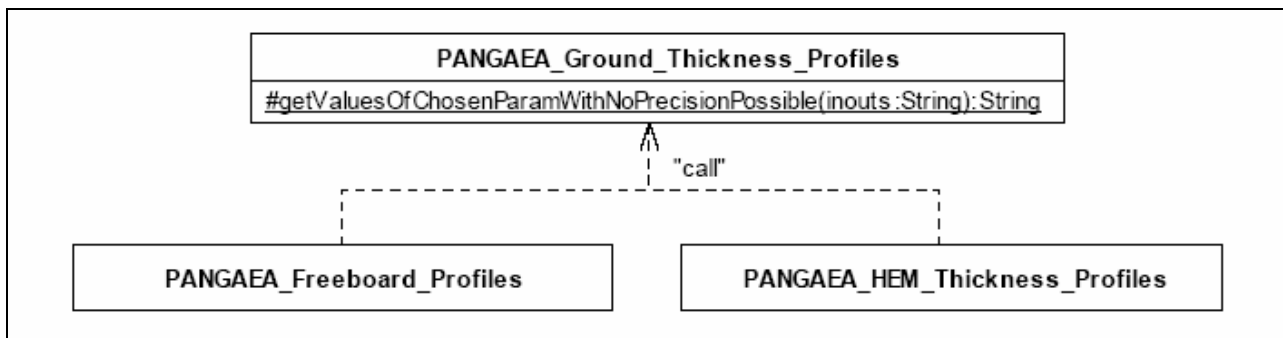
+ CampaignViewer(sitDB: SI_Thickness_DB, title: String, modal: boolean)
+ actionPerformed(event: ActionEvent)
+ itemStateChanged(e: ItemEvent)
# getCampaignID(): int
# getCampaignNameArea(): String[]
# getMessageExceptionOccured(): String
- getDataForTable(): Object[][]
- getLabelProjectSelection(): String
- getToolTipTextForCBProjects(): String
- getColumnNames(): String[]
- getMessageEmptyMetadataTable(metadataTable: String): String
- getTitleEmptyMetadataTable(): String
- getLabelChoseCampaign(): String
- getToolTipTextForTable(): String
- getLabelForBShowMetadata(): String
- getToolTipTextForBShowMetadata(): String
- getLabelForBBack(): String
- getLabelForBExit(): String
- getToolTipTextForBExit(): String
```

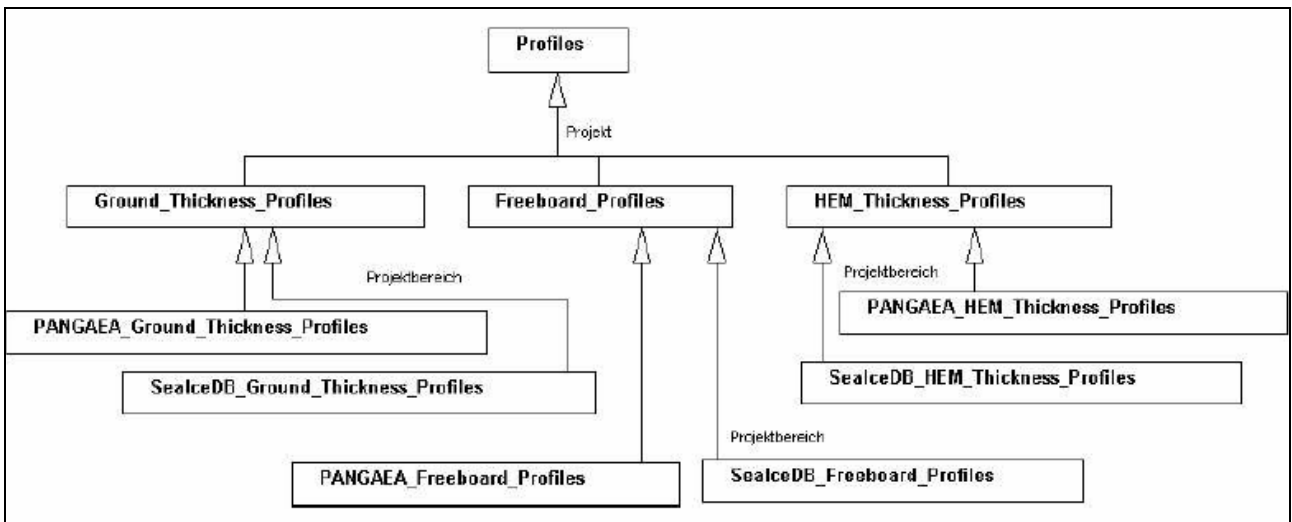
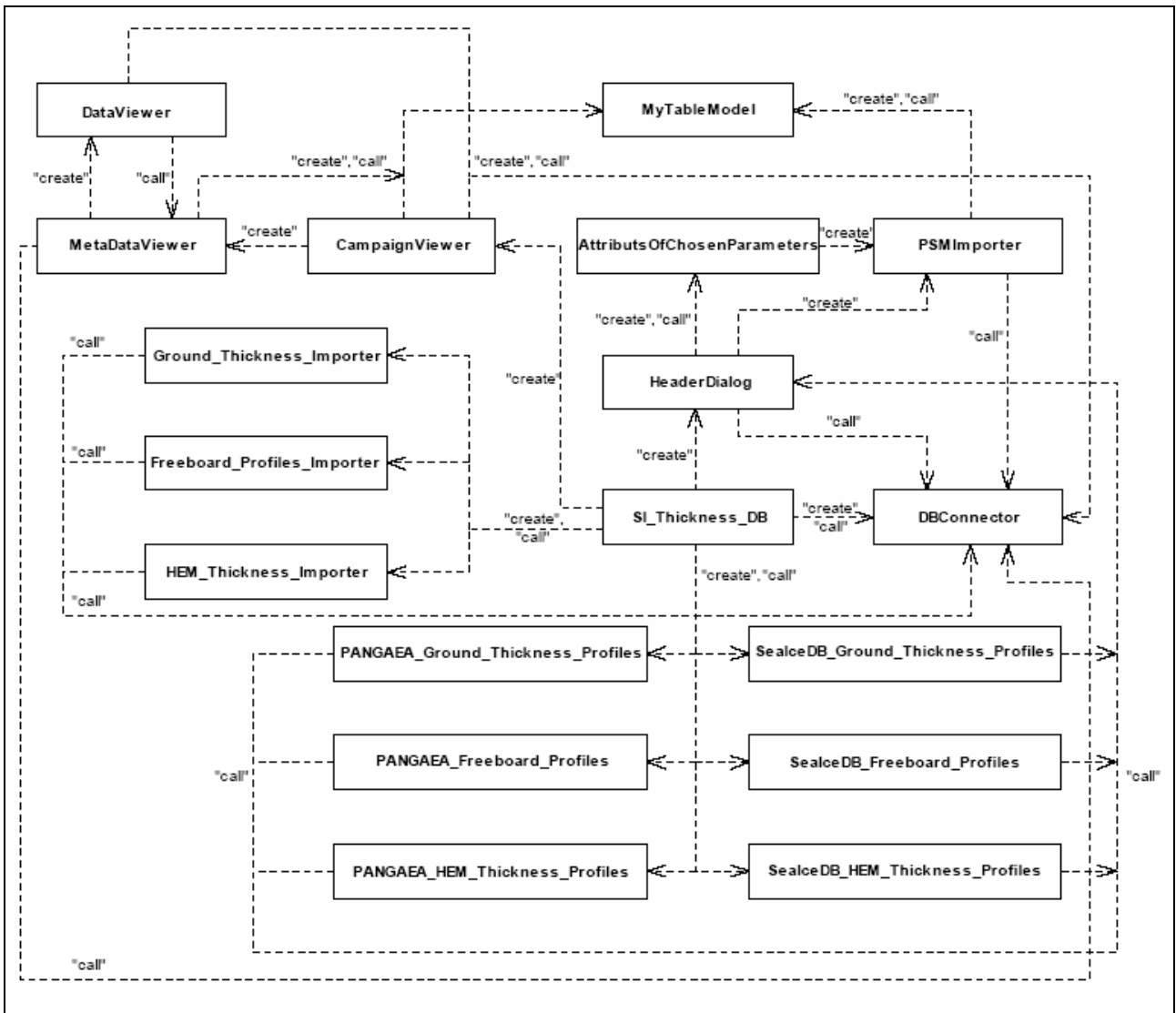
Presenting::MetaDataViewer

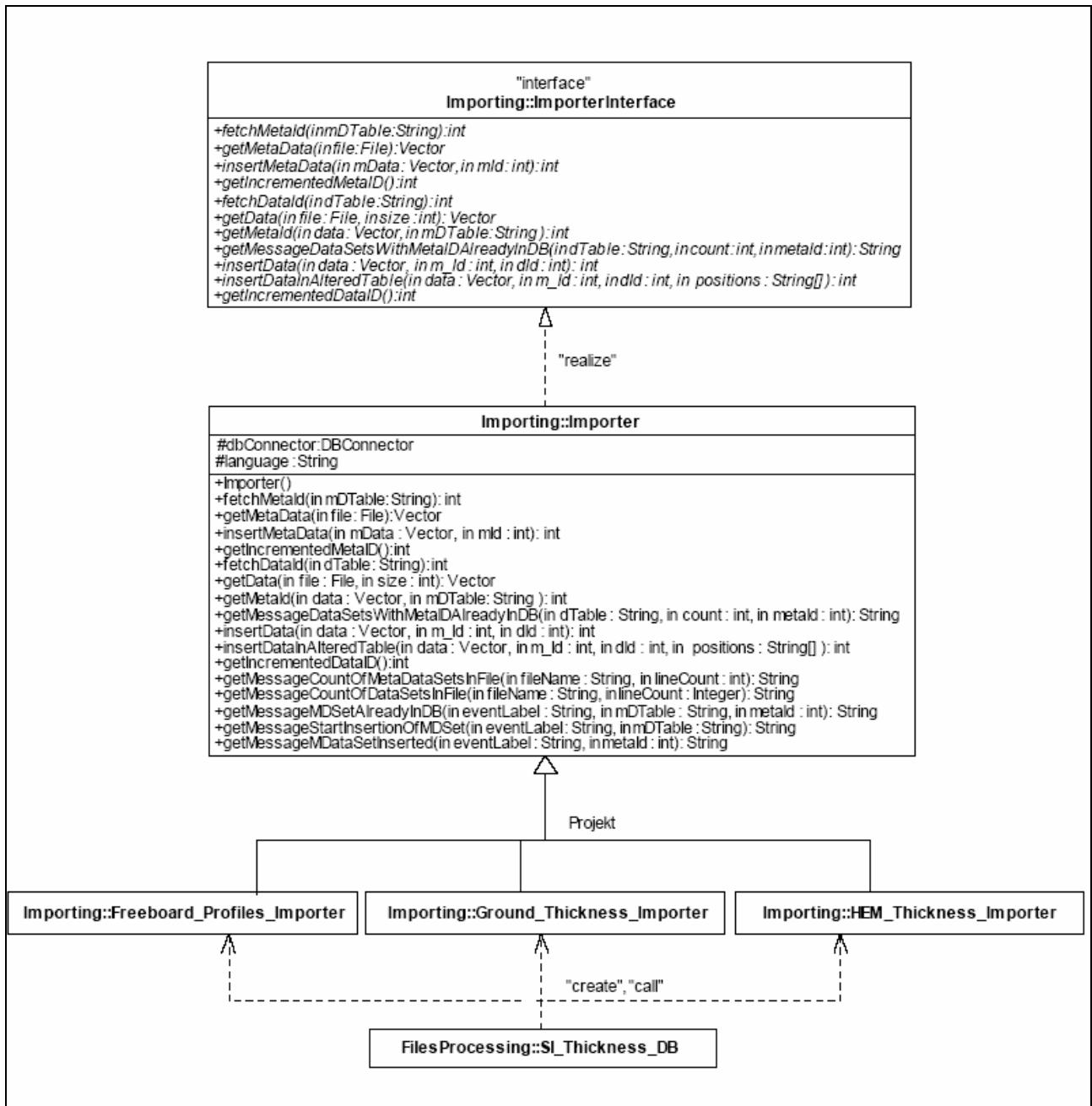
```
- project: String
- language: String
- mDataTableInDB: String
- tableModel: MyTableModel
- dbConnector: DBConnector
- table: JTable
- mDataForTable: Object
- exceptionOccured: boolean
- countOfMetaDataRows: int
- bSave: JButton
- bSubmit: JButton
- bInsertRemarks: JButton
- selectedRows: int
- selectedMetalDs: int
- selectedEventLabels: String
- dataTableInDB: String
- namesOfColumnsInThisTable: String
- lineSeparator: String
- vDataForTable: Vector
- positionsColumnsInDTable: String
- lengthOfVRow: int
- vColumnNamesForTableInDViewer: Vector
- dataViewer: DataViewer

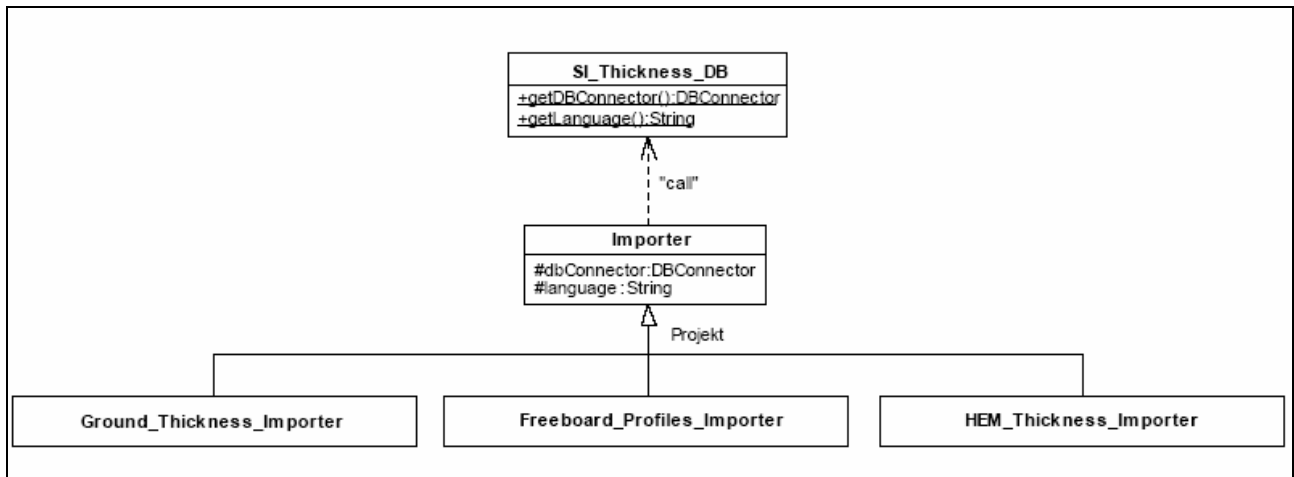
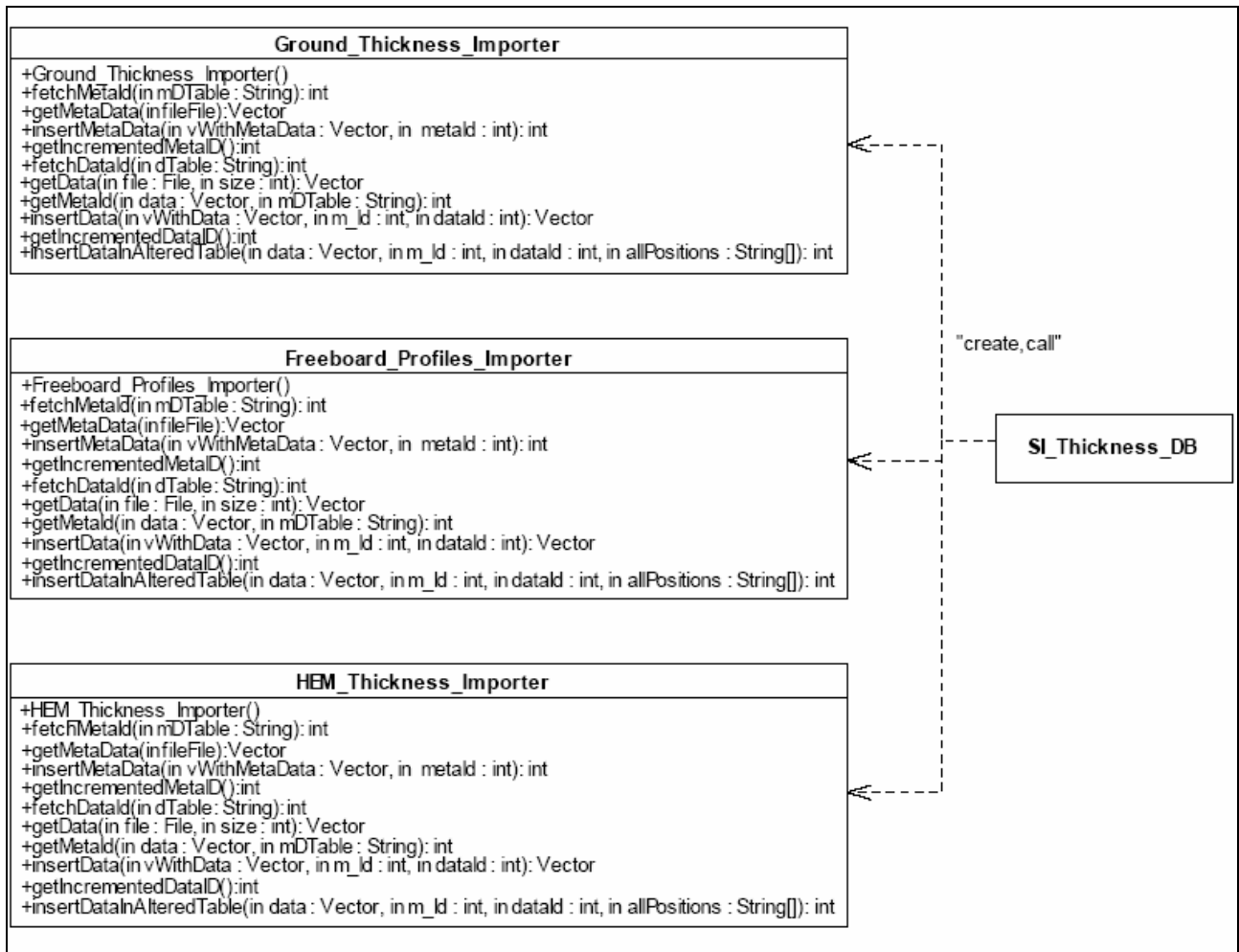
+ MetaDataViewer(in cv: CampaignViewer, in title: String, in modal: boolean)
+ actionPerformed(in event: ActionEvent)
+ getNameOfDTableInDB(in project: String): String
+ getNamesOfPositionsColumnsFromDataTableInDB(in project: String): String[]
# getDataForTable(): Vector
# getColumnNamesForTableInDViewer(): Vector
# getMessageSubmitDirectoryName(in path: String): String
- getNameOfMDTableInDB(): String
- getNamesOfColumnsInThisMDTable(): String[]
- getMDataForTable(): Object[][]
- addLabelsToPanel(in panel: JPanel)
- getWidthOfWindow(): int
- insertRemark()
- getMessageCountOfCampaignsMDataSets(in rowCount: int): String
- getMessageChooseMDataSets(): String
- getMessageShowData(): String
- getLabelForBSave(): String
- getLabelForBSubmit(): String
- getLabelForBInsertRemarks(): String
- getToolTipTextForBInsertRemarks(): String
- getLabelForBClear(): String
- getToolTipTextForBClear(): String
- getLabelForBBack(): String
- getTitleOfSavingDialog(): String
- getMessageMDataSaved(in count: int, in fileName: String, in path: String): String
- getTitleMDataSaved(): String
- getMessageNoDataForChosenMetadataset(in eventLabel: String, in metalid: int): String
- getMessageNoDataForChosenMetadatasets(): String
- getTitleNoDataForChosenMetadatasets(): String
- getMessagePutRemarkIn(): String
- getTitleInsertRemark(): String
- getErrorMessageTitle(): String
- getMessageNotValidRemark(): String
- getMessageSubmitInsertionOfRemark(): String
- getMessageCountOfDataSetsWithChosenMetalD(in metalid: int, in eventLabel: String, in countOfDataRows: int): String
```

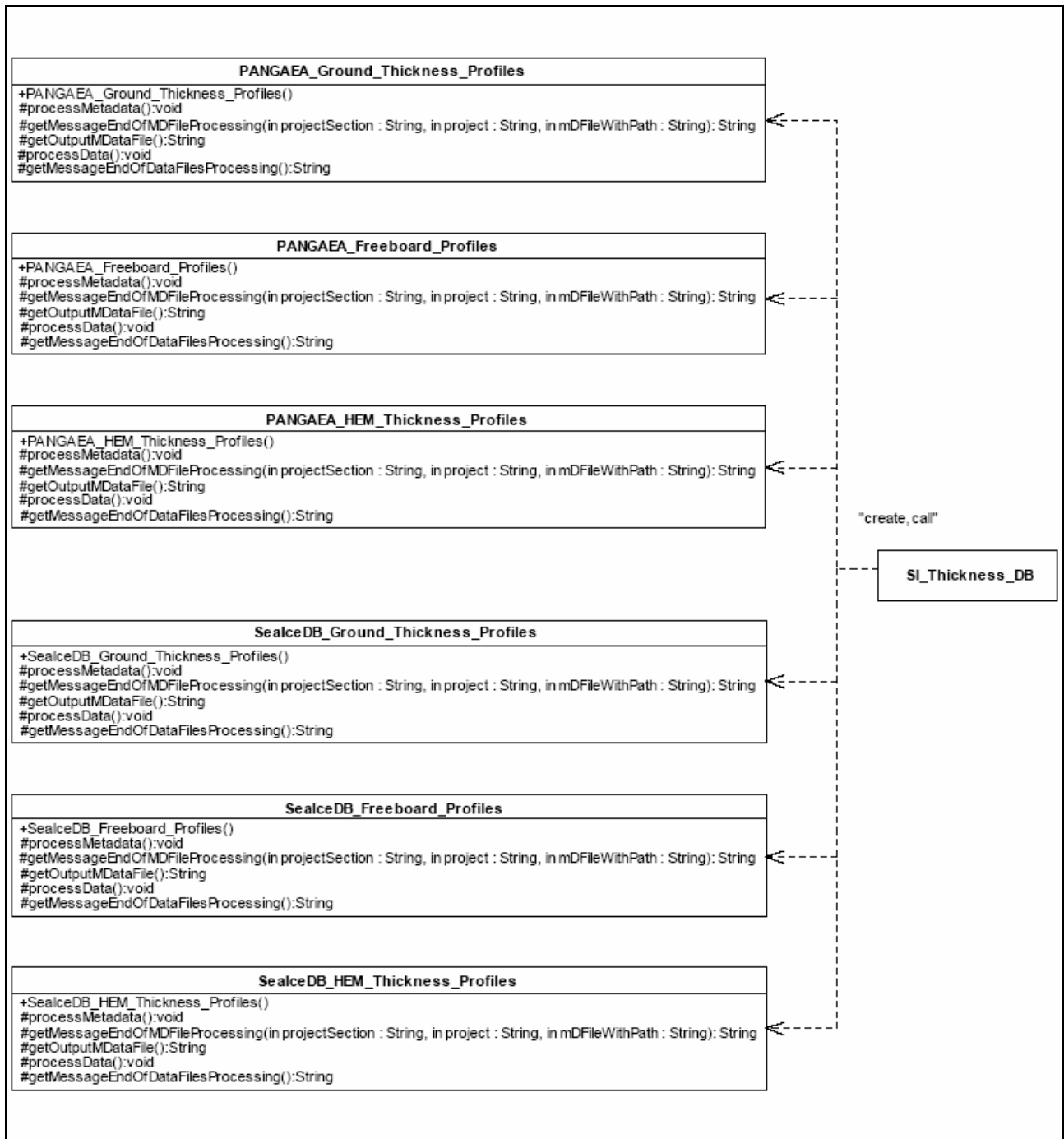
Presenting::DataViewer
<ul style="list-style-type: none"> - pForLabels: JPanel - vDataForTable: Vector - countOfDataRows: int - columnNames: String - language: String - lineSeparator: String
<ul style="list-style-type: none"> + DataViewer(in mdViewer: MetaDataViewer, in title: String, in modal: boolean) + actionPerformed(in event: ActionEvent) - initColumnNames(in vColNames: Vector): String[] - getColumnNamesOfHeaderTable(): String[] - getDataForHeaderTable(in positionsColumnNames: String, in vColNames: Vector): String[][] - getLabelForBsave(): String - getToolTipTextForBsave(): String - getLabelForBBack(): String - getToolTipTextForBBack(): String - getTitleOfSavingDialog(): String - getLabelForSavingDialog(): String - getMessageReadyWithEventLabel(in eventLabel: String, in fileLineCounter: int): String - getMessageDataSaved(in count: int, in fileCounter: int, in path: String): String - getTitleDataSaved(): String

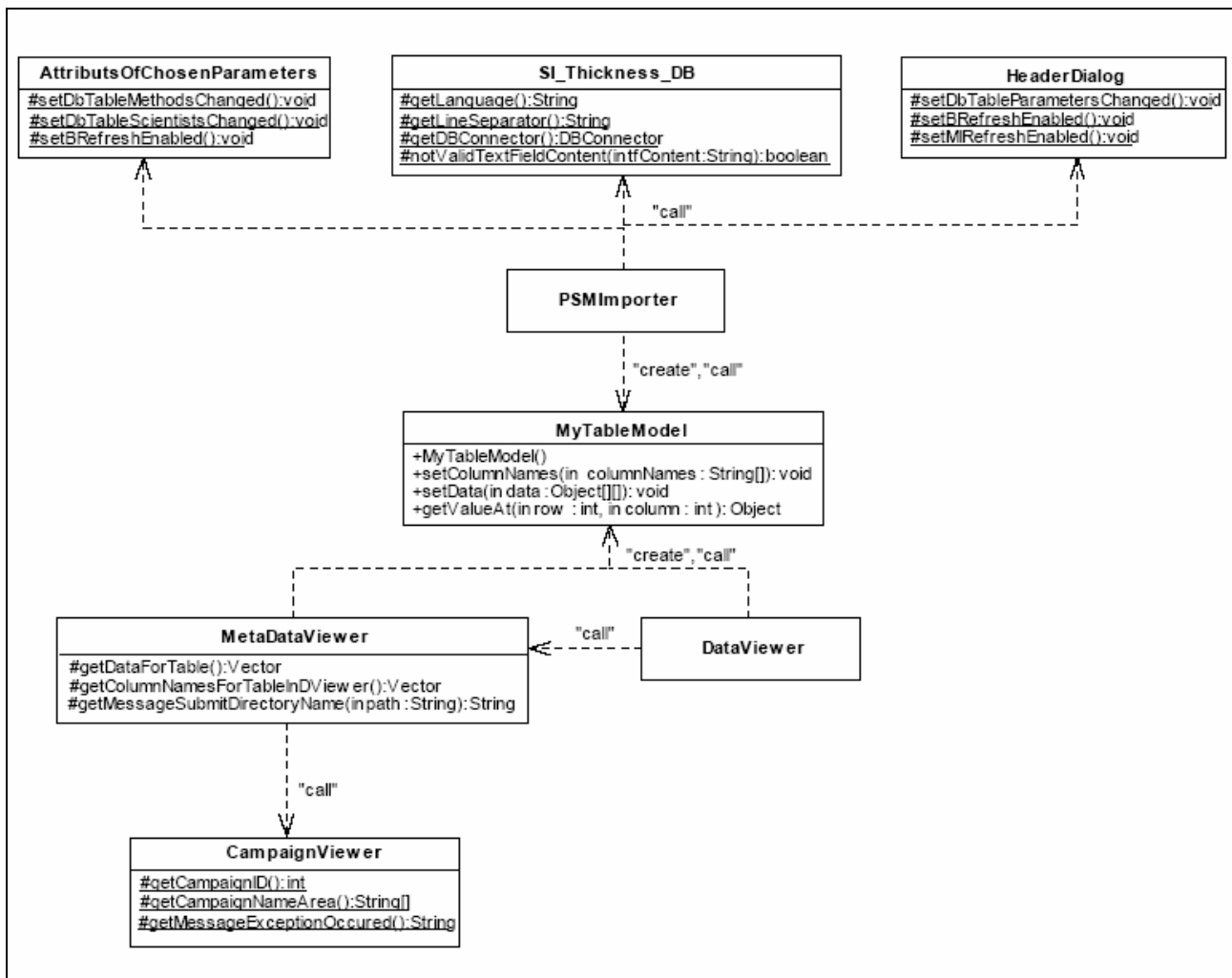


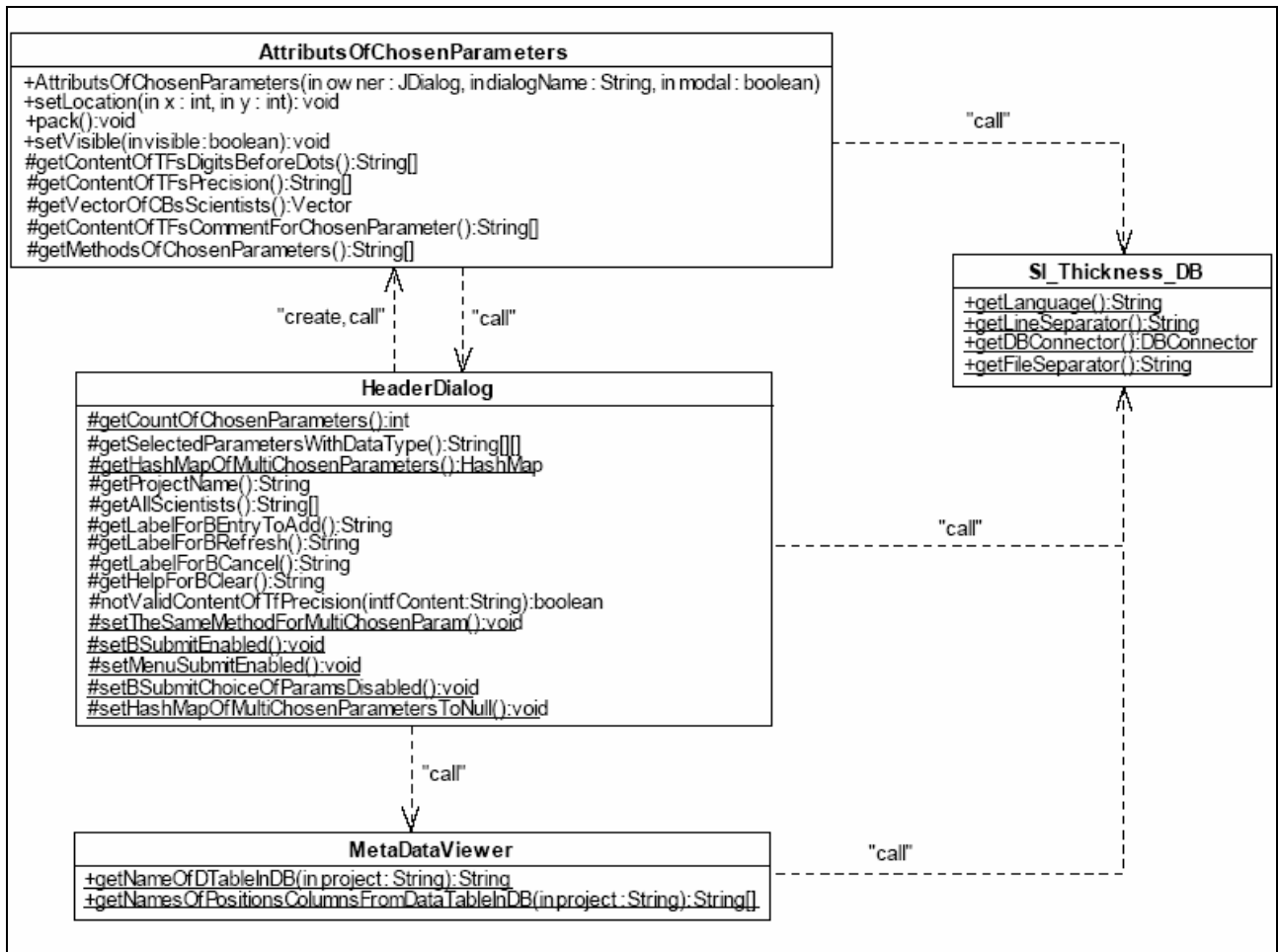


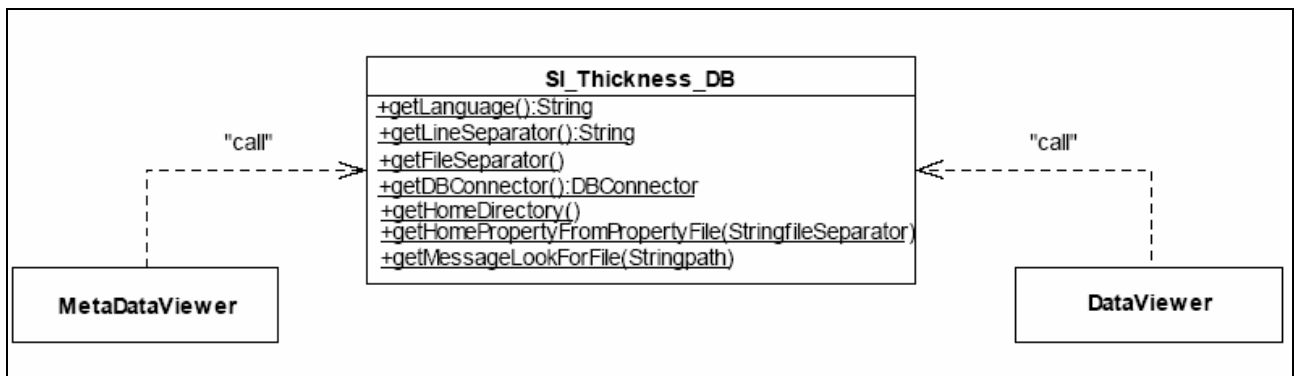
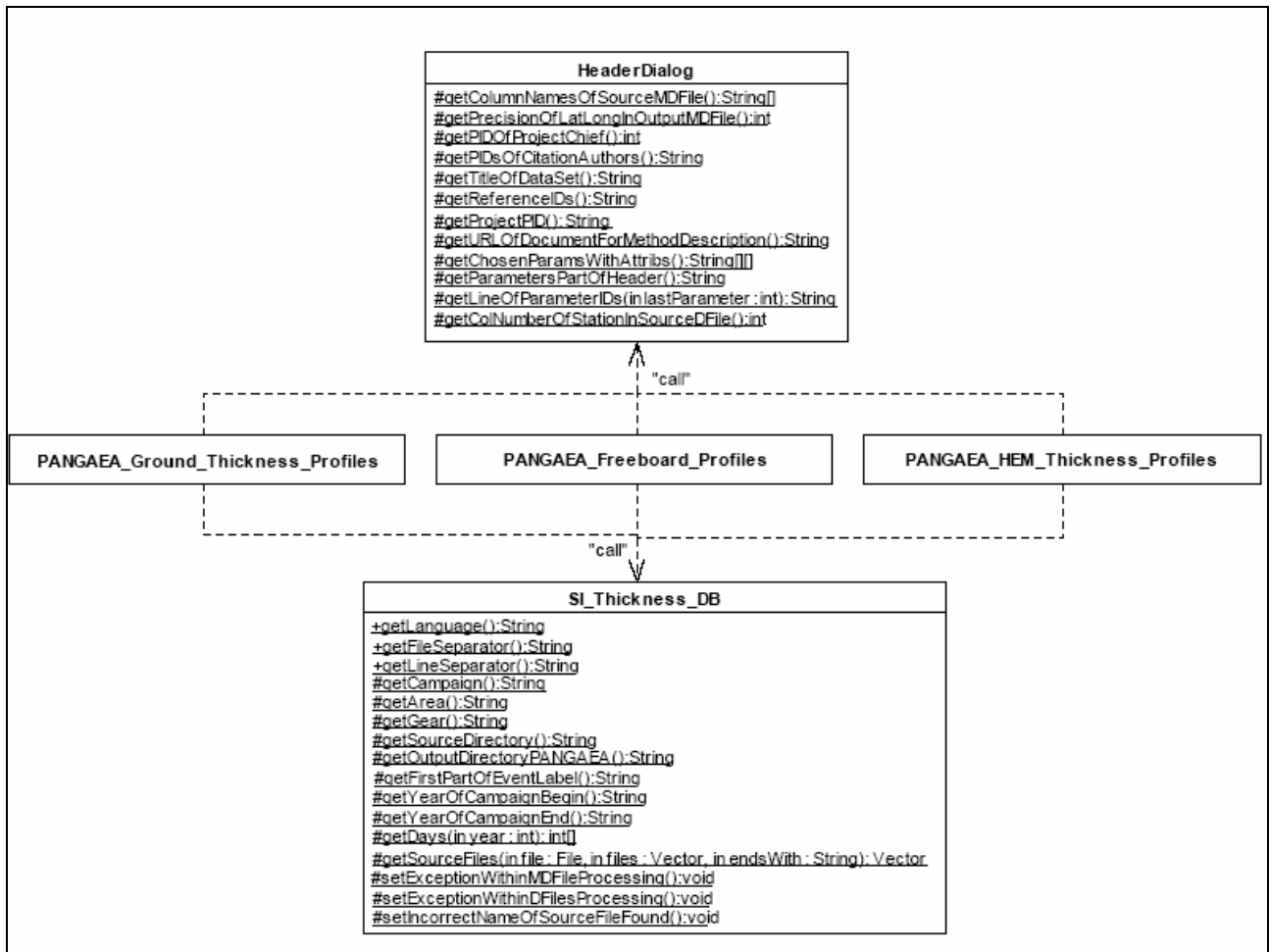


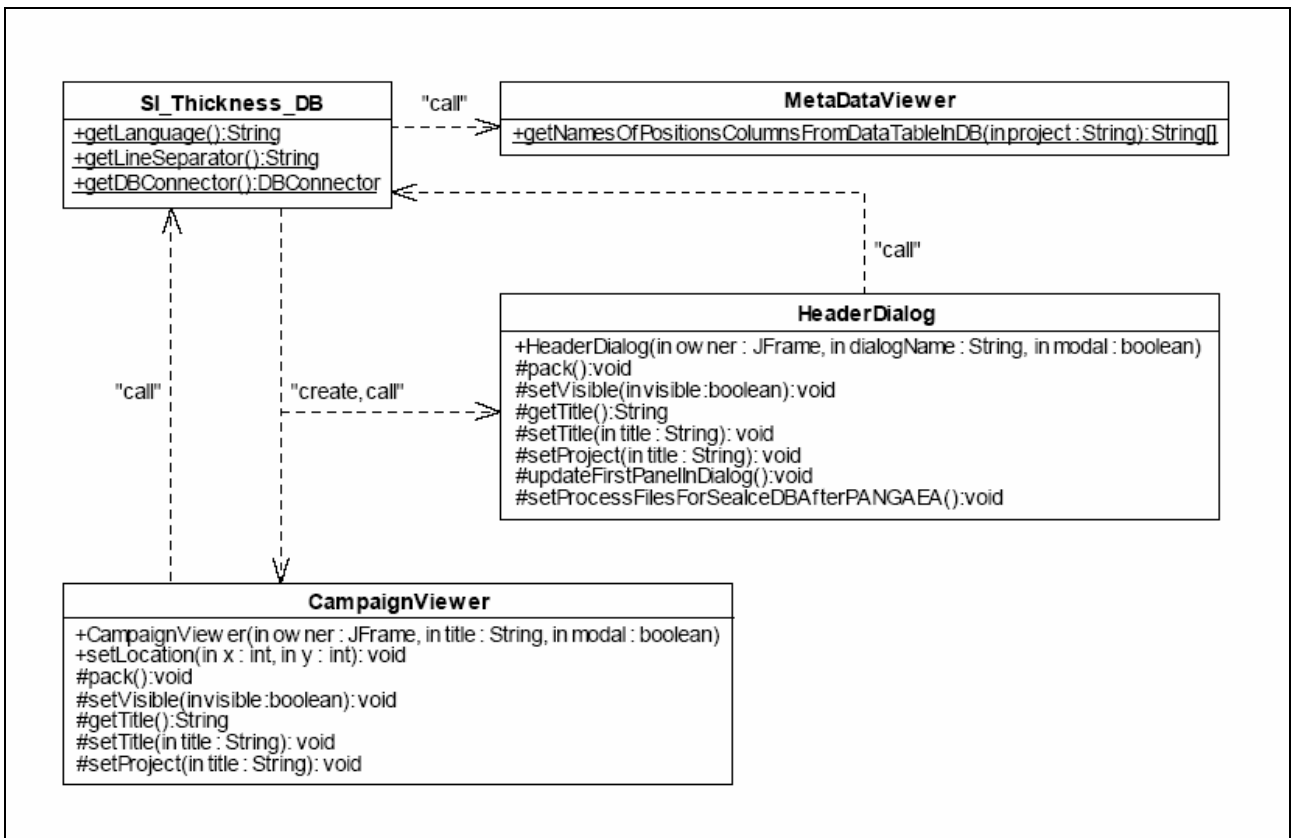
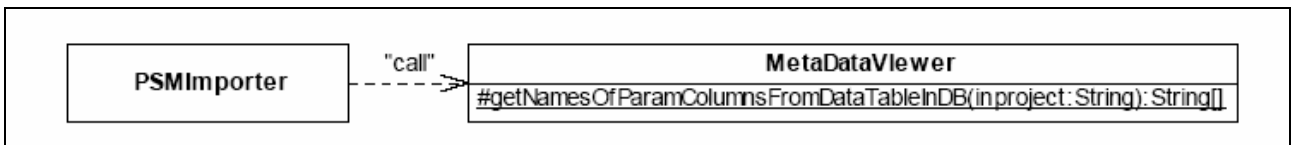
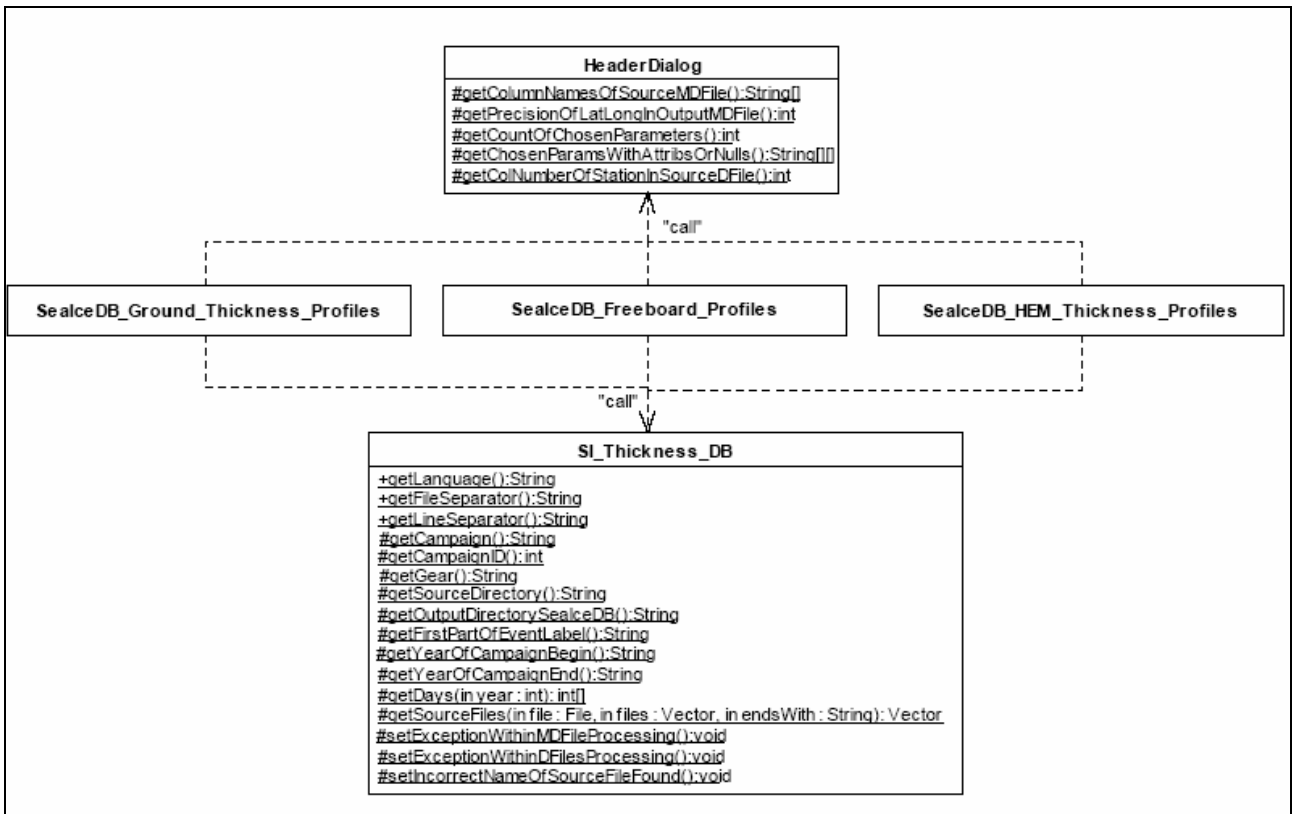




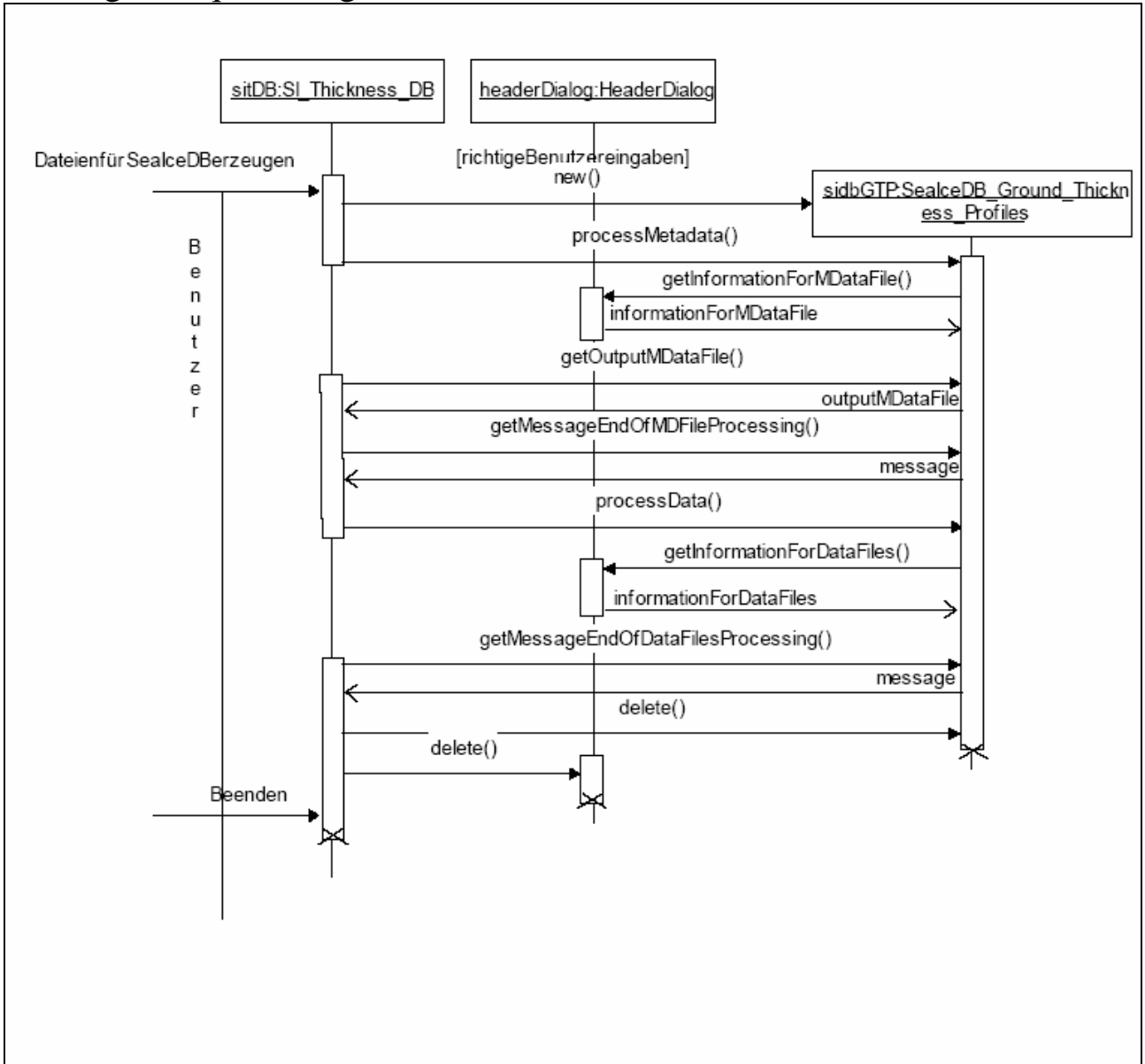


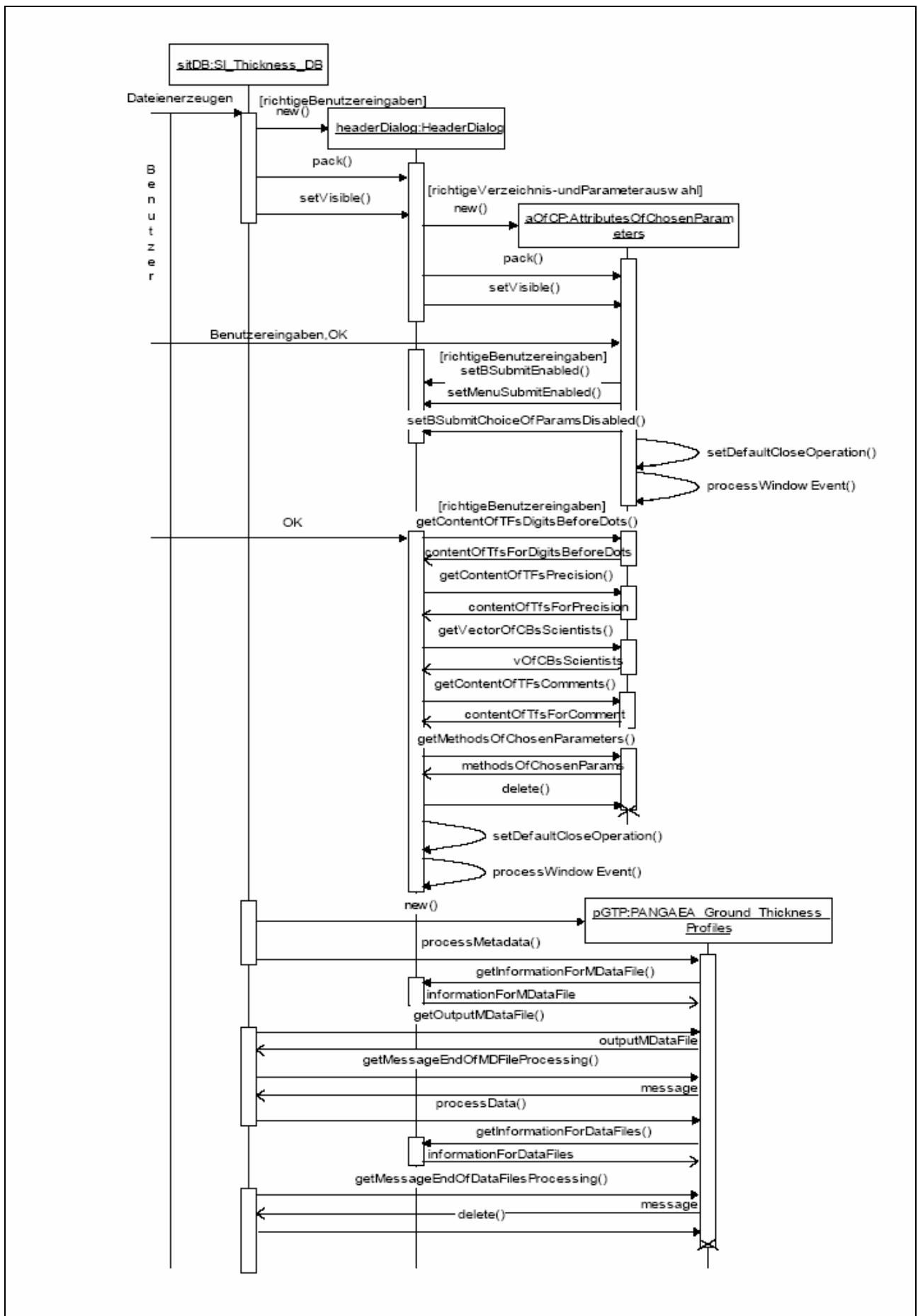




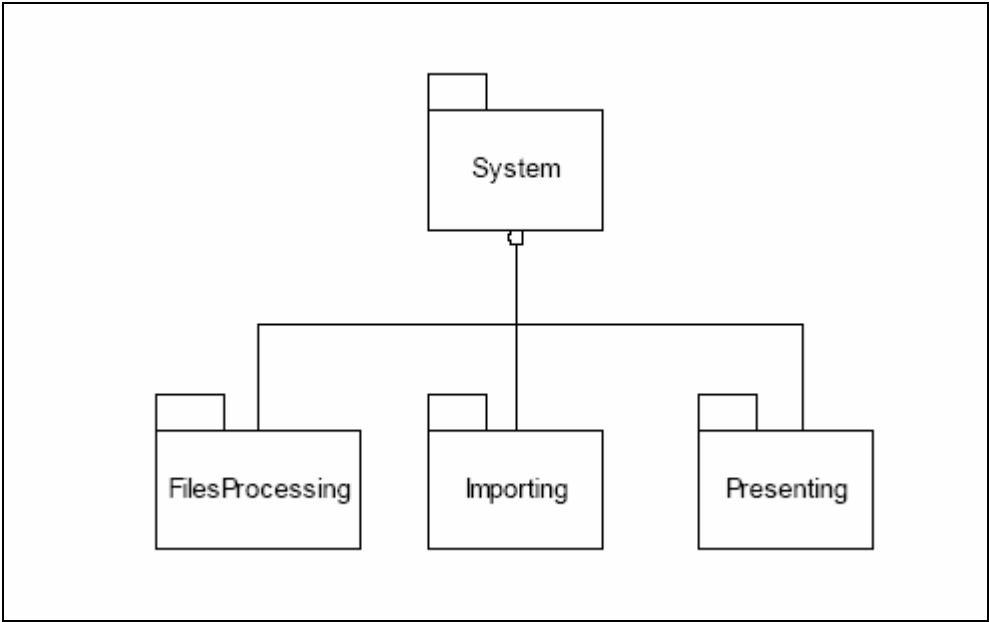


Anhang E Sequenzdiagramme

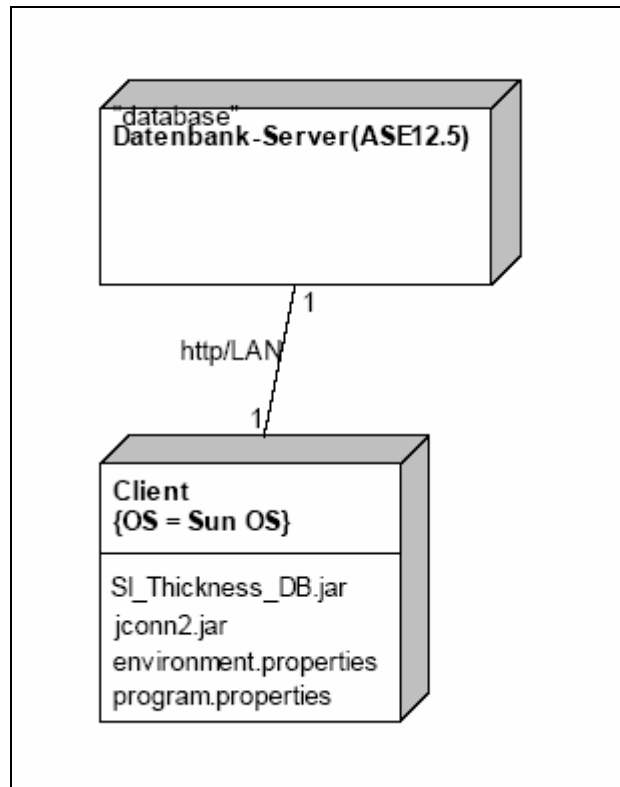




Anhang F Paketdiagramm



Anhang G Verteilungsdiagramm



Anhang H SQL-Skript

```
if exists (select 1
           from sysobjects
           where id = object_id('scientists')
           and type = 'U')
alter table scientists
drop constraint Unique_scientists

if exists (select 1
           from sysobjects
           where id = object_id('scientists')
           and type = 'U')
alter table scientists
drop constraint PK_scientists

if exists (select 1
           from sysobjects
           where id = object_id('scientists')
           and type = 'U')
drop table scientists

create table scientists (
    scientist_id int not null,
    scientist varchar(70) not null,
    constraint PK_scientists primary key (scientist_id),
    constraint Unique_scientists unique (scientist)
)

if exists (select 1
           from sysobjects
           where id = object_id('measurementPositions')
           and type = 'U')
alter table measurementPositions
drop constraint FK1_mPosition

if exists (select 1
           from sysobjects
           where id = object_id('measurementPositions')
           and type = 'U')
alter table measurementPositions
drop constraint FK2_mPositions

if exists (select 1
           from sysobjects
           where id = object_id('parameters')
           and type = 'U')
alter table parameters
drop constraint Unique2_parameters

if exists (select 1
           from sysobjects
           where id = object_id('parameters')
           and type = 'U')
alter table parameters
drop constraint Unique1_parameters
```

```

if exists (select 1
           from sysobjects
           where id = object_id('parameters')
           and type = 'U')
alter table parameters
drop constraint PK_parameters

if exists (select 1
           from sysobjects
           where id = object_id('parameters')
           and type = 'U')
drop table parameters

create table parameters (
    parameter_id int not null,
    parameter varchar(70) not null,
    short_name varchar(20) not null,
    data_type varchar(20) not null,
    constraint PK_parameters primary key (parameter_id),
    constraint Unique1_parameters unique (parameter),
    constraint Unique2_parameters unique (short_name)
)

if exists (select 1
           from sysobjects
           where id = object_id('methods')
           and type = 'U')
alter table methods
drop constraint Unique2_methods

if exists (select 1
           from sysobjects
           where id = object_id('methods')
           and type = 'U')
alter table methods
drop constraint Unique1_methods

if exists (select 1
           from sysobjects
           where id = object_id('methods')
           and type = 'U')
alter table methods
drop constraint PK_methods

if exists (select 1
           from sysobjects
           where id = object_id('methods')
           and type = 'U')
drop table methods

```

```

create table methods (
    method_id          int          not null,
    method             varchar(80) not null,
    short_name         varchar(20) not null,
    constraint PK_methods primary key (method_id),
    constraint Unique1_methods unique (method),
    constraint Unique2_methods unique (short_name)
)

if exists (select 1
           from sysobjects
           where id = object_id('measurementPositions')
           and type = 'U')
alter table measurementPositions
    drop constraint Unique_mPositions

if exists (select 1
           from sysobjects
           where id = object_id('measurementPositions')
           and type = 'U')
alter table measurementPositions
    drop constraint PK_mPositions

if exists (select 1
           from sysobjects
           where id = object_id('measurementPositions')
           and type = 'U')
drop table measurementPositions

create table measurementPositions (
    parameter_id int          not null,
    method_id    int          not null,
    position     varchar(41) not null,
    constraint PK_mPositions primary key (parameter_id, method_id),
    constraint FK1_mPositions foreign key (parameter_id)
        references parameters(parameter_id),
    constraint FK2_mPositions foreign key (method_id)
        references methods(method_id),
    constraint Unique_mPositions unique (position)
)

if exists (select 1
           from sysobjects
           where id = object_id('campaigns')
           and type = 'U')
alter table campaigns
    drop constraint Unique_campaigns

if exists (select 1
           from sysobjects
           where id = object_id('campaigns')
           and type = 'U')
alter table campaigns
    drop constraint PK_campaigns

```

```

if exists (select 1
           from sysobjects
           where id = object_id('campaigns')
           and type = 'U')
drop table campaigns

create table campaigns (
    campaign_id    int                not null,
    campaign       varchar(20)        not null,
    area           varchar(50)        not null,
    campaign_begin datetime          not null,
    campaign_end   datetime          not null,
    constraint PK_campaigns primary key (campaign_id),
    constraint Unique_campaigns unique (campaign)
)

if exists (select 1
           from sysobjects
           where id = object_id('iceThicknessData')
           and type = 'U')
alter table iceThicknessData
drop constraint FK_iceThickData_iceThickMData

if exists (select 1
           from sysindexes
           where id = object_id('iceThicknessData')
           and name = 'iceThickData_iceThickMData_FK'
           and indid > 0
           and indid < 255)
drop index iceThicknessData.iceThickData_iceThickMData_FK

if exists (select 1
           from sysobjects
           where id = object_id('iceThicknessData')
           and type = 'U')
alter table iceThicknessData
drop constraint PK_iceThicknessData

if exists (select 1
           from sysobjects
           where id = object_id('iceThicknessData')
           and type = 'U')
drop table iceThicknessData

if exists (select 1
           from sysobjects
           where id = object_id('iceThicknessMetaData')
           and type = 'U')
alter table iceThicknessMetaData
drop constraint FK_iceThicknessMetaData_camp

if exists (select 1
           from sysobjects
           where id = object_id('iceThicknessMetaData')
           and type = 'U')
alter table iceThicknessMetaData
drop constraint PK_iceThicknessMetaData

```

```

if exists (select 1
           from sysobjects
           where id = object_id('iceThicknessMetaData')
           and type = 'U')
drop table iceThicknessMetaData

create table iceThicknessMetaData (
    meta_id          int          not null,
    event_label     varchar(30) not null,
    campaign_id     int          not null,
    gear            varchar(10) not null,
    date_time       datetime     not null,
    latitude        float        not null,
    longitude       float        not null,
    remark          varchar(1024) null,
    constraint PK_iceThicknessMetaData primary key (meta_id),
    constraint FK_iceThicknessMetaData_camp foreign key (campaign_id)
        references campaigns(campaign_id)
)

create table iceThicknessData (
    data_id          int          not null,
    meta_id          int          not null,
    Distance_RT      float null,
    Ice_thickness_EM float null,
    Ice_thickness_DH float null,
    Freeboard_DH     float null,
    Draft_EM         float null,
    Draft_DH         float null,
    Snow_thickness_DH float null,
    Zsl_DH           float null,
    Zse_DH           float null,
    Zmeltpond_DH     float null,
    Zmeltpondbot_DH float null,
    constraint PK_iceThicknessData primary key (data_id),
    constraint FK_iceThickData_iceThickMData foreign key (meta_id)
        references iceThicknessMetaData (meta_id)
) on datasegment

create index iceThickData_iceThickMData_FK on iceThicknessData (meta_id ASC
) on indexsegment

if exists (select 1
           from sysobjects
           where id = object_id('laserData')
           and type = 'U')
alter table laserData
    drop constraint FK_laserData_laserMetaData

if exists (select 1
           from sysindexes
           where id = object_id('laserData')
           and name = 'laserData_laserMetaData_FK'
           and indid > 0
           and indid < 255)
drop index laserData.laserData_laserMetaData_FK

```

```

if exists (select 1
           from sysobjects
           where id = object_id('laserData')
           and type = 'U')
alter table laserData
drop constraint PK_laserData

if exists (select 1
           from sysobjects
           where id = object_id('laserData')
           and type = 'U')
drop table laserData

if exists (select 1
           from sysobjects
           where id = object_id('laserMetaData')
           and type = 'U')
alter table laserMetaData
drop constraint FK_laserMetaData_campaigns

if exists (select 1
           from sysobjects
           where id = object_id('laserMetaData')
           and type = 'U')
alter table laserMetaData
drop constraint PK_laserMetaData

if exists (select 1
           from sysobjects
           where id = object_id('laserMetaData')
           and type = 'U')
drop table laserMetaData

create table laserMetaData (
    meta_id          int                not null,
    event_label     varchar(30)         not null,
    campaign_id     int                not null,
    gear            varchar(10)         not null,
    start_date_time datetime           not null,
    start_latitude  float              not null,
    start_longitude float              not null,
    end_date_time   datetime           not null,
    end_latitude   float              not null,
    end_longitude  float              not null,
    remark         varchar(1024)        null,
    constraint PK_laserMetaData primary key (meta_id),
    constraint FK_laserMetaData_campaigns foreign key (campaign_id)
    references campaigns(campaign_id)
)

```

```

create table laserData (
    data_id          int          not null,
    meta_id          int          not null,
    Distance_Calc_from_GPS float    null,
    Freeboard_Heli_laser float    null,
    Ice_coverage_Laser_echo tinyint null,
    constraint PK_laserData primary key (data_id),
    constraint FK_laserData_laserMetaData foreign key (meta_id)
        references laserMetaData (meta_id)
)
on datasegment

```

```

create index laserData_laserMetaData_FK on laserData (
    meta_id ASC
)
on indexsegment

```

```

if exists (select 1
            from sysobjects
            where id = object_id('iceThickLaserData')
            and type = 'U')
alter table iceThickLaserData
    drop constraint FK_iceThLasData_iceThLasMData

```

```

if exists (select 1
            from sysindexes
            where id = object_id('iceThickLaserData')
            and name = 'iceThLasData_iceThLasMData_FK'
            and indid > 0
            and indid < 255)
drop index iceThickLaserData.iceThLasData_iceThLasMData_FK

```

```

if exists (select 1
            from sysobjects
            where id = object_id('iceThickLaserData')
            and type = 'U')
alter table iceThickLaserData
    drop constraint PK_iceThickLaserData

```

```

if exists (select 1
            from sysobjects
            where id = object_id('iceThickLaserData')
            and type = 'U')
drop table iceThickLaserData

```

```

if exists (select 1
            from sysobjects
            where id = object_id('iceThickLaserMetaData')
            and type = 'U')
alter table iceThickLaserMetaData
    drop constraint FK_iceThickLaserMetaData_camp

```

```

if exists (select 1
           from sysobjects
           where id = object_id('iceThickLaserMetaData')
           and type = 'U')
alter table iceThickLaserMetaData
drop constraint PK_iceThickLaserMetaData

if exists (select 1
           from sysobjects
           where id = object_id('iceThickLaserMetaData')
           and type = 'U')
drop table iceThickLaserMetaData

create table iceThickLaserMetaData (
    meta_id          int          not null,
    event_label     varchar(30) not null,
    campaign_id     int          not null,
    gear            varchar(10) not null,
    date_time       datetime     not null,
    start_latitude  float        not null,
    start_longitude float        not null,
    end_latitude    float        not null,
    end_longitude   float        not null,
    remark          varchar(1024) null,
    constraint PK_iceThickLaserMetaData primary key (meta_id),
    constraint FK_iceThickLaserMetaData_camp foreign key (campaign_id)
        references campaigns(campaign_id)
)

create table iceThickLaserData (
    data_id          int          not null,
    meta_id          int          not null,
    Latitude_GPS     float        null,
    Longitude_GPS    float        null,
    Distance_Calc_from_GPS float null,
    Fid_Ng           int          null,
    Ice_thickness_HEM float        null,
    HEIGHT_Laser     float        null,
    constraint PK_iceThickLaserData primary key (data_id),
    constraint FK_iceThLasData_iceThLasMData foreign key (meta_id)
        references iceThickLaserMetaData (meta_id)
) on datasegment

create index iceThLasData_iceThLasMData_FK on iceThickLaserData (
    meta_id ASC
) on indexsegment

```