

# Konzeptuelles Design einer raum-zeitlichen Datenbank für Tierbewegungen

**Masterarbeit GEO 511**

Universität Zürich  
Geographisches Institut  
Abteilung GIS

*Autor:*  
Sebastian SCHMIDIG

*Matrikelnummer:*  
05-713-193

*Betreuung:*  
Dr. Patrick LAUBE

Dr. Erik WILLEMS  
Universität Zürich  
Anthropologisches Institut

*Fakultätsvertreter:*  
Prof. Dr. Robert WEIBEL

eingereicht am 31.01.2012

**Kontakt:**

Sebastian Schmidig

Gerbi 38

CH-8752 Näfels

sschmidig@gmx.ch

+41 79 258 55 65

# Danksagung

Zum Beginn dieser Masterarbeit möchte mich zuerst bei allen Leuten bedanken, welche mir beim Abschluss des Studiums und insbesondere dieser Arbeit geholfen haben.

Besonderer Dank gilt dabei meinem GIUZ-internen Betreuer Patrick Laube für dessen kompetente Unterstützung und Ratschläge. In zahlreichen Sitzungen brachte er Kritik an und gleichzeitig neue Ideen ins Spiel, womit er viel zum Gelingen dieser Masterarbeit beitrug.

Weiter bedanke ich mich bei meinem zweiten Betreuer Erik Willems für die gute Zusammenarbeit. In einigen Sitzungen und vielen Emails half er mir das *Inkawu Vervet Project* näher zu bringen und beantwortete geduldig meine zahllosen Fragen. Ausserdem möchte ich mich für die guten Vorschläge des Datenbankexperten Rolf Meile zu Beginn der Arbeit bedanken.

Mein grösster Dank gilt meinen Eltern, welche mir das Studium erst erlaubt und mich während der ganzen Zeit unterstützt haben – moralisch und auch finanziell. Ganz besonders möchte ich mich auch bei Dustin Gygli für das aufmerksame Gegenlesen der Arbeit bedanken. Ausserdem danke ich meinen Mitstudenten für die wertvollen Tipps während dem Verfassen dieser Arbeit.





# Zusammenfassung

Schon seit den Anfängen der Ökologie Mitte des 19. Jahrhunderts war den Forschern bewusst, dass Tiere nicht zufällig in der Wildnis umherstreifen, sondern dass Bewegung immer einem Zweck dient. Durch die Analyse der Bewegungsmuster kann auf ökologische Prozesse geschlossen werden, sie stellen die Verbindung zwischen Verhalten, Umwelt und Populationsdynamik dar. Mit dem Aufkommen neuer Technologien wie VHF und GPS wurde die Art und Weise der Datenerfassung grundlegend verändert und damit der Standpunkt der Forschung von den Forschern zu den Tieren selbst verschoben. Für das Testen von wissenschaftlichen Hypothesen zur Gewinnung neuer Einsichten, sollten die rohen Positionsdaten jedoch mit Feldbeobachtungen, Experimenten und Informationen zur Umwelt ergänzt werden. Dadurch entstehen über die Jahre eines Langzeitforschungsprojektes zahlreiche Daten in unterschiedlichsten Formaten, deren Management und Analyse zusehends komplexer wird. Für eine konsistente Speicherung solcher Informationen eignet sich am besten eine Datenbank, wo Daten strukturiert und mit Integritätsbedingungen versehen werden können. Das Handling von Bewegungsdaten mit räumlichen und zeitlichen Komponenten in einer Datenbank gestaltet sich jedoch als nicht trivial. Während von Seiten der Datenbankforschung diverse konzeptuelle und einige Prototypen präsentiert werden, fehlt es nach wie vor an einem in der Praxis anwendbaren System, welches beispielsweise auch in der bewegungsökologischen Forschung eingesetzt werden könnte.

In dieser Arbeit wird an einem ökologischen Forschungsprojekt aufgezeigt, wie mit einer handelsüblichen Datenbank und zugehörigen räumlichen Tools Bewegungsdaten gehandhabt werden können. Dazu wurden anhand des üblichen Datenbankerstellungsprozesses die vorhandenen Daten sowie das jetzige Datenmanagement untersucht und dabei der Fokus auf die räumlichen, zeitlichen und raum-zeitlichen Komponenten gelegt. Verbesserungen für die aufgedeckten Schwächen in der raum-zeitlichen Struktur wurden anhand eines relationalen Datenbankmanagementsystems (DBMS) mit räumlicher Erweiterung und einem kleinen Datensatz demonstriert. Dabei wurde das Potential des vom Forschungsprojekt erfassten Datensatzes mit einigen Beispielen zu möglichen raum-zeitlichen Anreicherungen hervorgehoben. Eine stärker vernetzte Datenstruktur, einige Möglichkeiten zur Automatisierung von Prozessen in der Datenbank, sowie die Verwendung von räumlichen Datentypen (*spatial data types*) wurden mit visualisierten Beispielen präsentiert. Zudem wurde gezeigt, dass auch komplexere raum-zeitliche Abfragen zu Beziehungen zwischen unterschiedlichen Objekten in Bewegung mit SQLs (*structured query language*) direkt auf der Datenbank gestellt und beantwortet werden können. Räumliche und zeitliche Attribute eines Objektes sind getrennt gespeichert und müssen deshalb auch bei der Formulierung von raum-zeitlichen Queries separat adressiert werden, was mit zunehmender Komplexität der Abfragen stark verschachtelte SQLs zur Folge hat. Für solche vielschichtigen Abfragen und deren Visualisierung eignen sich Geographische Informationssysteme (GIS) besser und sollten deswegen in Kombination mit Datenbanken für solche ökologische Projekte verwendet werden.



# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>ix</b>
<b>SQL-Verzeichnis</b>	<b>xi</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Wissenschaftliche Motivation . . . . .	1
1.2 Forschungsfragen . . . . .	2
1.3 Ziele . . . . .	2
1.4 Struktur der Arbeit . . . . .	3
<b>2 Beschrieb Inkawu Vervet Projekt</b>	<b>5</b>
<b>3 Theoretischer Hintergrund</b>	<b>7</b>
3.1 Raum- und Bewegungsinformation in der Bewegungsökologie . . . . .	7
3.2 Bewegungsdaten . . . . .	8
3.3 Erfassungstechniken für Bewegungsdaten . . . . .	9
3.4 Anforderungen an Managementsystem für Wildtierdaten . . . . .	10
3.5 Speicherung von Bewegungsdaten . . . . .	12
3.5.1 Datenbanken . . . . .	14
3.5.2 Räumliche Datenbanken . . . . .	14
3.5.3 Zeitliche Datenbanken . . . . .	16
3.5.4 Raum-Zeitliche Datenbanken . . . . .	17
3.5.5 Moving Objects Databases . . . . .	20
3.6 Analyse von Bewegungsdaten . . . . .	23
3.6.1 Bewegungsanalyse in der Ökologie . . . . .	25
3.6.2 Abfragen zu raum-zeitlichen Objekten . . . . .	27
3.6.3 Abfragen in Moving Objects Databases . . . . .	29
3.7 Anwendungen . . . . .	30
3.7.1 MOD-Forschungsprototypen . . . . .	31
3.7.2 Anwendungen in der Bewegungsökologie . . . . .	31
3.8 Forschungslücken . . . . .	36
<b>4 Methodik</b>	<b>39</b>
4.1 Anforderungsanalyse . . . . .	39
4.2 Datenbankentwurf . . . . .	40
4.2.1 Konzeptueller Entwurf . . . . .	41
4.2.2 Logisches Datenbankmodell . . . . .	41
4.3 Analyse der raum-zeitlichen Informationsstruktur . . . . .	42
4.4 Raum-zeitliche Anreicherung . . . . .	42

<b>5</b>	<b>Anforderungsanalyse</b>	<b>43</b>
5.1	Forschungskonzept Inkawu Vervet Monkey Project . . . . .	43
5.2	Struktur der existierenden Daten . . . . .	44
5.3	Sitzungen mit Kontaktperson . . . . .	47
5.4	Abfragen . . . . .	48
<b>6</b>	<b>Datenbankentwurf</b>	<b>51</b>
6.1	Konzeptueller Entwurf . . . . .	51
6.2	Logisches Datenmodell . . . . .	55
<b>7</b>	<b>Raum-zeitliche Informationen im IVP</b>	<b>59</b>
7.1	Räumliche Informationen . . . . .	59
7.2	Zeitliche Informationen . . . . .	59
7.3	Raum-Zeitliche Informationen . . . . .	60
<b>8</b>	<b>Raum-zeitliche Anreicherung</b>	<b>63</b>
8.1	Zuweisung der rohen GPS-Daten an Individuen . . . . .	63
8.1.1	Handhabung der Gruppeninformation . . . . .	67
8.2	Ortszuweisung regelmässiger Beobachtungen . . . . .	68
8.3	Analyse von Gruppen-Trajektorien . . . . .	71
8.4	Anreicherung mit Kontextinformation . . . . .	73
<b>9</b>	<b>Diskussion</b>	<b>79</b>
9.1	Verbesserung des Datenmanagements durch Datenbankentwurf . . . . .	79
9.2	ER für Modellierung raum-zeitlicher Komponenten . . . . .	80
9.3	Handling von Bewegungsdaten in SRDBMS . . . . .	81
9.4	Raum-zeitliche Fragen in SRDBMS . . . . .	83
9.5	Zielerreichung und Einschränkungen . . . . .	84
9.6	Empfehlungen zur Implementation im Inkawu Vervet Projekt . . . . .	84
<b>10</b>	<b>Schlussfolgerung</b>	<b>87</b>
10.1	Zusammenfassung . . . . .	87
10.2	Wissenschaftlicher Beitrag . . . . .	88
10.3	Ausblick . . . . .	88
	<b>Literaturverzeichnis</b>	<b>91</b>
<b>A</b>	<b>Anforderungstext</b>	<b>95</b>
<b>B</b>	<b>Entity Relation Model</b>	<b>99</b>
	<b>Persönliche Erklärung</b>	<b>105</b>

# Abbildungsverzeichnis

2.1	Lage des Forschungsgeländes und Abbild eines <i>vervet monkey</i> . . . . .	5
3.1	<i>Geospatial lifeline</i> , bestehend aus mehreren <i>fixes</i> . . . . .	8
3.2	<i>Trajectory</i> als semantische Segmentierung . . . . .	8
3.3	Zeitliche Snapshots . . . . .	12
3.4	Objektänderungen von raum-zeitlichen Daten . . . . .	13
3.5	Topologische Beziehungen nach <i>9-Intersection Model</i> . . . . .	15
3.6	Mögliche Veränderungen von raum-zeitlichen Objekten . . . . .	17
3.7	Relationales Schema am Beispiel der Storchmigration . . . . .	22
3.8	Relationales Datenmodell zu Abbildung 3.7 . . . . .	22
3.9	<i>Primitive patterns</i> von Bewegungsdaten . . . . .	24
3.10	<i>Compound patterns</i> von Bewegungsdaten . . . . .	25
3.11	<i>Kernel density</i> von <i>moving objects</i> . . . . .	26
3.12	Unsicherheit in Bewegungsdaten . . . . .	30
3.13	Ausschnitt von Datenbankschema für Wildtierdaten . . . . .	33
3.14	GUI von <i>R</i> mit Klasse <i>ltraj</i> . . . . .	34
3.15	<i>Movebank</i> . . . . .	35
5.1	<i>Data collection protocol</i> Inkawu Vervet Projekt . . . . .	44
5.2	<i>Access</i> -Datenbank Inkawu Vervet Projektes . . . . .	45
6.1	Farbcode des <i>entity relationship models</i> . . . . .	51
6.2	ER LOG und GROUP_CHANGES . . . . .	52
6.3	ER BETWEEN_GROUP_ENCOUNTER . . . . .	53
6.4	Mögliche ER-Struktur der Vegetationskarte (VEGETATION_MAP) . . . . .	54
6.5	Logisches Modell um LOG und GROUP_CHANGES . . . . .	55
6.6	View von GROUP_CHANGES . . . . .	56
6.7	Logisches Modell rund um BETWEEN_GROUP_ENCOUNTER . . . . .	57
6.8	Logisches Modell für zeitlich variierende Vegetationskarte . . . . .	58
7.1	Schematische Darstellung der raum-zeitlichen Daten vom IVP . . . . .	61
8.1	ER-Struktur für automatische Zuordnung der GPS-Daten . . . . .	64
8.2	Automatische Zuweisung der rohen GPS-Daten an Individuen . . . . .	66
8.3	Entitätstyp GROUP_TRAJECTORY . . . . .	68
8.4	Problemskizze zu automatischer Ortszuweisung . . . . .	70
8.5	Detektion von Gruppenkonflikten . . . . .	72
8.6	Kontextinformation bei OBS_SCAN-Daten . . . . .	75
8.7	<i>Between group encounter</i> mit Kontextinformation . . . . .	76

9.1	Automatische Zuweisung der <i>handheld</i> -GPS-Daten . . . . .	85
-----	---	----

# SQL-Verzeichnis

3.1	Beispiel für eine Abfrage einer räumlichen Datenbank. . . . .	15
3.2	Beispiel für eine Query in einer zeitlichen Datenbank. . . . .	16
3.3	Beispiel für eine Query in einer raum-zeitlichen Datenbank. . . . .	19
3.4	Beispiel für eine Query mit sich bewegenden Objekten. . . . .	21
3.5	Beispiels-Query mit FTL . . . . .	30
3.6	Zuweisung der rohen GPS-Daten an Individuen im ISAMUD-System. . . .	33
6.1	Veränderungen sowie Zusammensetzung ausgewählter Gruppe . . . . .	56
6.2	Änderungen in Gruppenzusammensetzung während Zeitperiode . . . . .	56
8.1	Automatische Zuweisung der rohen GPS-Daten an Individuen . . . . .	65
8.2	Abfrage von geographischer Länge und Breite der rohen GPS-Daten. . . . .	67
8.3	Trajektorie einer Gruppe an ausgewähltem Datum . . . . .	68
8.4	Automatische Lokalisierung von OBS_SCAN über Zeitgleichheit . . . . .	69
8.5	Automatische Lokalisierung von OBS_SCAN mit 15min Buffer . . . . .	69
8.6	Detektion von Gruppenzusammenstößen . . . . .	71
8.7	Suche von Kontextinformation bei OBS_SCAN-Daten . . . . .	74
8.8	Abfrage der MONKEY_LOCATIONS nach Kontextinformationen . . . . .	75
8.9	Detektion von Gruppenkonflikten inkl. Vegetationsinfo . . . . .	76





# Kapitel 1

## Einleitung

### 1.1 Wissenschaftliche Motivation

Bei der Verhaltensforschung an Tieren in Ökologie und Biologie ist die räumliche Komponente von grosser Bedeutung, das Verständnis von *wie* und *wieso* der Tierbewegungen ist das Ziel eines mechanistischen Ansatzes. Durch das Studium von benutzten Ressourcen, Interaktionen zwischen Artgenossen, Konkurrenz- und Fortpflanzungsverhalten kann auf die Fitness der Tiere geschlossen werden. Die Bewegung der einzelnen Individuen verbindet diese Prozesse, wodurch deren Modellierung fundamental für alle ökologischen Fragen ist (Cagnacci *et al.*, 2010).

Um Tiere in freier Wildbahn zu untersuchen und wissenschaftlich Hypothesen zu testen, braucht es zahlreiche Beobachtungen und Experimente im Feld, wobei generell eine sehr grosse Menge an Daten anfällt. Diese Datenmenge wird in der aktuellen Forschung an Wildtieren nochmals immens vergrössert, da die Beobachtungen mit Positionsdaten von modernen Positionierungsmethoden ergänzt werden.

Diese, für ökologische Analysen nötigen, Daten werden aus verschiedenen Quellen zusammengetragen und besitzen oftmals sehr unterschiedliche Eigenschaften. Im Feld werden auf *personal digital assistants* (PDAs) Beobachtungen einerseits stichwortartig vermerkt, andererseits aus vorgefertigten Lösungsmöglichkeiten ausgewählt. Meist werden sie automatisch mit einem Zeitstempel versehen, technisch ist auch eine automatische Lokalisierung der Notiz möglich. Aktuelle Fernmessmethoden wie *global positioning system* (GPS) ermöglichen reichhaltige raum-zeitliche Informationen zu Tieren, welche drahtlos heruntergeladen werden können.

Die Speicherung und Analyse dieser grossen Datenmenge ist nicht einfach, denn für ökologische Studien ist eine Datenkonsistenz über mehrere Jahre notwendig, damit die gesammelten Fakten der früheren Jahre wegen Langzeitbeobachtung einfach zugänglich sind. Mit einer Datenbank lässt sich diese Aufgabe am besten erfüllen (Urbano *et al.*, 2010) und zusätzlich kann eine vernetzte Speicherung der unterschiedlichsten Datenformate angestrebt werden.

Da für eine ganzheitliche Untersuchung gewisser Verhaltensweisen die Bewegung der Tiere entscheidend ist, muss diese in der Datenbank eingeschlossen werden (Cagnacci *et al.*, 2010; Morales *et al.*, 2010). Die Speicherung und Abfrage von Objekten in Bewegung stellt eine grosse Herausforderung dar und herkömmliche Datenbankmanagementsysteme (DBMS) bieten nur einen beschränkten Support. Die Integration von Bewegungsdaten mit komplexen raum-zeitlichen Strukturen benötigt deshalb einen gut ausgearbeiteten Entwurf, denn die im Voraus festgelegten Entitäten, Attribute, Relationen und Einschränkungen stellen die Grundlage der Datenbank dar.

## 1.2 Forschungsfragen

Im Hinblick auf eine vereinfachte Analyse sollen mögliche Datenstrukturen und Methoden untersucht werden. Die in dieser Masterarbeit untersuchten Forschungsfragen behandeln das *Handling* von Bewegungsdaten aus der Forschung an Tieren in freier Wildbahn:

- 1) Wie kann der generelle Datenbankentwurf einem bereits bestehenden biologischen Forschungsprojekt mit räumlichen, zeitlichen und raum-zeitlichen Fragen helfen, das existierende Datenmanagement zu verbessern?
- 2) Ist das *entity relationship model* geeignet um die Datenstruktur eines bewegungsökologischen Forschungsprojektes mit räumlichen, zeitlichen und raum-zeitlichen Komponenten abzubilden und dessen Implementierung vorzubereiten?
- 3) Welches sind die Vorteile im Hinblick auf die Speicherung und Analyse von Bewegungsdaten aus der Wildtierforschung mit einem räumlichen, relationalen Datenbankmanagementsystem (*spatial relational database management system* SRDBMS) gegenüber einem *file*-basierten Ansatz?
- 4) Welche, für ein ökologisches Forschungsprojekt interessanten, raum-zeitlichen Fragen betreffend Bewegungen der Tiere können mit Hilfe eines handelsüblichen, räumlichen, relationalen Datenbankmanagementsystems (*spatial relational database management system* SRDBMS) untersucht werden?

## 1.3 Ziele

Die hier genannten Ziele stellen annähernd den Weg dar, wie in dieser Masterarbeit vorgegangen wird und die oben erwähnten Forschungsfragen bearbeitet werden sollen:

- 1) Durchführen einer Anforderungsanalyse des *Inkawu Vervet Projects*, ausarbeiten was in der Datenbank kommen soll.
- 2) Verfassen von raum-zeitlichen Queries, welche im behandelten Projekt verwendet werden könnten.
- 3) Erstellung eines konzeptuellen Modells für die jetzige Situation des Inkawu Vervet Projekts; ganzheitliches ER mit 'allen' Entitäten und Relationen, ausgewählte Elemente im logischen Modell.
- 4) Aufzeigen der jetzigen Struktur der raum-zeitlichen Daten.
- 5) Erstellung einer relationalen Datenbank mit einem kleinen Set an fiktiven Bewegungsdaten, wie sie ungefähr im Projekt vorkommen könnten.
- 6) Darstellung der Möglichkeiten eines relationalen Datenbankmanagementsystems im Handling von Bewegungsdaten anhand einiger Beispiele mit räumlichen und zeitlichen Datentypen sowie Abfragemethoden.

## 1.4 Struktur der Arbeit

Grundsätzlich ist diese Masterarbeit in vier Teilbereiche gegliedert. Im ersten Teil werden die theoretischen Grundlagen vermittelt: Einem kurzen Beschrieb des bearbeiteten Forschungsprojektes folgt der theoretische Teil, wo auf die Möglichkeiten zur Speicherung und Analyse von Bewegungsdaten eingegangen wird. Im Anschluss an die Vorstellung einiger Anwendungen aus der Datenbankforschung sowie der Bewegungsökologie werden die Forschungslücken präsentiert und das genaue methodische Vorgehen in dieser Arbeit erläutert.

Im nächsten Teil folgt die Anforderungsanalyse und der Datenbankentwurf, welcher den konzeptuellen als auch den logischen Entwurf beinhaltet. Darin wird spezifisch auf die Situation des Forschungsprojektes eingegangen und die jetzige Datenstruktur auf räumliche, zeitliche sowie raum-zeitliche Komponenten untersucht, um Schwachpunkte als auch denkbare Verbesserungsmöglichkeiten zu detektieren.

Im dritten Teil wird auf das Potential der vorhandenen Bewegungsdaten eingegangen, indem mittels einer Datenbanklösung verschiedene Methoden zur räumlichen Anreicherung präsentiert werden. Dies wird anhand von vier konkret ausgearbeiteten Beispielen an einem kleinen Datensatz erreicht.

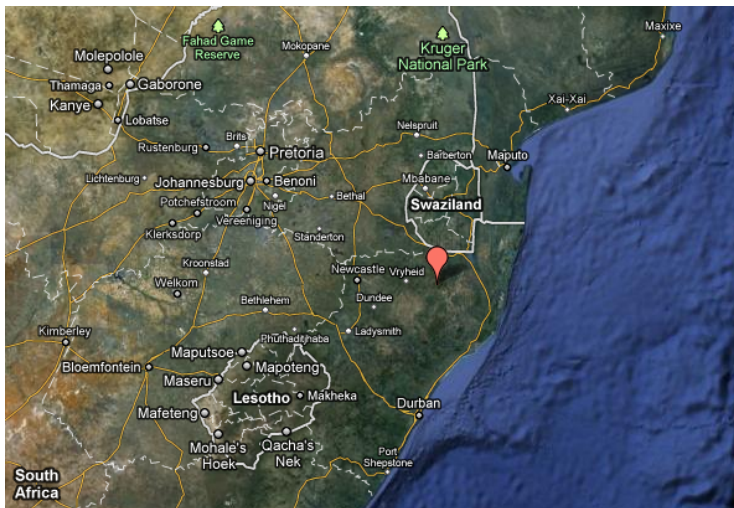
Im letzten Teil werden die anfangs gestellten Forschungsfragen und Ziele aufgrund der Ergebnisse sowie der vorhandenen Literatur diskutiert und aufgetretene Schwierigkeiten erwähnt. Die Arbeit wird mit einer Zusammenfassung und einem Ausblick abgeschlossen.



## Kapitel 2

# Beschrieb Inkawu Vervet Projekt

Diese Arbeit beschäftigt sich mit dem *Inkawu Vervet Project*, einem gemeinsamen, ökologischen Forschungsprojekt verschiedener Universitäten (Neuenburg, Zürich, St.-Andrews, Kapstadt). Die vertretenen Forscher kommen aus unterschiedlichen Bereichen von Biologie, Anthropologie, Zoologie, Psychologie sowie Ökologie und untersuchen anhand von *vervet monkeys* (grünen Meerkatzen, Gattung *Chlorocebus*, siehe auch Abbildung 2.1(b)) verschiedene Verhaltensmuster von Individuen wie auch von Gruppen. Stationiert im Osten Südafrikas (siehe Abbildung 2.1(a)) werden auf einem Privatgrundstück von rund 12'000 Hektaren mehrere Gruppen dieser höchst sozialen Tierart erforscht. Die Forscher interessiert vor allem der Zusammenhang zwischen Konflikten einzelner Gruppen und der Kooperation innerhalb einer Gemeinschaft von *vervets*. Ziel ist es, Rückschlüsse auf evolutionäre Ursprünge von Kriegshandlungen und Selbstlosigkeit im menschlichen Verhalten zu gewinnen.



(a) Lage des Forschungsgeländes



(b) Vervet monkey

Abbildung 2.1: Lage des Forschungsgeländes, *maps.google.ch*; Grüne Meerkatze, Quelle: Erik Willems.

Kriegshandlungen, von Menschen oder von Primaten, sind eine extreme Form von Konkurrenzkampf zwischen zwei Gruppen, dessen Ursachen und Konsequenzen zur Zeit nur beschränkt verstanden werden. Zum Teil liegt dies an empirischen Problemen, da mehrere Gruppen schwierig zu verfolgen und Zusammenstöße relativ selten sind. Die zweite Schwierigkeit liegt in der Komplexität der Angelegenheit:

Der grundsätzliche ökonomische Ansatz sagt Territorialität voraus, sobald der Nutzen des Zugangs zu einer Ressource die Kosten dessen Verteidigung aufheben. Komplikationen tauchen jedoch wegen unterschiedlichen Rollen der Geschlechter auf. Während Weibchen grundsätzlich am meisten an der Verteidigung von Nahrungsressourcen interessiert sein sollten, beschäftigt die Männchen generell die Verteidigung des sexuellen Zugangs zu den Weibchen. Doch männliche Tiere können sich auch an nahrungsbedingten Konflikten beteiligen, falls damit das Überleben der Nachkommen gesichert wird und die eigenen Gene weiter bestehen. Männchen, welche sich an Konflikten um Nahrungsressourcen zum Zweck des eigenen Fortpflanzungserfolges beteiligen, werden als *hired guns* bezeichnet.

Allgemein führen kollektive Handlungen von Mitgliedern innerhalb der Gruppen eher zu erfolgreichen Kämpfen. Kooperation in einer Gemeinschaft führt also zu einem Allgemeinwohl, wobei nicht teilnehmende Individuen mit einer höheren Fitness wegen Nichtkämpfens belohnt werden und damit das kooperative Verhalten untergraben. Demzufolge sollte starke Konkurrenz zwischen Gruppen zu Kooperation innerhalb Gruppen führen, das oben genannte Problem der kollektiven Handlungen (*collective action problem*) sollte jedoch Konflikte eher verhindern (van Schaik, Willems und Krützen, unveröffentlicher Bericht).

Diese Punkte sollen mittels einer starken Zusammenarbeit zwischen verschiedenen Forschern sowie einem grossen Stichprobenumfang von ungefähr zehn *vervet*-Gruppen adressiert werden. Da Grüne Meerkatzen in überschaubaren Aktionsräumen von wenigen Quadratkilometern leben, eignen sich diese gut für umfassende Beobachtungen. Bereits seit 2009 werden Daten zu diversen Eigenschaften der Individuen aber auch der Gruppen sowie der Umgebung aufgenommen. Kontinuierliche Beobachtungen zum Verhalten einzelner Tiere, wie z.B. deren Nahrungseinnahme, sowie auch auftretende Ereignisse wie Verletzungen oder Alarmrufe werden zeit- und ortsbezogen per PDA und *handheld*-GPS-Geräte als Punktdaten aufgenommen. Pro Gruppe ist zudem mindestens ein Individuum mit einem GPS-Empfänger ausgerüstet, womit regelmässig Positionen übertragen werden und damit die Territorien aller Gruppen zu verschiedenen Zeiten verfügbar sind. Eine Vegetationskarte, sowie aus verschiedenen Satellitenbildern abgeleitete Werte zu Habitatsproduktivität und Verfügbarkeit von Nahrung, werden als Polygon- und Rasterfiles zu Analysezwecken als Basis erstellt. Weiter werden Experimente durchgeführt, wobei z.B. mit geschickt platzierter Nahrung zu verschiedenen Jahreszeiten Gruppenzusammenstösse (*between group encounter*) ausgelöst oder durch Repräsentationen von natürlichen Feinden Gruppenkooperation hervorgerufen werden. Die genutzte *Microsoft Access*-Datenbank bietet relativ simple Abfrage- und Analysemöglichkeiten, mit *R* können zudem kompliziertere statistische Untersuchungen durchgeführt und Visualisierungen erstellt werden.

## Kapitel 3

# Theoretischer Hintergrund

Diese Arbeit ist im interdisziplinären Forschungsgebiet von GIScience, Datenbanken und Ökologie situiert, wobei in letzterem der Fokus auf die Verhaltensforschung an Tieren in freier Wildbahn gelegt wird. In den folgenden Kapiteln soll aufgezeigt werden, welchen Stellenwert die räumlichen und v.a. die Bewegungs-Informationen in der biologischen und ökologischen Verhaltensforschung haben und wie diese gehandhabt wird. Die Möglichkeiten von Datenbanken sollen im Bereich Speicherung und Analyse von räumlichen, zeitlichen und raum-zeitlichen Daten untersucht werden, womit der Schwerpunkt auf *spatio-temporal databases* sowie *moving objects databases* (MOD's) liegen wird.

### 3.1 Raum- und Bewegungsinformation in der Bewegungsökologie

Viele angewandte Fragen der Umweltforschung werden in einem räumlichen Kontext gestellt (Morales *et al.*, 2010). Können Populationen von Tieren in den geplanten Schutzzonen erhalten werden? Wie schnell werden gerade eben geschützte Habitate besiedelt? Widerspiegelt sich der Klimawandel im Migrationsverhalten gewisser Zugvögel?

Das Wissen, wohin Tiere gehen, hilft den Forschern die Schlüsselkonzepte der Ökologie besser zu verstehen (Cagnacci *et al.*, 2010). Dazu gehört die Benutzung der verfügbaren Ressourcen durch die Tiere, ihre Habitate und wechselnde Verteilung in einer sich immer ändernden Umwelt. Im Endeffekt führen die Reaktionen von Individuen auf veränderte Verteilung von Ressourcen zu Änderungen im Verhalten und letztendlich zeigt sich dies in der Dynamik einer ganzen Population. Die Bewegungsmuster von Tieren in freier Wildnis sind also die Verbindung zwischen Verhalten, Umwelt und Populationsdynamik (Morales *et al.*, 2010).

Um eine Population nicht anhand einzelner Beobachtungen von Geburten, Todeszahlen, Immigrationen und Emigrationen, sondern mit Hilfe moderner *Tracking*-Technologien zu studieren, werden geeignete Hilfsmittel benötigt. Neben geeigneten Algorithmen und statistischen Modellen für die Analyse, bedarf es eines ausgearbeiteten Managementsystems für die Bewegungsdaten aus Positionierungsmethoden. Doch was ist Bewegung genau? Wie wird diese gespeichert?

### 3.2 Bewegungsdaten

Andrienko *et al.* (2008, 4) definieren Bewegung als '*change in the physical position of an entity with respect to some reference system within which one can assess positions. Most frequently, the reference system is geographical space.* Dodge *et al.* (2008) fügen bei, dass auch die Änderung geometrischer Attribute, wie z.B. Ausdehnung, ein sich bewegendes Objekt festlegen kann. Da die Dimension eines sich bewegendes Objektes in vielen Anwendungen von kleinerer Bedeutung ist, wird jedoch oft der Begriff *Moving Point Object* (MPO) verwendet, dessen Weg durch Zeit und Raum als *trajectory* bezeichnet wird.

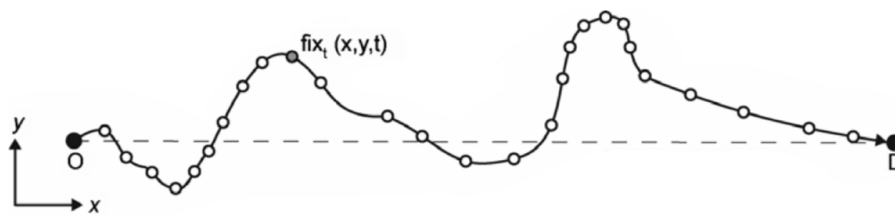


Abbildung 3.1: *Geospatial lifeline* eines Objektes aus zahlreichen *fixes* von Startpunkt O zu Endpunkt D, aus Laube *et al.* (2007).

Die an sich kontinuierliche Bewegung eines Objektes wird bei der Messung diskretisiert und meist als Abfolge von sogenannten *fixes* in der Form  $(x,y,(z),t)$  gespeichert, wobei  $x$ ,  $y$  und  $z$  die Position im Raum und  $t$  den Zeitpunkt repräsentiert (Laube *et al.*, 2007). Eine Sammlung von Positionen (bzw. *fixes*) eines Individuums über eine bestimmte Zeitperiode wird dann als *geospatial lifeline* (Abbildung 3.1) bezeichnet und die Verbindung zwischen den *fixes* als *steps* (Mark, 1998).

Spaccapietra *et al.* (2008) sehen eine Trajektorie eher als anwendungsspezifische Grösse, wobei eine Bewegung nicht unbedingt immer ein Teil davon ist, da sie nicht relevant für den behandelten Bereich ist. Man beachte in Abbildung 3.2, dass Teile des Weges (*path*) nicht als Trajektorien behandelt werden.

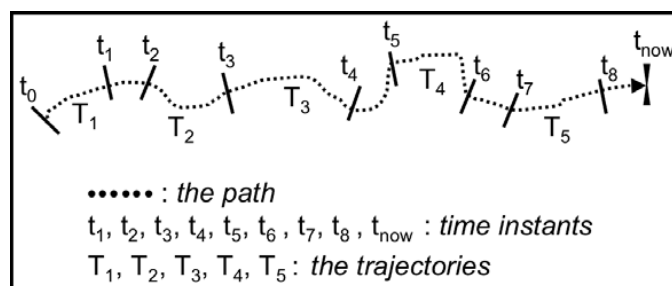


Abbildung 3.2: Weg (*path*) eines Objektes und *trajectories*, definiert mit einer semantischen Zerlegung des Weges, z.B.  $t_0$  als Start- und  $t_1$  als Endzeitpunkt von Trajektorie  $T_1$ , aus Spaccapietra *et al.* (2008).

Die zeitliche Granularität, der zeitliche Abstand zwischen den einzelnen *fixes*, entscheidet unter anderem wie genau die wirkliche Bewegung durch die Daten wiedergegeben wird. Weiter beeinflusst die Art der Bewegung die Repräsentativität der gesammelten Daten und gerade in der (Bewegungs-) Ökologie entscheidet die Spezies des zu untersuchenden



Tieres über die Form des Bewegungspfades. So bewegen sich migrierende Huftiere eher kontinuierlich (*continuous path*), während sich bei Bienen Bewegungen und Stopps immer abwechseln und die Stillstände ähnlich wichtig sein können (*discontinuous path*) (Dodge *et al.*, 2008).

Um *undersampling* (zu grobe zeitliche Granularität) oder *oversampling* (zu viele *fixes*) zu vermeiden, muss je nach Bewegungsart des Objekts bei der Datenerfassung die Abtaststrategie angepasst werden. Bei Ersterem besteht die Gefahr, dass wichtige Informationen verloren gehen, bei Letzterem können durch Nachbearbeitung unerwünschte Effekte, wie z.B. *noise*, immer noch entfernt werden (Laube *et al.*, 2005).

Grundlegend besitzen alle raum-zeitliche Daten (und somit auch Bewegungsdaten) drei Komponenten: Raum (wo), Zeit (wann) und Objekt (was) (Peuquet, 1994). Daraus lassen sich drei Arten von Anwendungen schliessen: 1) Beschreibung des Objektes (oder Set von Objekten) zu festgelegter Zeit und Örtlichkeit, 2) Beschreibung des Ortes (oder Set von Orten) wo sich festgelegtes Objekt zu bestimmter Zeit befindet und 3) Beschreibung des Zeitpunktes (oder Set von Zeitpunkten), wann bestimmtes Objekt an festgelegtem Ort ist. Calenge *et al.* (2009) unterscheiden bei Bewegungsdaten zusätzlich zwischen einer Abfolge von Positionen ohne absolute Zeitangaben (Typ I) und den Standorten mit einem Zeitstempel der Aufnahme (Typ II). Erstere Daten können z.B. durch indirekte Beobachtungen (siehe nächstes Kapitel) erfasst werden, Typ II Bewegungsdaten entstehen typischerweise aus Messungen mit Fernmesstechniken wie GPS. Werden die Daten in regelmässigen Abständen gemessen, bezeichnet man sie als regulär, bei unregelmässigen Messintervallen als irregulär. Generell sind die Bewegungsdaten aus GPS-Messungen also regulär und vom Typ II (siehe nächstes Kapitel).

### 3.3 Erfassungstechniken für Bewegungsdaten

Für die Erfassung der Bewegungen gibt es unterschiedliche Ansätze, da die Trajektorien durch begrenzte Sequenzen von zeitlich referenzierten Positionen wiedergegeben werden müssen. Andrienko *et al.* (2008) zählen die verschiedenen Wege auf, um Bewegungen zu beobachten und zu speichern:

- 1) Zeit-basierte Aufnahme: Positionen der Entitäten werden in regelmässigen Zeitabständen gemessen.
- 2) Änderungs-basierte Aufnahme: Eine Aufzeichnung wird gemacht, sobald die Position der Entität nicht mehr der letzten entspricht.
- 3) Orts-basierte Aufnahme: Sobald eine Entität nahe einer spezifischen Position ist, wird z.B. durch einen Sensor eine Aufzeichnung erstellt.
- 4) Ereignis-basierte Aufnahme: Sobald ein bestimmtes Ereignis auftritt, wird die Position der Entität gespeichert.

Natürlich kann auch eine Kombination der oben genannten Methoden verwendet werden, damit die gesammelten Positionsdaten möglichst der Anwendung entsprechen. In der Bewegungsökologie werden verschiedene Methoden angewendet, um Daten zu den Tieren in Bewegung zu sammeln. Einerseits werden direkte Beobachtungen im Feld gemacht, wo die Forscher den Tieren folgen und die Trajektorien anhand von Merkmalen in der Landschaft einzeichnen. Andererseits wird auch anhand indirekter Beobachtungen, wie z.B. anhand von Spuren im Sand oder Schnee ein Bewegungspfad erstellt.

Die technischen Möglichkeiten zur Verfolgung und zum Studium von Tieren aus der Ferne haben sich in den letzten 35 Jahren kontinuierlich verbessert und so erstaunt es nicht, dass die Verwendung dieser Fernmesstechniken auch in der Bewegungsökologie zugenommen hat (Tomkiewicz *et al.*, 2010). *Very High Frequency* Triangulation (VHF) wird mit dem Aufkommen vom satellitengestützten Positionierungssystems *Global Positioning System* (GPS) zunehmend abgelöst. Eine höhere Präzision der gemachten Messungen, ununterbrochene Einsatzfähigkeit mit möglichen Positionsupdates alle Sekunden – auch bei schlechtem Wetter – sind Vorteile von GPS, und gekoppelt mit Datenübermittlungstechnologien ist diese Methode zudem mit geringerem Aufwand verbunden. Die Aktualisierung des Standortes wird in der Praxis sehr selten mit der höchstmöglichen Häufigkeit durchgeführt, da die Batterielebensdauer der heutigen Geräte mit Anzahl GPS-*fixes* und anschliessender Übermittlung limitiert ist. Mit langlebigeren Batterien, verbrauchsarmen GPS-Empfängern und grösserer internen Speicherkapazität wird es in naher Zukunft möglich sein, die Tiere jederzeit zu lokalisieren. Durch die Verwendung von Trägheitsnavigationssystemen (*Inertial Navigation System* INS) als Ergänzung zu GPS könnte man dieses Energieproblem umgehen. Für die Erstellung eines solchen System ist ein Schrittzähler, ein elektronischer Kompass sowie ein dreidimensionaler Beschleunigungsmesser notwendig. Der mit gereister Distanz immer grösser werdende Fehler kann mit gut getimten GPS-*fixes* klein gehalten werden. In Tomkiewicz *et al.* (2010) befindet sich eine gute Übersicht zu den aktuellen Positionierungstechniken aus der Ferne.

Zusätzlich zu der höheren Genauigkeit durch die Verwendung von Positionierungstechniken wurde auch der Standpunkt der Forschung ein wenig geändert. Die Perspektive wurde vom Forscher und seinen festgelegten Hypothesen hin zu einer eher Tier-basierten Perspektive, den unvoreingenommenen Bewegungsdaten, verschoben (Hebblewhite und Haydon, 2010). Diese Entwicklung kann durchaus als Fortschritt gesehen werden, doch mit dem zunehmenden Einsatz von diversen technischen Hilfsmitteln wird die Forschung nicht automatisch einfacher. Oftmals sind diese räumlichen Daten in separaten *flat files* gespeichert und können nur durch einzelne Nutzer aufgerufen werden. Dies erfordert das Kopieren und Verändern von gesammelten Daten, sowie Export- und Importprozedere in lokal installierte Programme, was viel Zeit verbraucht und zudem eine Fehlerquelle ist. Aktuell stellt die Verwaltung und Analyse dieser grossen, kontinuierlichen Datensätze über Bewegungen zahlreicher Individuen, versehen mit Informationen über deren Verhalten und Umgebung, eine immer grössere Herausforderung dar.

### 3.4 Anforderungen an ein Managementsystem für Wildtierdaten

Aufgrund dieser Problematik mit dem Datenhandling präsentieren Urbano *et al.* (2010) die grundsätzlichen Anforderungen, wie solche Wildtierdaten gespeichert und verwaltet werden sollen:

- 1) **Datenskalierbarkeit:** Mit abnehmendem Kosten für die Technologie werden pro Tier pro Tag anstatt ein paar Dutzend Positionen wohl bald einige hundert oder sogar tausend Lokalisierungen vorgenommen werden. Zudem werden immer mehr Tiere mit GPS-Empfängern ausgerüstet, was zu komplizierterer Datenstruktur und sehr grossen Datenmengen führt. Um Konsistenz der Daten zu gewährleisten, sind Persistenz und Kapazität der Speicherung wichtig.

- 2) Langzeitspeicherung für Wiederverwendung der Daten: Die Daten sollen konsistent und unabhängig der Anwendungen auf lange Zeit gespeichert werden, um auch für andere Studien zugänglich zu sein. Auch Cagnacci *et al.* (2010) wünschen sich, dass die Daten der Wissenschaft erhalten bleiben und gerade für ökologische Fragen über grosse Zeitabschnitte sowie ausgedehnte räumliche Dimensionen wiederverwendet werden sollen. Bei komplexen Phänomenen mit vielen Faktoren, wie z.B. dem Einfluss des Klimawandels auf das Verhalten oder die Verteilung von Tieren, könnten solche Querverbindungen von unschätzbarem Wert sein.
- 3) Periodische und automatische Datenerfassung: Da der Zugriff auf die Daten fast in *real time* erwartet wird und die Updates der Positionsdaten in immer höheren Frequenzen abgefragt werden, müssen gewisse Abläufe wie beispielsweise Empfang, Prüfung und Speicherung der Daten automatisch ablaufen können.
- 4) Effizientes Abfragen der Daten: Für effizientes Management und Analyse der Daten sind schnelle Suche-, Abfrage- und Manipulationstools unumgänglich.
- 5) Verwaltung räumlicher Informationen: GPS-Daten sind grundsätzlich räumliche Daten und benötigen deshalb spezifisch auf den räumlichen Bereich zugeschnittene Werkzeuge.
- 6) Globale räumliche und zeitliche Referenzen: Gerade für Studien mit regionalen oder sogar globalen Perspektiven sollte das Arbeiten mit globalen räumlichen und zeitlichen Referenzen (WGS84<sup>1</sup> vs. Landspezifisches Koordinatensystem, Coordinated Universal Time vs. lokale Zeit) in der Datenstruktur integriert werden.
- 7) Heterogenität der Applikationen: Um die vielfältigen Anwendungen von GPS-Daten in verschiedenen Bereichen der Ökologie zu ermöglichen, sollte die Softwarestruktur die Integration verschiedener Tools zulassen.
- 8) Einfache Implementierung von Algorithmen: Die Analyse von räumlichen Daten wild lebender Tiere hat sich in den letzten Jahren stark verändert und wird sich wohl auch in Zukunft weiter entwickeln. In der Softwareumgebung sollte deshalb die Implementierung und spezifische Anpassung von Algorithmen einfach möglich sein.
- 9) Integration von verschiedenen Datenquellen: Daten aus der Fernerkundung, von Umwelt- oder anderen Datenbanken wie auch Beobachtungsdaten oder Messwerte aus der Gefangennahme von Tieren sollen korrekt und effizient in die Datenstruktur integriert werden können.
- 10) Multi-User Unterstützung: Innerhalb von Forschungsgruppen, öffentlichen Institutionen und Umweltorganisationen sollten viele Nutzer gleichzeitig die Daten abrufen können. Mit lokalem oder auch mit Fernzugang sollen die verschiedenen Zugangsrechte vergeben und mit den damit implizierten Kontroll- und Sicherheitsmechanismen versehen werden.
- 11) Gemeinsame Datennutzung: Gerade im Kontext der interdisziplinären Forschung bei ökologischen Fragen ist der Datenaustausch untereinander von hoher Priorität. Damit verbunden ist auch die Festlegung von Standards im Bereich Datenformat, Metadaten und Methoden zur Datenspeicherung um die Interoperabilität zu gewährleisten.

---

<sup>1</sup>World Geodetic System 1984: geodätisches Referenzsystem

- 12) Verbreitung der Daten: Es ist wichtig, die Daten innerhalb der wissenschaftlichen Gemeinschaft aber auch in der Öffentlichkeit zu präsentieren, um etwa administrative Prozesse zu beschleunigen, Geldmittel zu beschaffen, Leute auf die Wichtigkeit von Projekten (und der Umwelt allgemein) hinzuweisen und womöglich gar selbst zur Informationsquelle zu machen. In diesem Sinne ist die Integration eines spezifischen Zugangstools nötig, zum Beispiel in Form eines Web-GIS.
- 13) Kosteneffektivität: Gerade für Institutionen mit limitierten Mitteln ist die Kosteneffektivität von den zu beschaffenden Softwaretools ein sehr wichtiger Faktor, da im Datenmanagement gesparte Gelder beispielsweise in die Produktion der Daten (mehr GPS-Geräte, mehr Personal, usw.) investiert werden können. Software beruhend auf Open Source Lösungen sind deshalb besonders zu beachten (siehe auch Kapitel 10.3).

### 3.5 Speicherung von Bewegungsdaten

Will man raum-zeitliche (oder Bewegungs-) Daten speichern, so stellt sich die Frage, wie man dies am effektivsten machen sollte. Es gibt diverse Konzepte und Modelle für die Speicherung, Abfrage und Analyse von raum-zeitlichen Daten. Grundsätzlich lassen sich diese in drei raum-zeitliche Bezugssysteme nach Worboys (2005) einordnen: *temporal snapshots*, *object change* und *events, actions & processes*.

*Stage Zero* repräsentiert dabei den jetzigen Zustand der meisten vorhandenen Softwarelösungen wo meist nur ein einzelner Stand eines Objektes gespeichert werden kann. Obwohl theoretisch die Vergangenheit oder die Zukunft repräsentiert werden könnten, ist nur das Speichern eines einzelnen Moments möglich, und es wird gewöhnlich der Stand mit den möglichst aktuellen Daten gewählt.



Abbildung 3.3: Drei *temporal snapshots* von Maine 1902, 1946 und 1955 nach Worboys (2005).

In *Stage One* werden zeitliche Momentaufnahmen (*snapshots*) erstellt und gespeichert und ergeben durch ihre Sequenz eine Art zeitliche Abfolge. In Abbildung 3.3 ist klar ersichtlich, dass Änderungen zwischen den einzelnen Aufnahmen passiert sind. Es wird jedoch nicht klar, wann genau diese passiert sind und unterstreicht damit die Wichtigkeit von *valid* und *transaction time* (Kapitel 3.5.3). Dieser Ansatz ist in aktuellen raum-zeitliche Datenbankmodellen der am weitesten verbreitete, was sich in Konzepten wie Zeitstempel,

zeitliche Granularität oder zeitliche Indizierung zeigt. Die Wahl der zeitlichen Struktur ist abhängig von der Anwendung und sollte gut gewählt werden (z.B. kontinuierliche vs. diskrete Änderungen). Das Modell kann sehr speicherintensiv werden, und dies ohne dass allenfalls eine Veränderung zwischen zwei Zeitständen erfolgt ist.

In *Stage Two* bewegt sich der Fokus von einer zeitlichen Sequenz von Objekten, ihren Attributen und Beziehungen zu den Änderungen welche bei Objekten, Attributen und Beziehungen eintreffen können (*object changes*). Die möglichen Veränderungen nach Hornsby und Egenhofer (2000) beziehen sich entweder nur auf das Objekt selbst (*creation, continuation, disappearance, reappearance, transformation, death*), auf zwei Objekte (*transmission*) oder aus einem Objekt werden zwei (*cloning*). Diese Änderungen sind jedoch nicht komplett, so müsste etwa eine Bewegung auch miteinbezogen werden. Ein Modell der Welt mit Objekten, welche sich mit der Zeit entwickeln, ihre Identität jedoch beibehalten, klingt natürlich. Doch es tauchen Probleme auf, welche in Heller (1990) vertieft behandelt werden. Wie auch in *Stage One* werden hier geographische Entitäten mit ihren Attributen und Beziehungen nach der in Grenon und Smith (2004) erläuterten SNAP-Ontologie dargestellt, also Sequenzen von verschiedenen Objektkonfigurationen.

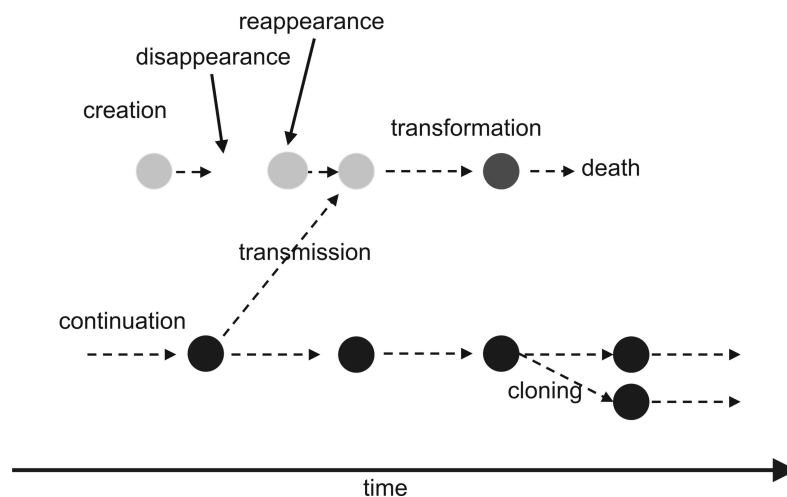


Abbildung 3.4: Mögliche Änderungen von raum-zeitlichen Daten im zweiten Modell *object change* nach Worboys (2005).

Im Gegensatz dazu behandelt *Stage Three* Veränderungen mit Ereignissen, Aktionen und Prozessen (*events, actions, processes*) nach der SPAN-Ontologie (Grenon und Smith, 2004). Es muss unterschieden werden zwischen *continuants*, also Objekten, welche über die Zeit fortbestehen und Ereignissen (*occurents*), welche eintreten und dann wieder verschwinden. Solche Ereignisse können Auswirkungen auf diverse Objekte haben und mit verschiedenen anderen in Beziehung stehen. Eine Schwierigkeit besteht in der Definition der verschiedenen Begriffe: Was ist nun ein Ereignis, was eine Aktion? Ist eine Lawine nun eine Aktion oder ein Ereignis? Wahrscheinlich ist noch entscheidend ob diese nun natürlich oder durch einen für die Anwendung interessanten 'Agenten' ausgelöst wurde. Weiter besteht die Frage, welche Prozesse nun als solche gelten und welche man ignorieren soll. Trotz der Komplexität mit Begriffen und Definitionen, mit einem komplett ausgearbeiteten ereignisorientierten Modell würde der Fokus von Abfragen zu Zuständen mehr in Richtung

der Abfragen zu Erklärungen verschoben werden. Das Zusammenspiel von Objekten mit Events und Beziehungen von Events untereinander könnte neue Einblicke gewähren.

Im nächsten Kapitel werden die Grundbegriffe von Datenbanken sowie Vorteile gegenüber einem *file*-basierten Ansatz kurz vorgestellt, um dann den Schwerpunkt auf Datenbanken zu legen, mit welchen ein Handling von raum-zeitlichen sowie Bewegungsdaten möglich ist.

### 3.5.1 Datenbanken

Eine Datenbank (DB) stellt Aspekte der realen Welt dar und ist eine logisch zusammenhängende Sammlung von Daten mit einer bestimmten inhärenten Bedeutung. Zufällig angehäuften Daten werden nicht als Datenbank angesehen, denn eine DB wird für bestimmte Zwecke und Benutzer innerhalb einer oder mehrerer Organisationen entworfen, entwickelt und mit Daten gefüllt (Elmasri und Navathe, 2009; Teorey *et al.*, 2011).

Ein Datenbankmanagementsystem (DBMS) ist ein generelles Softwaresystem für die Erstellung und Unterhaltung von Datenbanken. Dazu gehört die Definition (Spezifikation der Datentypen, Strukturen und Einschränkungen der Daten), die Konstruktion (Speichern von Daten, welche vom DBMS kontrolliert werden) sowie die Manipulation (Abfragen und Änderungen der DB).

Als Datenbanksystem (DBS) wird die Kombination von Datenbank und Datenbankmanagementsystem bezeichnet.

Die Vorteile eines Datenbanksystems gegenüber einem traditionellen, *file*-basierten Ansatz sind Redundanzkontrolle, Zugriffsbeschränkungen, sowie das Vorhandensein von strukturierten und beschriebenen Daten, Skalierbarkeit und Erweiterbarkeit, Mehrfachnutzung, Flexibilität, die Trennung von Daten und Anwendung, Datenpersistenz und -sicherung bzw. Recovery. Zudem können Integritätsbedingungen der Daten und Beziehungen zwischen den Datenelementen festgelegt werden. Zur besseren Bedienungsfreundlichkeit, der Konsolidierung aller Informationen, Verfügbarkeit und schnelleren Zugriffs kommt ausserdem die Möglichkeit eines Web-basierten globalen Zugriffs. Gerade bei zunehmender Komplexität von Daten und der Nutzung durch mehrere Personen offenbart ein DBMS seine Vorteile gegenüber einem *file*-basierten Ansatz. Auch kontinuierlich zunehmende und sich ändernde Daten können in einem DBMS viel einfacher verwaltet werden (Elmasri und Navathe, 2009).

Nachteile können neben den eher hohen Anschaffungskosten für Soft- und Hardware das gegebenenfalls nötige, qualifizierte Personal oder die (zu) hohe Komplexität sein.

In den nächsten Kapiteln wird auf die Begriffe für die Speicherung und Analyse von räumlichen sowie zeitlichen Informationen in Datenbanken eingegangen, im Hinblick auf eine genauere Fokussierung in Richtung raum-zeitliche Datenbanken sowie *Moving Objects Databases*.

### 3.5.2 Räumliche Datenbanken

Ziel der räumlichen Datenbankforschung war die Erweiterung von gängigen DBMS Datenmodellen und Querysprachen, um Geometrien naturgemäss zu speichern und abzufragen. Dabei müssen Datenstrukturen für geometrische Formen, Algorithmen für geometrische Berechnungen, Queries, Indizierungstechniken für multidimensionalen Raum sowie Erweiterungen für den Optimizer geschaffen werden. Dabei war die Unterstützung für Geogra-

phische Informationssysteme (GIS) die treibende Kraft, es gibt aber auch noch andere Anwendungsgebiete wie z.B. die Medizin (3D Modell des menschlichen Körpers, Proteinstrukturen). Wichtig ist die Unterscheidung von Bilddatenbanken und räumlichen Datenbanken. Obwohl geographischer Raum durch Bilder repräsentiert werden kann, wird der Schwerpunkt auf Objekte mit einer klar definierten Ausmass und Position im Raum gelegt (Güting und Schneider, 2005).

Disjoint	Meet	Overlap	Equal
Covers	CoveredBy	Inside	Contains

Abbildung 3.5: Mögliche topologische Beziehungen nach dem *9-Intersection Model* aus Pelekis *et al.* (2004).

Mit der Erstellung von räumlichen Datentypen (*spatial data types* SDTs) wird die räumliche Komponente in die Datenstruktur eingekapselt (*encapsulated*) und ermöglicht damit z.B. Abfragen zu topologischen Beziehungen nach Egenhofer und Herring (1994) (wie in Abbildung 3.5 dargestellt) oder Berechnungen von Geometrie wie Fläche oder Länge. Verschiedene Raummodelle können benutzt werden, dazu gehören *set-based*, *topological*, *network*, *Euclidian* und *metric space* (Shekhar *et al.*, 1999). In relationalen und objekt-relationalen Datenbanken wird das *set-based* Raummodell angewendet. Wenn räumliche Datentypen definiert und festgelegt sind, so können sie in einem DBMS als Attributtypen eingebunden werden. Neben den üblichen Datentypen wie *string*, *int*, usw. können zusätzlich die räumlichen Datentypen *point*, *line*, *polygon* verwendet werden und ermöglichen die Verwendung von räumlichen Indizes (Güting und Schneider, 2005).

Beispiel aus Güting und Schneider (2005) wie mit einem möglichen räumlichen Operator (ADJACENT) die Anzahl Nachbarstaaten aufgelistet werden kann:

```
SELECT s.name, COUNT(*)
FROM states s, states t
WHERE s.area ADJACENT t.area
GROUP BY s.name
```

SQL 3.1: Beispiel für eine Abfrage einer räumlichen Datenbank.

Für Bewegungsdaten sind räumliche Datenbanken nicht optimal geeignet, da sie meist statisch sind und bei Änderungen der letzte Stand verloren geht. Die Eigenschaften einer räumlichen Datenbank sind jedoch die Grundvoraussetzung, um bewegliche Objekte zu modellieren.

### 3.5.3 Zeitliche Datenbanken

In normalen DBMS wird die Zeit durch explizite Zeitattribute vermerkt und zeitliche Datentypen wie *date* und *time* sind nur in begrenztem Mass unterstützt. So wird ein *join* zweier Tabellen über ein Zeitattribut in Standard DBMS nicht effizient und eventuell auch nicht fehlerfrei gelöst. Wird jedoch 'wahre' zeitliche Unterstützung eingebaut, kann dies automatisch und ohne Vorbedingungen in der Query effizient gelöst werden. Das Ziel der zeitlichen Datenbankforschung ist die Erweiterung des Systems sowie die Integration von zeitlichen Konzepten tief im DBMS Modell und der Abfragesprache, um effiziente Anwendungen zu erzielen (Güting und Schneider, 2005).

Generell wird Zeit als eindimensionaler Raum angesehen, welcher sich von der Vergangenheit in die Zukunft ausdehnt – begrenzt oder unendlich. Zeit kann diskret, dicht (*dense*) oder kontinuierlich modelliert werden. Obwohl Menschen Zeit als kontinuierlich wahrnehmen, wird sie in zeitlichen Datenbanken aus praktischen Gründen diskret modelliert (Güting und Schneider, 2005). Aus abstrakter Sicht sind verschiedene Auffassungen von Zeit möglich: linear (wie in den meisten Datenbanken), zyklisch, verästelt oder alternative Zeitachsen (Pelekis *et al.*, 2004).

Weiter können mit zeitlichen Datenbanken Abfragen zu zeitlicher Topologie gestellt werden, um z.B. nur Ereignisse während einer gewissen Periode auszuwählen. Allen (1983) schlug die 13 möglichen zeitlichen Beziehungen vor, ähnlich dem 9-Intersection Modell aus der räumlichen Datenbankforschung. So kann 'ein Zeitpunkt T *innerhalb* einer Zeitperiode A sein, welche wiederum die Periode B *trifft*. In zeitlichen Datenbanken gibt es analog zu den in 3.5.2 angesprochenen räumlichen Datentypen *point*, *line*, *region* spezielle zeitliche Datentypen wie z.B. *instant*, *period*, *interval* (siehe Güting und Schneider (2005)). Mit passenden zeitlichen Operatoren wie *INSIDE* oder *MEETS* können oben erwähnte Abfragen durchgeführt werden.

Es können mehrere Zeitarten gleichzeitig gespeichert werden (*multiple times*): *Valid time* markiert den Zeitpunkt wann die Information in der realen (Anwendungs-) Welt gültig war, ist oder sein wird. Diese kann beispielsweise als Weltzeit (z.B. UTC, *Coordinated Universal Time*), als lokale Zeit und/oder als vom Nutzer definierte Zeit vermerkt werden. Die Transaktionszeit (*transaction time*) markiert den Zeitpunkt, wann die Information in der Datenbank gespeichert wurde. Eine Datenbank wird als bitemporal bezeichnet, wenn sowohl *valid* als auch *transaction time* registriert werden.

Beispiel einer Query, wie sie in einer zeitlichen Datenbank z.B. mit der Abfragesprache TSQL2 (*temporal SQL*) erstellt und angewendet werden könnte: Mit einem zeitlichen Operator (*INSIDE*) werden alle Konzerte während den Osterferien ausgewählt.

```
SELECT c.name, c.validtime
FROM holidays h, concerts c
WHERE c.validtime INSIDE h.validtime
AND h.name = 'easter'
```

SQL 3.2: Beispiel für eine Query in einer zeitlichen Datenbank.



### 3.5.4 Raum-Zeitliche Datenbanken

In raum-zeitlichen Datenbanken wird die Speicherung und Analyse von Anwendungen behandelt, welche sich durch räumliche sowie zeitliche Eigenschaften auszeichnen. Ziel ist die Abbildung von ausgesuchten, dynamischen Bereichen der realen Welt, wo das Verhalten von Objekten in Raum und Zeit aufgezeichnet wird. Als mögliche Einsatzgebiete zählen Geographische Informationssysteme, verschiedene Bereiche der Raumplanung und automatische Kartendienste. Um Daten zu speichern, welche sowohl eine räumliche als auch eine zeitliche Komponente besitzen, braucht es ein geeignetes Datenmodell. Naheliegender ist eine simple Kombination der Datentypen und Abfragesprachen aus der räumlichen und der zeitlichen Datenbankforschung, wobei jedoch nicht das volle Potential abgerufen wird. Raum- und Zeitinformation eines Objekts würden zwar mit spezifischen Datentypen abgebildet, wären aber trotzdem noch als separate Attribute gespeichert. Frank (2003) untersuchte, in wieweit der raum-zeitliche Aspekt implizit in den Daten integriert werden sollte und präsentierte eine Ontologie für die raum-zeitliche Datenbankmodellierung. Grundsätzlich kann ein Objekt im Raum seine Geometrie, Topologie sowie seine semantischen Attribute mit dem Laufe der Zeit ändern. Daraus ergeben sich acht mögliche Arten von Veränderung, inklusive 'keine Änderung' (Abbildung 3.6). Die Speicherung der Änderungen von Attributen, Beziehungen und Geometrie der raum-zeitlichen Daten kann unterschiedlich gelöst werden (siehe *Three Stages* nach Worboys (2005) auf Seite 12), Pelekis *et al.* (2004) präsentieren verschiedene Modelle für eine Eingliederung der Raum- und Zeitinformation in ein Datenmodell.

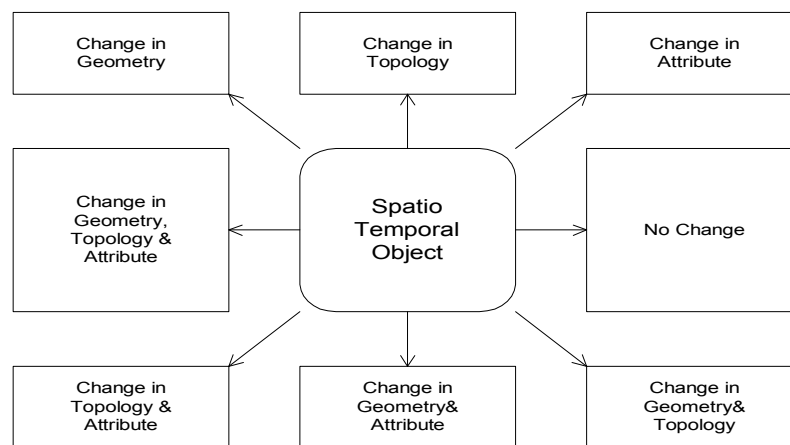


Abbildung 3.6: Mögliche Veränderungen von raum-zeitlichen Objekten, aus Pelekis *et al.* (2004).

Im *Snapshot Model* beschreibt jeder *Snapshot* die Situation zu einem bestimmten Zeitpunkt, somit weiss man wo welche Objekte in welchem Zustand waren. Zeit wird als Attribut der Position angesehen und nur die Valid Time wird unterstützt. Dies ist das einfachste Modell um raum-zeitliche Informationen zu speichern und ist nur geeignet, um sehr einfache Abfragen zu behandeln. Das Modell ist zugleich auch sehr limitiert, da für die Beantwortung komplexer Queries kostspielige Algorithmen notwendig sind. Nur schon die Detektion der Änderungen zwischen zwei Zeitpunkten benötigt ausgiebige Vergleiche der verschiedenen Snapshots und die zu den Unterschieden führenden Prozesse werden nicht einmal erklärt. Zudem ist die Menge an redundanten Information in diesem Modell sehr hoch. Regeln für die interne Logik oder Integrität sind schwierig einzuführen, da im

Bereich der zeitlichen Struktur keine Einschränkungen herrschen (Langran, 1992).

Ein ähnlicher Ansatz nach Armenakis (1992) beinhaltet die Unterteilung in die drei Ansätze *static mode*, *differential mode* und *sequential updating mode* mit unterschiedlichen Speicherungs-, Abfrage- und Aktualisierungseffizienzen. 1) Der statische Modus entspricht dabei dem oben besprochenen. Auch bei nur minimalen Änderungen an einem einzigen Objekt wird die gesamte Datenmenge für einen neuen Zeitpunkt erfasst und gespeichert. Dies kann zu enormen Datenmengen und Redundanzen führen, und Veränderungen zwischen verschiedenen Zeitständen müssen aufwändig berechnet werden. 2) Beim differentiellen Modus wird nur der anfängliche Zustand gespeichert und Aktualisierungen regelmässig mit sogenannten *delta files* festgehalten, wobei zwischen Änderungen vom Anfangszustand und der vorherigen Lage unterschieden wird. Alternativ kann der aktuellste Stand komplett und Änderungen zurück als *delta files* gespeichert werden. 3) Der dritte Ansatz unterscheidet sich vom zweiten, indem nicht regelmässig Veränderungen erfasst werden, sondern nur falls welche auftreten.

Beim *Space Time Composite Data Model* (STC) werden alle Linien in Raum und Zeit auf die räumliche Ebene projiziert und miteinander verschnitten, wobei jedes dabei entstehende Polygon seine eigene Attributgeschichte hat. Das Modell unterstützt nur Vektorgeometrien mit diskreten Veränderungen, Bewegung wird nicht unterstützt. Bei einem Update der Datenbank müssen alle Objekte neu rekonstruiert werden, und wichtige Abfragen zu Beziehungen und Verhalten von Objekten werden schlecht unterstützt. Das Modell kann am ehesten dem *Stage One* nach Worboys (2005) zugeordnet werden.

Jedes Objekt wird im *Simple Time Stamping* Modell mit einem Paar von Zeitstempeln versetzt, einen für den Zeitpunkt der Erstellung und einen für die Beendigung der Gültigkeit. Das Modell ist relativ simpel, mit einfachen Queries können verschiedene Zeitpunkte der Vergangenheit abgefragt werden, jedoch nur *valid time*-Zustände. Es wird jeweils der aktuellste Stand eines Objektes bereit gehalten und über Referenzen kann auf die älteren Versionen zugegriffen werden.

Ähnlich dem *Stage Three* nach Worboys (2005) bestehen einige *Event-Oriented Models*, welche jedoch vor allem den Namen gemeinsam haben. TEMPEST, als temporaler GIS-Prototyp, wird als Repräsentant dieser Kategorie gesehen. Damit werden Änderungen des Grundzustandes einer Basiskarte in einer zeitlichen Abfolge gespeichert, wobei eine abrupte Änderung oder das Überschreiten eines Grenzwertes bei einem kontinuierlichen Prozess als *event* gilt. Mit diesem Ereignis kann eine Liste von Veränderungen verbunden werden und entweder nur der Unterschied zur vorherigen Situation oder ein ganz neuer Zustand registriert werden. Mit dem *Event Oriented Spatio-Temporal Data Model* (ESTDM) präsentieren Peuquet und Duan (1995) ein raster-basiertes, ereignis-orientiertes Modell.

Es existieren viele weitere konzeptuelle Modelle, wie das *Three-Domain Model* nach Yuan (1997), wo die drei 'Domänen Raum, Zeit und Semantik separat gespeichert werden, oder das *History Graph Model*, welches die Handhabung von Objekten wie Events erlaubt und entweder als *object versions* oder *transitions* registriert werden. Tryfona und Jensen (2000) legen mit STER (*spatio-temporal entity relationship model*) eine Erweiterung des ERs vor, um die Geometrie und Zeitinformation von raum-zeitlichen Anwendungen semantisch darzustellen. Pelekis *et al.* (2004) stellen noch verschiedene *Object-Relationship Models* sowie *Spatio-Temporal Object-Oriented Data Models* vor. Die wenigsten dieser konzeptuellen Modelle wurden implementiert und getestet, womit sich deren Stärken und Schwächen schlecht abschätzen lassen.

Nachfolgend ist ein Beispiel für eine raum-zeitlichen Abfrage aus Güting und Schneider (2005) mit einem räumlichen und zeitlichen Operator (ADJACENT bzw. OVERLAPS). Es sol-

len alle Grundstücke ausgewählt werden, welche am 17. März 1977 mit Charles Smith benachbart waren. Man beachte, dass dies eine bitemporale Datenbank ist und der SQL die *valid time* anspricht.

```
SELECT r2.area, r.owner
FROM real_estate r1, real_estate r2
WHERE r1.owner = 'Charles Smith
AND VALID(r1) OVERLAPS DATE '[1977-03-17]' AND r1.area ADJACENT r2.area
```

SQL 3.3: Beispiel für eine Query in einer raum-zeitlichen Datenbank.

Dies ist jedoch 'nur' die Kombination von räumlichen und zeitlichen Datenbanken und benötigt räumliche und zeitliche Datentypen plus eine geeignete Abfragesprache. Traditionell fokussierten raum-zeitliche Datenbanken meist auf diskrete (menschliche) Änderungen wie z.B. der Wechsel von Grundbesitzern oder Änderungen an Grundstücken. Für viele Anwendungen genügt eine Kombination der Möglichkeiten aus räumlicher sowie zeitlicher Datenbankforschung um die gewünschten Abfragen zu beantworten. Es gibt jedoch auch unzählige Dinge aus der realen Welt, welche sich nicht mit diskreten Schritten verändern, sondern sich kontinuierlich bewegen. Betrachtet man eine Auflistung der zehn Klassen an raum-zeitlichen Daten nach Güting und Schneider (2005), so reicht dieser grundsätzliche Raum-Zeit Ansatz bei ziemlich vielen Anwendungen:

- 1) Events in Zeit und Raum - (*point, instant*). Beispiele sind archäologische Entdeckungen, Flugzeugabstürze, Vulkanausbrüche, Erdbeben (in grossem zeitlichen Massstab, wo Dauer nicht relevant ist)
- 2) Orte gültig für bestimmte Zeitperioden - (*point, period*). Beispiele: Städte, welche innerhalb bestimmter Zeit gebaut wurden, Baustellen, Kohleminen oder Ölbohrstellen
- 3) Set von örtlichen Events - Sequenz von (*point, instant*). Gesamthaft gesehene Einheiten von 1) wie z.B. alle Vulkaneruptionen aus dem letzten Jahr
- 4) Stufenweise konstante Orte - Sequenz von (*point, period*). Beispiele sind Hauptstädte von Ländern, Hauptsitz eines Unternehmens, die Unterkünfte eines Reisenden
- 5) Bewegliche Einheiten - *moving point*. Z.B. Menschen, Flugzeuge, Autos, Satelliten, Tiere
- 6) Regionsevents in Zeit und Raum - (*region, instant*). Zum Beispiel ein Waldbrand in grossem Massstab gesehen
- 7) Regionen gültig für bestimmte Zeitperiode - Sequenz von (*region, period*). Beispiel: Eine zeitlich begrenzt gesperrte Gegend wegen akuter Lawinengefahr
- 8) Set von Regionenevents - Sequenz von (*region, instant*). Beispiele sind die Olympischen Sommerspiele alle vier Jahre in grossem Massstab gesehen
- 9) Stufenweise konstante Regionen - Sequenz von (*region, period*). Beispiele: Länder, Grundstückbesitz, variierende landwirtschaftliche Bodennutzung
- 10) Bewegliche Einheiten mit einem Ausmass - *moving region*. Beispiele sind das Ausbreiten von Wüsten, Waldbrände in kleinem Massstab, Völker in der Geschichte der Menschheit

Nur Punkte 5) und 10) können mit rein räumlichen und zeitlichen Datentypen nicht modelliert und abgefragt werden, sondern müssen wegen kontinuierlicher Bewegung anders behandelt werden (siehe *moving objects databases* im nächsten Kapitel).

### 3.5.5 Moving Objects Databases

Betrachtet man ganz allgemein die Art von Veränderungen, welche Objekte in der realen Welt erfahren, so ist eine Unterscheidung zwischen diskreten und kontinuierlichen Änderungen auszumachen. Klassische Forschung zu raum-zeitlichen Datenbanken hatte den Schwerpunkt bei den diskreten, meist menschlich verursachten Veränderungen. Objekte, welche sich durch kontinuierliche Änderungen in Raum und Zeit auszeichnen, werden üblicherweise als *moving objects*, und Datenbanken, welche sich mit diesen befassen als *moving objects databases* definiert. Die Speicherung und Analyse von Objekten, die ihre Position oder ihre Form kontinuierlich mit der Zeit ändern können, stellt eine grosse Herausforderung dar und ist noch ein relativ junger Forschungszweig (seit ca. 1990) (Güting und Schneider, 2005; Pelekis *et al.*, 2004).

*Moving objects databases* können nicht nur die Trajektorie im Raum beinhalten, sondern auch zusätzliche Charakteristiken der Bewegung selbst wie z.B. Geschwindigkeit, Richtung oder Dynamik, aber auch Charakteristiken und Attribute der bewegten Objekte selbst. Es wird versucht, die Bewegungen der Objekte mit ihrer Umgebung, Phänomenen und Events in Beziehung zu setzen (Andrienko *et al.*, 2008).

In traditionellen Datenbankmanagementsystemen werden Daten als konstant angesehen, bis sie explizit verändert werden, was für die Modellierung von *moving objects* nicht ideal wäre. Aufgrund fehlender Konzepte präsentierten Sistla *et al.* (1997) mit dem *moving objects spatio-temporal data model* (MOST) ein Datenmodell speziell für sich bewegende Objekte. Denn die Speicherung der Position eines sich bewegenden Flugzeuges mit GPS-Empfänger stellt den Nutzer vor einen Zielkonflikt: Entweder werden dauernd Updates gemacht (Überlastung des Netzwerks mit ständigen *uploads/downloads*, der Computer und der Batterie) oder die Position ist nicht aktuell, was in der Flugsicherheit nicht ungefährlich wäre. Dieses Problem lösen die Autoren damit, dass die Position durch eine Funktion der Zeit (Position des Flugzeugs als Funktion eines Bewegungs-Vektors) repräsentiert wird, welche generell viel weniger aktualisiert werden muss. Mit dynamischen Attributen ermöglicht diese Art Speicherung von Bewegung auch Queries über die zukünftige Entwicklung von beweglichen Objekten, weshalb eine passende Abfragesprache entwickelt wurde (*future temporal logic* FTL). Angewendet wurde dieses Datenmodell später durch dieselben Autoren anhand von DOMINO (siehe Kapitel 3.7.1).

Güting und seine Gruppe entwickelten diese Idee weiter und stellen mit einer formalen Theorie über *moving points* sowie *moving regions* und einer voll funktionsfähigen Implementation (SECONDO, siehe Kapitel 3.7.1) einen der wichtigsten Beiträge zu den *Moving Object Databases* dar. Das Resultat ist ein reiches Set an abstrakten und diskreten Datentypen (u.a. mit dynamischen Attributen), deckende räumliche, zeitliche und bewegliche Typen als Erweiterung der raum-zeitlichen Datenbank (Güting und Schneider, 2005). Das oben genannte Beispiel mit dem Flugzeug könnte in etwa so aussehen:

```
flight (
flight_number: int,
airline: string,
pos: (x: dynamic number, y: dynamic number)
)
```

Je nach Abfragezeitpunkt wird die wiedergegebene Position der Datenbank dank der dynamischen Attribute angepasst. Trotzdem muss die Funktion (bzw. der Bewegungsvektor)

in der Datenbank von Zeit zu Zeit abgestimmt werden. Um eine Balance zwischen Update-Frequenz und Präzision zu erreichen, führen Güting und Schneider (2005) die sogenannte *dead-reckoning* Methode ein. Immer wenn die Position des sich bewegenden Objektes um einen festgelegten Grenzwert von der in der Datenbank gespeicherten Position (d.h. der Bewegungs-Funktion) abweicht, wird der aktuelle Bewegungsvektor übertragen.

Beispiel einer Query aus Güting und Schneider (2005) mit sich bewegenden Objekten. Es sollen alle Paare von Flugzeugen ausgewählt werden, welche sich irgendwann während ihrem Flug innerhalb von 500 Metern befanden.

```
SELECT f.id, g.id
FROM flight f, flight g
WHERE f.id <> g.id AND MIN(DISTANCE(f.route, g.route)) < 0.5
```

SQL 3.4: Beispiel für eine Query mit sich bewegenden Objekten.

Es gibt grundsätzlich wenige Datenbankprototypen für die Speicherung und Analyse von *moving objects*, welche diese Art Queries zulassen (siehe Kapitel 3.7.1). Eine Unterstützung für die Anreicherung von Trajektorien mit Zusatzinformationen fehlt jedoch bei diesen Prototypen, obwohl dies aus Anwendungssicht nötig wäre (Spaccapietra *et al.*, 2008). Aus diesem Grund präsentieren die Autoren ein konzeptuelles Modell für Trajektorien, um einerseits einfache Reisen von Startpunkt zu Ziel, und andererseits komplexe Bewegungen mit Unterbrüchen darstellen zu können. Die rohe Bewegungsinformation soll sinnvoll strukturiert und mit Attributen oder (raum-zeitlichen) Beziehungen semantisch angereichert werden können. Zugvögel beispielsweise machen auf ihrem Weg in den Süden diverse Stopps während der Dunkelheit für die Nahrungsaufnahme und ein wenig Schlaf. Durch eine anwendungsspezifische Sequenzierung der Trajektorien ist es möglich, der rohen Bewegungsinformation eine semantische Ebene beizufügen und damit eine sinnvolle Speicherung und Analyse der Daten zu ermöglichen. Im Gegensatz zu anderen MOD-Modellen ist die Trajektorie damit fester Bestandteil im Datenmodell und kann mit Attributen, semantischen und topologischen Bedingungen sowie Beziehungen zu anderen Objekten charakterisiert werden.

Spaccapietra *et al.* (2008) definieren eine Trajektorie als benutzerdefinierte Aufzeichnung einer Entwicklung der Position eines Objektes,  $t_{\text{begin}}$  als Start- und  $t_{\text{stop}}$  als Endzeitpunkt (siehe auch Abbildung 3.2). Weiter unterteilen die Autoren eine Trajektorie in *moves* und *stops* ein, wobei Erstere Bewegungen repräsentieren und sich bei Letzteren die Position nicht ändert (ähnlich Abbildung 3.10(b)). Physische Stopps müssen dabei nicht unbedingt semantische Stopps in der Trajektorie sein, je nach Anwendung kann dies der Nutzer definieren.

Im relationalen Schema (Abbildung 3.7) besteht eine Trajektorie (Entitätstyp *Trajectory*) aus den Komponenten *Begin*, *End* und *Stop* (B.E.S.), welche wiederum mit den *Moves* verbunden sind. Dadurch kann dem Beginn, dem Ende, jedem Stopp und jeder Bewegung einer einzelnen Trajektorie Zusatzinformation in jeder Form beigelegt werden. Beispielsweise kann dem Start einer Trajektorie (in B.E.S.  $\text{KIND}=\text{start}$ ) durch die topologische Beziehung zu *Country* der Ursprungsort angehängt werden. Gleich kann mit jedem Stopp auf dem Weg nach Afrika eine semantische Anreicherung des Landes, der vorherrschenden Vegetation usw. gemacht werden.

Spaccapietra *et al.* (2008) präsentieren zwei Verschiedene Datenmodelle, in Ersterem werden Trajektorien und deren Komponenten explizit in der Datenbank dargestellt (Abbildung 3.7 und 3.8), während im zweiten Modell möglichst viel Daten zu den Trajektorien im zweckbestimmten Datentypen *TrajectoryType* verborgen werden. Die Datenmodelle bauen

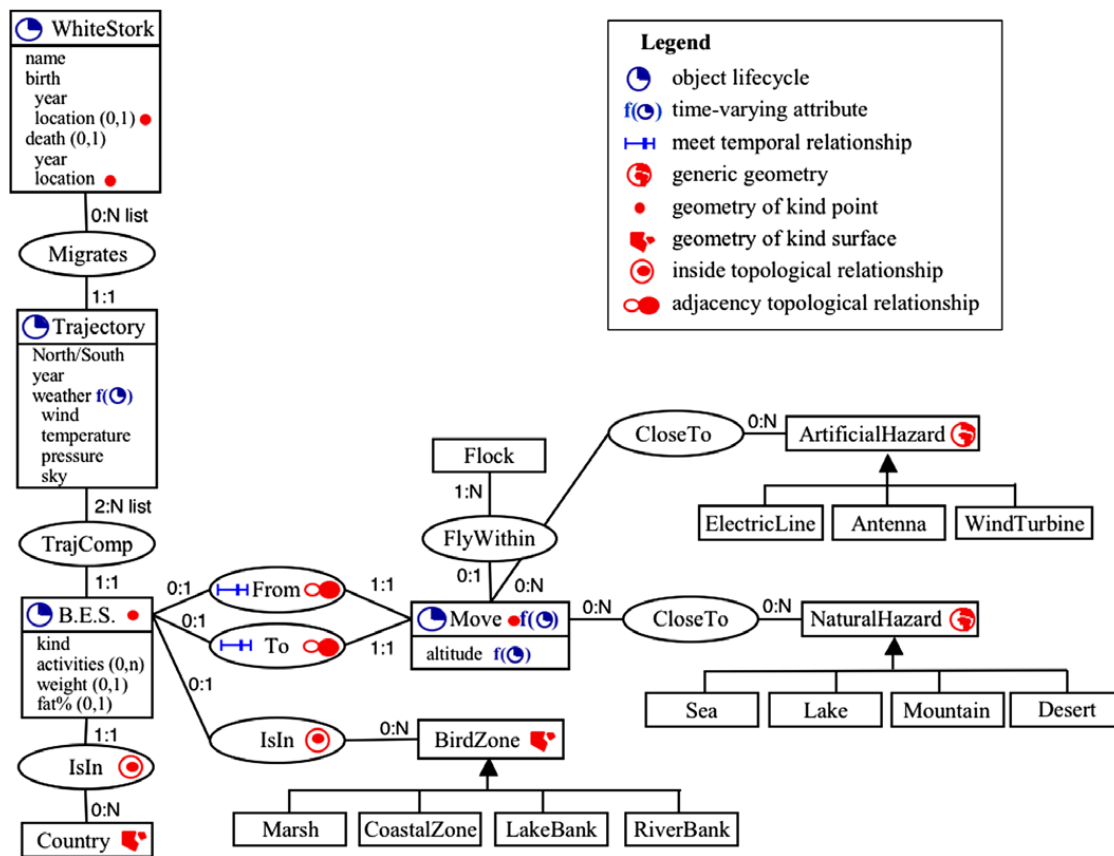


Abbildung 3.7: Relationales Schema am Beispiel der Storchmigration, aus Spaccapietra *et al.* (2008).

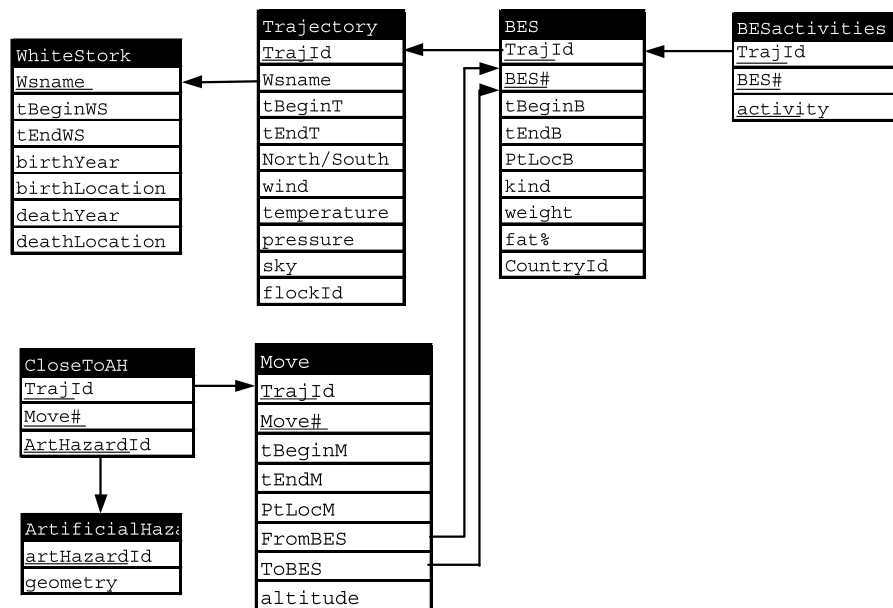


Abbildung 3.8: Relationales Datenmodell zu Abbildung 3.7, aus Spaccapietra *et al.* (2008).

auf MADS (*Modeling of Application Data with Spatio-temporal features*, (Parent *et al.*, 2006)) auf, einem konzeptuellen raum-zeitlichen Datenmodell mit den nötigen Funktionen (wie z.B. Unterstützung von zeitlich variierenden Attributen, raum-zeitlichen Objekten, usw.).

Bogorny *et al.* (2010) legen eine Erweiterung dieses Modells vor, welches auch implementiert wurde. Da Objekte in Bewegung automatisch Trajektorien produzieren, fokussiert ein weiterer Ansatz spezifisch auf die Speicherung von Trajektorien. Kuijpers und Othman (2010) beschreiben ein Datenmodell für die effiziente Speicherung von Trajektorien und ihren Unsicherheiten. Dabei wird auch eine spezielle Abfragesprache eingeführt, welche Queries direkt zu Eigenschaften der Trajektorie wie z.B. Geschwindigkeit erlaubt. Ein weiteres abstraktes Datenmodell für die Abfrage von *moving objects*, genannt FuMMO (*Future Movements of Moving Objects*), findet sich in Praing und Schneider (2007).

### 3.6 Analyse von Bewegungsdaten

Bei der Analyse von Bewegungsdaten kann der Fokus auf einem einzelnen, sich bewegenden Objekt oder auf einem ganzen Set von sich bewegenden Objekten beruhen. Andrienko und Andrienko (2007) beschreiben Ersteres als *individual movement behavior* (IMB) und die Analyse von mehreren Objekten *momentary collective behavior* (MCB) oder *dynamic collective behavior* (DCB). MCB beschreibt das Verhalten mehrerer Objekte während einer Momentaufnahme, beim DCB wird das Verhalten über eine Zeitspanne analysiert.

Anhand dieser Einteilung können zu IMB momentbedingte Charakteristiken wie Position zu bestimmtem Zeitpunkt, Geschwindigkeit, Richtung usw. analysiert werden. Bei der Betrachtung der gesamten (oder eines Ausschnitts der gesamten) Trajektorie eines *moving point objects* MPOs kann die gereiste Distanz, verbrauchte Zeit oder Muster in der Geometrie analysiert werden. Zwei oder mehr Trajektorien können des Weiteren miteinander auf Elemente wie Berührung, Kreuzung oder einfache Attribute (z.B. Länge) miteinander verglichen werden. Eine ausführliche Liste findet sich in Andrienko und Andrienko (2007). Dodge *et al.* (2008) können der Unterscheidung von MCB und DCB nicht viel abgewinnen und verweisen auf die Wichtigkeit von funktionalen Beziehungen, welche einzelne Objekte verbinden. Deswegen unterscheiden sie zwischen Gruppen, welche eine Gemeinsamkeit (wie z.B. Schafe einer gleichen Herde) besitzen und Kohorten, welche wenig relevante Übereinstimmungen (wie z.B. gleiches Geschlecht) aufweisen. Je nach Anwendung kann sich diese Einteilung von Gruppen und Kohorten ändern und unter Umständen das Geschlecht auch entscheidend sein.

In der GIScience wird stark an der Detektion allgemeiner Muster aus der Untersuchung von Bewegungsdaten geforscht, um Wissen aus diesen grossen Datensets generieren zu können. Die Analyse von MPOs wurde zu einem Schwerpunkt in manchen Forschungsprojekten (Dodge *et al.*, 2008), wobei noch keine einheitliche Terminologie zu Bewegungsdaten und ihrer Analyse besteht. Andrienko *et al.* (2008) präsentieren mögliche Typen von Fragen an ein Set von Bewegungsdaten, unterteilt in elementare und synoptische Fragen. Dabei ist das Ziel nicht nur die Charakterisierung und Beschreibung des gesamten Bewegungsverhaltens einer Sammlung von Objekten während einer Zeitperiode, sondern auch deren Erklärung (*connection discovery*: Korrelation, Abhängigkeit / Beeinflussung, strukturelle Verbindung). Eine andere Klassifizierung nach Dodge *et al.* (2008) sieht die Unterteilung in generische sowie Verhaltensmuster vor, wobei Erstere noch weiter unterteilt werden in primitive und zusammengesetzte Muster (*primitive / compound patterns*). Die Verhaltensmuster wiederum sind anwendungsorientiert, d.h. abhängig vom jeweiligen Kontext sowie

der Art des MPOs und bestehen aus generischen Mustern. Bei primitiven Mustern variiert nur ein Bewegungsparameter, ein Beispiel dafür ist das Zusammentreffen von zwei Objekten in Raum und Zeit (Abbildung 3.9). *Full co-incidence* könnte in einer Anwendung z.B. das Zusammentreffen zweier Storchschwärme bedeuten, während *lagged co-incidence* in einer anderen Anwendung die Verfolgung einer Herde Büffel durch Raubtiere darstellen könnte. Zusammengesetzte Muster (*compound patterns*) sind wiederum aus mehreren Primitiven zusammengesetzt und bestehen aus komplexen Beziehungen zwischen Objekten. Ein Beispiel wäre ein Zusammentreffen diverser Objekte (*encounter*, siehe Abbildung 3.10(a)). Die Autoren versuchen eine detaillierte und einheitliche Taxonomie von Bewegungsmustern zu erarbeiten, um die Erstellung von Algorithmen und visuellen Tools für die Erkennung von Mustern zu fördern.

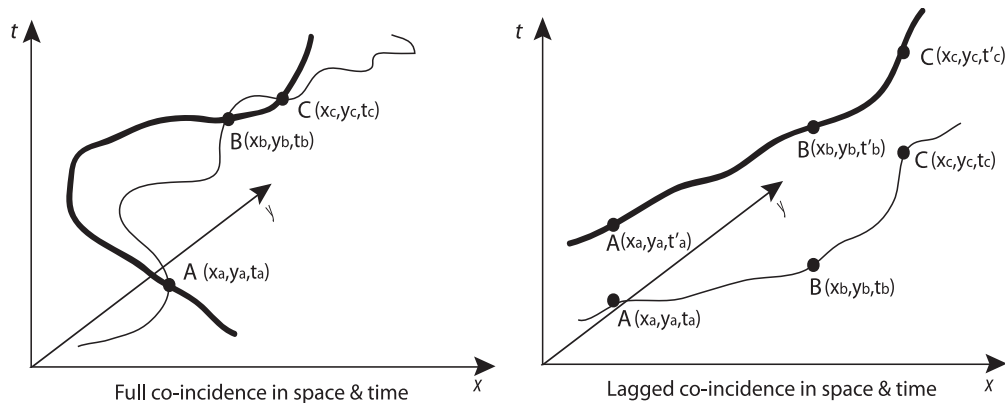


Abbildung 3.9: Mögliche primitive Muster (*primitive patterns*) von sich bewegenden Objekten, aus Dodge *et al.* (2008).

Mit einer Zusammenstellung von Methoden für die visuelle Analyse von Bewegungsdaten, verweisen Andrienko und Andrienko (2007) auf die kognitiven Fähigkeiten des Menschen, Muster zu erkennen. Zusammen mit dem Vorwissen des Analytikers kann mit visuellen Analysewerkzeugen Wissen aus Bewegungsdatensets gewonnen werden. Doch auch sie sehen Grenzen. Gerade bei riesigen Datenmengen seien computerunterstützte Methoden nicht zu umgehen und eine Kombination der Vorgehensweisen anzustreben.

So verweisen auch Laube *et al.* (2005) auf die Wichtigkeit der Verwendung von generischen Mustern, um unvoreingenommene Analysen mit computergestützten Methoden zu ermöglichen (*geographic knowledge discovery*, GKD). Mit rechenbetonten Verfahren sei die Entdeckung von mehr vordefinierten Mustern wie Konstanz, Übereinstimmung und *trend-setter* realisierbarer, als dies mit rein visuellen Methoden möglich wäre. Nichtsdestotrotz ist die Visualisierung ein wichtiger Bestandteil, z.B. um die detektierten Muster zu interpretieren.

Allgemein besteht das Problem, die (automatisch oder visuell) detektierten Muster in der Bewegung von Objekten einem anwendungsspezifischen Bewegungsverhalten (*movement behaviour*) zuzuweisen um damit Wissen zu generieren (Baglioni *et al.*, 2009). Denn das Bewegungsverhalten ist ein komplexer Prozess, bei welchem mehrere Faktoren die Bewegung selbst beeinflussen können, wie z.B. die Art des *moving object*, dessen Motivation und das geographische Umfeld in welchem sich die Objekte bewegen. In der Analyse von Bewegung wird (physischer) Raum immer noch oft als homogen betrachtet, obwohl dessen Eigenschaften stark variieren können. Gerade die Beschaffenheit des physischen Raumes



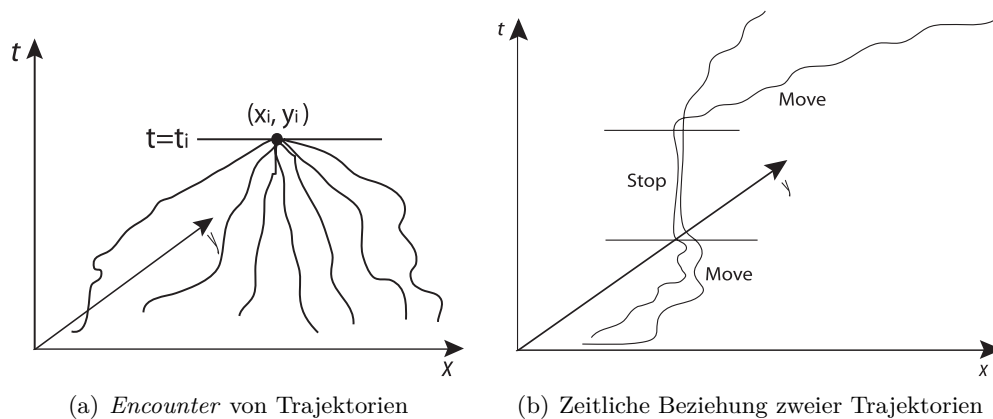


Abbildung 3.10: Zusammengesetzte Muster (*compound patterns*) von sich bewegenden Objekten: *encounter* aus Trajektorien mehrerer Objekte; zeitlich ähnliche Trajektorien zweier MPOs, unterteilt in *stops* und *moves*, aus Dodge *et al.* (2008).

(Höhe, Steigung, Land/Wasser, Vegetationstyp, usw.) kann grossen Einfluss auf die Bewegung haben und sollte bei der Analyse berücksichtigt werden (Andrienko *et al.*, 2008). Auch Dodge *et al.* (2008) sehen Bewegung in einer geographischen Umwelt immer als zweckbestimmt und nicht zufällig, weshalb der Kontext miteinbezogen werden sollte. Um die Komplexität von Bewegungsmustern aus 'rohen' Trajektorien zu verstehen, interpretieren und evaluieren, reichen auch hoch spezialisierte Mustererkennungsalgorithmen nicht aus und die Analyse ist auf Kenntnisse des geographischen Kontextes angewiesen (Baglioni *et al.*, 2009).

Die Forschung an der Analyse von MPOs hat in den letzten Jahren im Bereich GIScience stark zugenommen und viele raum-zeitliche Datenmodelle und Analysemethoden wurden präsentiert. Diese kamen jedoch kaum aus dem theoretischen Stadium heraus und wurden selten implementiert, geschweige denn in der Praxis angewendet (Dodge *et al.*, 2008).

### 3.6.1 Bewegungsanalyse in der Ökologie

Direkte Beobachtungen von Tieren können zwar detaillierte Einsichten in Verhaltensweisen liefern, doch die Daten sind nicht standardisiert und können deshalb nicht für die Widerlegung von ökologischen Hypothesen gebraucht werden (Cagnacci *et al.*, 2010). Das Konzept der *home range* ist nicht neu, bereits den ersten Ökologen war bewusst, dass Tiere nicht zufällig in der Gegend herumstreifen. Der Lebensraum eines Tieres umfasst die Fläche, die es in den alltäglichen Aktivitäten wie Futtersuche, Paarung oder Betreuung von Jungtieren durchquert. Gelegentliche Ausbrüche aus dem 'Territorium', z.B. aus Neugier sollten nicht zur *home range* gezählt werden (Kie *et al.*, 2010). Die Analyse hat sich von der Erstellung von simplen *minimum convex polygons* hin zu multidimensionalen relativen Streuungen in der Raumbenutzung der Tiere entwickelt. Mittlerweile sind Techniken zur Schätzung der Dichte, wie z.B. mit der Kernel-Methode (Abbildung 3.11), die beliebteste Methode, wobei immer komplexere, statistische Methoden zur Abschätzung der *home range* entwickelt werden (Kie *et al.*, 2010). Dabei spielt die Wahl der Glättungsparameter sowie die vorhandene Dichte an Positionsdaten eine entscheidende Rolle. Aus der Geostatistik wird die *kriging*-Methode gebraucht, welche räumliche Autokorrelation der Positionen annimmt. Neben diesen statistischen Methoden welche auf rein empirischen Daten beruhen, werden als relativ neue Methoden mechanistische Bewegungsmodelle für

die Erstellung von *home ranges* verwendet (Cagnacci *et al.*, 2010). Eine Übersicht über solche Modelle findet sich in Smouse *et al.* (2010) sowie in Fieberg *et al.* (2010), wo verschiedene Konzepte wie *random walk*, *Lévy flights/walks* oder *Brownian movement* diskutiert werden. Die Implementation solcher Modelle benötigt jedoch räumlich und zeitlich hoch aufgelöste Positionsdaten und ist somit eigentlich nur mit GPS-Technologie möglich. Die Einschränkungen bezüglich hoch aufgelösten Positionsdaten wurden bereits in Kapitel 3.3 erwähnt. Mit den technologischen Verbesserungen in naher Zukunft werden solche Modelle vermehrt zum Einsatz kommen und neue Einsichten in fein aufgelöste Habitate gewähren. Im Moment sind die Verfahren noch sehr komplex und benötigen Expertenwissen und komplexe Scripts, weshalb die Verwendung der traditionellen Methoden auch in naher Zukunft von Bestand sein wird (Kie *et al.*, 2010).

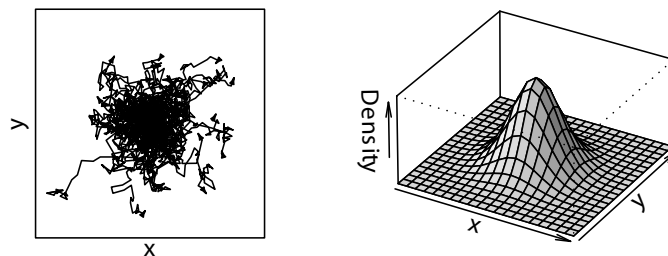


Abbildung 3.11: *Kernel Density* von *moving objects*, aus Morales *et al.* (2010).

Veränderungen von *home ranges* wurden traditionell mit Vergleichen statischer Positionen untersucht, z.B. von migrierenden Tieren zu verschiedenen Jahreszeiten. Mit dem Aufkommen von zeitlich hoch aufgelösten räumlichen Daten können die Lebensräume kleinerer zeitlicher Perioden miteinander verglichen werden und es ergeben sich neue Wege um die Wahl von Habitaten zu studieren (Fieberg *et al.*, 2010). Der Einbezug von zusätzlichen Kontextinformationen wie vorhandene Ressourcen, Orte von Artgenossen oder Raubtieren, natürliche Barrieren usw. ist essenziell um festzustellen wie und wieso ein Tier den Raum wie beobachtet benutzt (Kie *et al.*, 2010). Je feiner die zeitliche und räumliche Auflösung der Positionsdaten ist, desto weniger Information liegt im Datenpunkt an sich. Mit zunehmend 'kontinuierlichen Daten' sollte der Fokus weg von den einzelnen Punkten auf die Bewegungspfade gelegt werden (Kie *et al.*, 2010). Damit ist die Detektion der Verbindungen zwischen einzelnen Bereichen (z.B. zum Schutz vor Prädatoren in der Nacht und der Nahrungsaufnahme am Tag, oder Migration vom Sommer- zum Winterhabitat) möglich.

Neben *home range* oder *habitat / resource use* werden diverse weitere Masse verwendet, welche jedoch nicht einheitlich definiert sind. Theoretische *encounter* (Morales *et al.*, 2010) und *kill rates* (Merrill *et al.*, 2010) werden mit gemessenen Werten verglichen, um Rückschlüsse auf diverse Einflüsse zu testen. Owen-Smith *et al.* (2010) analysiert verschiedene Bewegungsmuster von Tieren auf Futtersuche. Es wird versucht von individuellen Bewegungsmustern auf Gruppen und von Gruppen auf ganze Populationen zu schliessen (Okubo *et al.*, 2001). Ein weiterer Forschungsbereich ist das Überwachen von Tieren über grosse Zeitperioden und Raumausdehnung für die Untersuchung komplexer Phänomene (Cagnacci *et al.*, 2010; Urbano *et al.*, 2010; Tomkiewicz *et al.*, 2010).

### 3.6.2 Abfragen zu raum-zeitlichen Objekten

Die Daten in raum-zeitlichen Datenbanken sind multidimensional und deshalb ist es eine grosse Herausforderung, eine verständliche Klassifizierung der Abfragen zu entwickeln. Die Identifikation der Query-Typen hat Einfluss auf Design des raum-zeitlichen Informationssystems: Query-Typen repräsentieren die Art von Information die der Benutzer erlangen will und reflektieren die Informationsstrukturen, in welchen die Nutzer Fragen stellen. Somit kann eine Klassifizierung der Abfragen Richtlinien für das Interface Design erbringen. Das Zusammenspiel von Indexstrukturen mit den Query-Typen vereinfacht das Finden der gewünschten Informationen. Anschliessend wird eine Klassifizierung in räumliche, zeitliche und raum-zeitliche Abfragen nach Yuan und McIntosh (2002) präsentiert.

Nachfolgend wird die Einteilung eingehender erklärt und für ein besseres Verständnis mit einfachen Beispielen ergänzt. Die Einteilung der Queries ist jeweils abhängig von der konkreten Modellierung der Datenbank, weshalb die Beispiele mit Sorgfalt zu geniessen sind. Weiter kann die Zuordnung der Query vom konkret abgefragten Zeitpunkt sowie den Zeitpunkten der gemachten Beobachtungen abhängen, je nachdem kann es z.B. eine Attribut- oder aber eine einfache zeitliche Abfrage sein.

#### Attribut Queries

Solche Queries können raum-zeitliche Eigenschaften besitzen und dennoch direkt abgefragt werden, sofern diese Daten bereitgestellt sind und nicht noch zuerst durch geometrische oder zeitliche Berechnung erstellt werden müssen.

Welche Länder haben eine Bevölkerungsdichte mit mehr als 300 Personen pro km<sup>2</sup>?

#### Räumliche Queries

Räumliche Abfragen werden weiter unterteilt in einfache, Abgrenzungs- und Beziehungsquery. Das Resultat dieser ist am effektivsten in graphischer Form.

**Einfache räumliche Abfrage** Diese Art von Abfrage ruft räumliche Entities zu dem gewünschten Kriterium auf. Es kann sich auch um räumliche Eigenschaften (z.B. Form, Dichte) handeln, welche noch berechnet werden müssten.

Wo befindet sich lehmiger Boden?

**Räumliche Abgrenzungsabfrage** Hier werden Informationen über Attribute oder Features innerhalb eines spezifizierten Gebietes gesucht. Dieses Gebiet kann geographisch durch Grenzen (z.B. ein Land) oder durch Nähe (z.B. innerhalb 1km zur Strasse A1) abgegrenzt werden. Dies ist eine der meist gebrauchten Abfragen, geographische Berechnungen sind nötig.

Wie viele Messstationen gibt es in der südlichen Ebene?

**Räumliche Beziehungsabfrage** Bei dieser Art von Abfrage werden Objekte und Phänomene untereinander nach ihren Beziehungen untersucht. Dies beinhaltet räumliche Relationen in Bezug zu Nähe oder Topologie, wobei Nähe hier nicht wie in der Abgrenzungsabfrage durch einen Buffer gehandelt wird, sondern z.B. durch Wahrscheinlichkeitswerte.

Welche Strassen queren Eisenbahnlinien?

### Zeitliche Queries

Gleich wie räumliche werden zeitliche Queries weiter klassiert in einfache, Abgrenzungs- und Beziehungsabfragen. In dieser Unterteilung wird nur die lineare Zeitstruktur beachtet (zyklisch, parallel und verzweigt wird ausgelassen).

**Einfache zeitliche Abfrage** Dabei werden Informationen zum Status eines Features an einem gewissen Zeitpunkt (Snapshot) gesucht. Der Unterschied zur Attribute-Query besteht darin, dass bei Nichtvorhandensein des abgefragten Zeitpunktes Berechnungen wie z.B. Interpolationen durchgeführt werden müssen. Die zeitliche Auflösung (granularity, resolution) ist entscheidend, ausserdem können je nach Art der Phänomene die Unterschiede zwischen den beobachteten Zeitständen abrupt oder graduell erfolgen.

Wer war der Besitzer dieses Hauses im Mai 2009?

**Zeitliche Abgrenzungsabfrage** In dieser Art Query wird die Information eines Merkmals während einer gewissen Zeitperiode gesucht. Es kann sich dabei um Entwicklung oder Rückgang eines Phänomens handeln und das Resultat sollte als statistische Zusammenfassung oder wenn nötig mit Karten, Animationen und weiteren Multimediatechniken präsentiert werden.

Wie oft hat der Besitzer dieses Hauses seit 2000 gewechselt?

**Zeitliche Beziehungsabfrage** Solche Abfragen untersuchen die zeitliche Beziehung von mehreren Entitäten untereinander. Analog zur räumlichen Abfrage besteht die zeitliche Beziehungsabfrage entweder aus zeitlicher Nähe oder zeitlicher Topologie. Zeitliche Nähe betrachtet Objekte innerhalb eines bestimmten Zeitintervalls (z.B. während des Sturms von Mitte Mai 2010), die Unterscheidung zur zeitlichen Abgrenzung wird durch die eher unklare Definition der Periode gemacht. Zeitliche Topologie kann durch Zeitpunkte oder Zeitintervalle begründet werden, es entstehen dadurch Beziehungstypen wie z.B. vor, nachher, während, überlappt, berührt, wobei mit der Intervall-basierenden Einteilung 13 Beziehungstypen bestimmt werden können (Allen, 1983).

Wie viele Erdbeben gibt es während eines grossen Sturms in der Regenzeit?

### Raum-zeitliche Queries

Die Suche nach Informationen wird sowohl in der räumlichen wie auch in der zeitlichen Dimension durchgeführt und kann raum- oder zeitdominant sein. Laufzeiten von Objekten und Phänomenen werden untersucht, um mit deren Entwicklung, Dynamik und Interaktionen die Beziehungen zusammenzufassen. Die dahinter liegenden Muster ermöglichen eine Erforschung und Erklärung der Phänomene und Prognosen können gemacht werden.

**Einfache raum-zeitliche Abfrage** In der einfachen raum-zeitlichen Abfrage gibt es drei mögliche Arten von Fragen: 1) wo ist das Objekt A am festgelegten Zeitpunkt, 2) wann sind die Objekte am bestimmten Ort und 3) was existiert an Stelle X zur festgelegten Zeit?

Wo war Präsident Obama am 20.04.2010?

**Raum-zeitliche Abgrenzungsabfrage** Dabei wird untersucht, was in einer Region in einem gewissen Zeitintervall passiert. Die Zunahme der Komplexität erfolgt durch die Erweiterung der räumlichen und zeitlichen Komponente von Punkten und Orten zu Regionen und Intervallen. Dabei können Änderungen nur Teile von Phänomenen betreffen, Objekte können sich teilen, verschmelzen oder ihre Eigenschaften fundamental ändern.

Welche Flächen haben zwischen 2000 und 2010 von Kultur- zu Siedlungsland geändert?

**Raum-zeitliche Verhaltensabfrage** In dieser Art Query werden Information über Änderungen betreffend Ort, Form, Grösse, Bewegungsmuster, Verteilung usw. von Objekten, Phänomenen, Ereignissen oder Prozessen der gleichen Art gesucht.

Wie hat sich der Niederschlag und die Temperatur während dem Sturm verändert?

**Raum-zeitliche Beziehungsabfrage** Im Gegensatz zu der raum-zeitlichen Verhaltensquery werden hier Objekte verschiedenen Typs miteinander verglichen. Dieser Abfragetyp ist sehr komplex, zeigt aber das Potential raum-zeitlicher Daten und die mögliche Untersuchung von Prozessen im Hintergrund.

Wo und wann wird die Flut Siedlungsgebiet erreichen?

Pelekis *et al.* (2004) verwenden in ihrer Übersicht zu raum-zeitlichen Datenbanken eine simplere, jedoch ähnliche Klassierung nach räumlichen, zeitlichen und raum-zeitlichen Eigenschaften. Zu ersterer Kategorie gehören Queries nach Position, räumlichen Eigenschaften und Relationen von statischen Objekten (*where, range / distance based, nearest neighbour, topology queries*). Analog werden die zeitlichen Abfragen unterteilt und beinhalten Queries ohne die räumliche Komponente. Die raum-zeitlichen Queries beinhalten beide Bestandteile und damit können Fragen zu diskret und kontinuierlich ändernden Objekten z.B. bezüglich Ähnlichkeit oder Verhalten über gewisse Zeit gestellt werden.

### 3.6.3 Abfragen in Moving Objects Databases

Zusätzlich zu den in 3.6.2 vorgestellten Queries klassifiziert nach Yuan und McIntosh (2002), sind in *moving object databases* mit der Abfragesprache *future temporal logic* (FTL) auch Queries über Zukunft möglich. Da in MODs die Bewegung eines Objektes durch einen Vektor repräsentiert wird und nur bei Überschreiten eines festgelegten Grenzwertes aktualisiert wird, gibt es immer eine Unsicherheit (*uncertainty zone* in Abbildung 3.12). Abfragen zu sich bewegenden Objekten mit Unsicherheit können in 'Punkt-' und 'Bereichsqueries' (*point & range queries*) unterteilt werden (Güting und Schneider, 2005). Erstere beinhalten Abfragen zu einem Objekt während einem festgelegten Zeitpunkt (**whereAt**) wo die erwartete Position (als zwei-dimensionaler Punkt) wiedergegeben wird. Mit **whenAt** können mögliche Zeiten abgefragt werden, wann ein Objekt einen Punkt (oder ein Set von Punkten) passieren wird.

Für Bereichqueries sind einige neue Konzepte notwendig, da ein Objekt immer oder nur irgendwann in einem festgelegten Gebiet sein kann (zeitliche Gültigkeit). Zweitens kann gefragt werden, ob das Objekt überall oder nur irgendwo in der Region sein wird (räumliche Gültigkeit) und drittens ob das Objekt definitiv oder nur möglicherweise die Abfrage

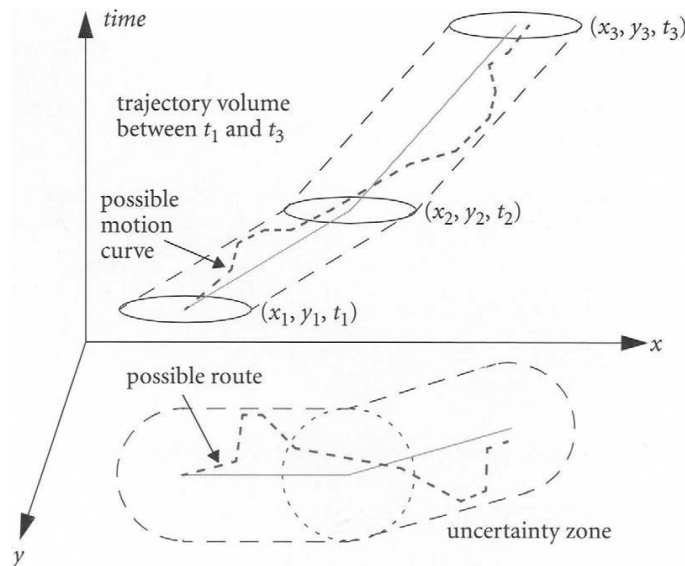


Abbildung 3.12: Bewegungsvektor mit Unsicherheitszone für eine mögliche Bewegung, aus Güting und Schneider (2005).

erfüllt (Gültigkeit der Unsicherheit). Aus den acht verschiedenen Kombinationen (und der semantischen Äquivalenz zweier) ergeben sich die sechs Prädikate *PossiblySometimeInside*, *PossiblyAlwaysInside*, *AlwaysPossiblyInside*, *DefinitelyAlwaysInside*, *DefinitelySometimeInside* und *SometimeDefinitelyInside* (Güting und Schneider, 2005). Andere Operatoren wie *eventually\_within*, *until*, *nexttime* oder *always\_for* könnten z.B. so eingesetzt werden:

```
RETRIEVE h
FROM helicopters h
WHERE eventually_within_15 (INSIDE(h, Valley) ^
always_for_5 (INSIDE(h, Valley)))
```

SQL 3.5: Beispiels-Query mit FTL, aus Güting und Schneider (2005).

Es werden dabei alle Helikopter wiedergegeben, welche im Tal innerhalb der nächsten 15 Minuten ankommen und für mindestens 5 Minuten bleiben können. FTL wurde im Zuge des DOMINO Projektes (Kapitel 3.7.1) entwickelt mit dem Ziel, auf Standard-DBMS ausführbar zu sein.

### 3.7 Anwendungen

Die Theorie zeigt, dass ein grosses theoretisches Wissen bezüglich Erfassung, Speicherung und Analyse von Bewegungsdaten vorhanden ist. Es stellt sich jedoch heraus, dass wenige dieser Datenmodelle und Analysenmethoden für raum-zeitliche oder Bewegungsdaten über das theoretische Stadium hinauskommen. Wenige Lösungen wurden implementiert, geschweige denn in der Praxis angewendet (Spaccapietra *et al.*, 2008; Dodge *et al.*, 2008; Silva und Santos, 2010). In den nächsten Kapiteln sollen deshalb die drei bekanntesten Prototypen aus der MOD-Forschung sowie Beispiele aus der angewandten (Bewegungs-) Ökologie vorgestellt werden.

### 3.7.1 MOD-Forschungsprototypen

Das *moving objects spatio-temporal data model* (MOST), als eines der ersten Datenmodelle mit dynamischen Attributen (siehe Kapitel 3.5.5) wurde als Erweiterung eines DBMS' implementiert. Dieser *moving objects database* Prototyp DOMINO (*Databases fOr MovINg Objects*) von Wolfson *et al.* (1999) erweitert bestehende objekt-relationale Datenbanken mit dynamischen Attributen und einer spezifischen Querysprache (FTL), um sich bewegende Objekte zu speichern und abzufragen.

Güting *et al.* (2010) implementierten in SECONDO das bereits vorgeschlagene Datenmodell aus Güting und Schneider (2005), siehe auch Kapitel 3.5.5. Mit räumlichen, zeitlichen und neuen raum-zeitlichen bzw. *moving objects* Datentypen können bestehende relationale, objekt-orientierte und andere DBMS Datenmodelle erweitert werden. Die Position eines sich bewegenden Objektes wird dabei als gewöhnlicher zeitlich variierender räumlicher Datentyp behandelt. Dieser Prototyp wurde konstruiert, um die Entwicklung von raum-zeitliche Algorithmen und Anwendungen zu unterstützen.

Mit HERMES präsentieren Pelekis *et al.* (2006) ein System für das Modellieren und Abfragen von beweglichen Objekten mit einem etwas anderen Ansatz. Anstatt einer Datenbank für die Speicherung von *moving objects* bietet die *HERMES Moving Data Cartridge* (HERMES-MDC) eine Sammlung von Trajektorien-Datentypen und -Operationen, basierend auf der Kombination der (statischen) räumlichen Datentypen von *Oracle Spatial* sowie zeitlichen Typen aus *TAU Temporal Literal Library* (TAU-TLL) (Pelekis *et al.*, 2002). Dieser MODBMS-Prototyp kann als rein räumliches oder zeitliches Datenbanksystem verwendet werden, wenn auch durch die *moving data types* der Fokus klar auf der Speicherung, Abfrage und Repräsentation von beweglichen Objekten liegt.

Diese drei bekanntesten Systeme für die Verwaltung und Abfrage von *moving objects* sind jedoch immer noch Forschungsprototypen und für kommerzielle Zwecke zu wenig ausgereift. Die handelsüblichen Datenbankmanagement Systeme bieten mit ihren räumlich und zeitlich getrennten Funktionalitäten (namentliche Beispiele *Oracle Spatial*, *SQL Server Spatial*, *PostgreSQL mit PostGIS*) genug Support für die meisten Anwendungen, besonders für die Anforderungen von räumlichen Daten wie in Geographischen Informationssystemen (GIS). Für kommerzielle Produkte von ausdrücklich auf *moving objects* ausgelegten Managementsystemen benötigt es weitere Forschung, oder wie es Yan (2011, 15) ausdrückt:

*'There is still a long way to go for the industry of moving object or trajectory databases.'*

### 3.7.2 Anwendungen in der Bewegungsökologie

Die zunehmende Komplexität und Quantität der Bewegungsdaten fordert geeignete Analysewerkzeuge mit vordefinierten Funktionen (Kie *et al.*, 2010; Cagnacci und Urbano, 2008; Cagnacci *et al.*, 2010; Urbano *et al.*, 2010). Denn aktuell müssen die Algorithmen für die Methoden aus der Theorie selbst geschrieben werden, was vertiefte Programmierfähigkeiten voraussetzt. Da Datenbanklehre nicht zum normalen Ausbildungsweg von Biologen und Ökologen gehört, werden in der Speicherung und Analyse oftmals Tabellenkalkulationsprogramme wie *Microsoft Excel* verwendet, welche flexibel und einfach zu lernen sind, sowie schnelle Ergebnisse liefern (Jones *et al.*, 2006). *Spreadsheet*-Programme liefern jedoch nicht die Strukturen für langfristiges Datenmanagement geschweige denn für räumliche Analysen von Bewegungsdaten. Deswegen greifen Forscher immer öfters zu relationalen DBMS

wie beispielsweise *Microsoft Access* und ergänzen diese mit Statistikprogrammen wie z.B. *R*, wobei das Knowhow selbst angeeignet werden muss. Auch wenn die Verwaltung und Analyse von Bewegungsdaten immer komplexer wird, müssen sich die Forscher aus Biologie und Ökologie damit befassen (Urbano *et al.*, 2010, 2183):

*'Data management is increasingly becoming a necessary skill for ecologists, as has already happened with statistics and GIS'*

In diesem Kapitel sollen einige Softwarelösungen vorgestellt werden, mit welchen in der Ökologie und ähnlichen Wissenschaften Bewegungsdaten von wild lebenden Tieren verwaltet und analysiert werden. Das Augenmerk gilt vor allem Softwarelösungen, welche auf GPS-Positionierungsdaten zugeschnitten sind. Denn viele Software-Tools für Studien an wild lebenden Tieren wurden für Analysen von VHF-Daten entwickelt und sind deshalb auf kleine, diskontinuierliche Datensätze ausgerichtet (Urbano *et al.*, 2010). Das bedeutet, dass diese mehr auf die Analyse und weniger auf das Datenmanagement fokussiert sind. Durch die immer grösser werdenden Datenmengen wird die Verwaltung jedoch immer wichtiger und es sind gut ausgedachte Systeme notwendig. Sogenannte *spatial information systems* (SIS) (Beyer *et al.*, 2010) beinhalten verschiedene Konzepte, Techniken und Technologien zur Erfassung, Speicherung, Visualisierung und Analyse von räumlicher Information. Diese umfassen Geographische Informationssysteme (GIS), *global positioning systems* (GPS), Fernerkundung und relationale Datenbank Management Systeme (RDBMS), welche relativ unabhängig voneinander entwickelt wurden, jedoch zunehmend in zusammenhängenden Systemen verwendet werden.

Eines dieser Systeme für Analyse und Management von Wildtierdaten ist das *Information System for Analysis and Management of Ungulate Data* (ISAMUD) (Cagnacci und Urbano, 2008; Urbano *et al.*, 2010). Die modulare Plattform beruht auf der objekt-relationalen Datenbanklösung *PostgreSQL*<sup>2</sup> und deren Erweiterung für geographische Objekte *PostGIS*<sup>3</sup>, welche die räumliche Datentypen und Funktionen nach den *Open Geospatial Consortium* (OGC)-Standards<sup>4</sup> verwendet. Ähnlich wie bei der räumlichen Erweiterung *Oracle Spatial* kann *PostgreSQL* in Kombination mit *PostGIS* als *Back-End* Datenbankserver für Geographische Informationssysteme genutzt werden. ISAMUD erlaubt die Integration der GIS-Lösungen *ArcGIS*, *QGIS*<sup>5</sup> sowie *GRASS*<sup>6</sup>, womit Analysen getätigt werden können. Weiter können mit dem Statistikprogramm *R*<sup>7</sup> eigene oder mit sogenannten *packages* integrierte Algorithmen ausgeführt werden. Als Datenbankmanagement-Tools bieten sich *pgAdmin iii*<sup>8</sup> oder *Microsoft Access* an und Webzugriff kann über *phpPgAdmin*<sup>9</sup> (Zugriff auf Relationen) oder *MapServer*<sup>10</sup> (Kartenerstellung) erfolgen. In einem solch komplexen System ist die Datenmodellierung enorm wichtig, Urbano *et al.* (2010) präsentieren das Kernelement eines Informationssystems für Wildtierdaten in Abbildung 3.13.

Die vorgelegte Grundlage des Datenbankmodells sieht folgendermassen aus: Die GPS-Geräte liefern Koordinaten der Tiere für einen bestimmten Zeitpunkt und werden in eine

<sup>2</sup><http://www.postgresql.org/> 27.12.2011

<sup>3</sup><http://postgis.refractory.net/> 27.12.2011

<sup>4</sup><http://www.opengeospatial.org/standards> 27.12.2011

<sup>5</sup><http://www.qgis.org/> 27.12.2011

<sup>6</sup><http://grass.fbk.eu/> 27.12.2011

<sup>7</sup><http://www.r-project.org/> 11.01.2012

<sup>8</sup><http://www.pgadmin.org/> 27.12.2011

<sup>9</sup><http://phpPgAdmin.sourceforge.net/doku.php?id=start> 27.12.2011

<sup>10</sup><http://mapserver.org/input/vector/postgis.html> 27.12.2011



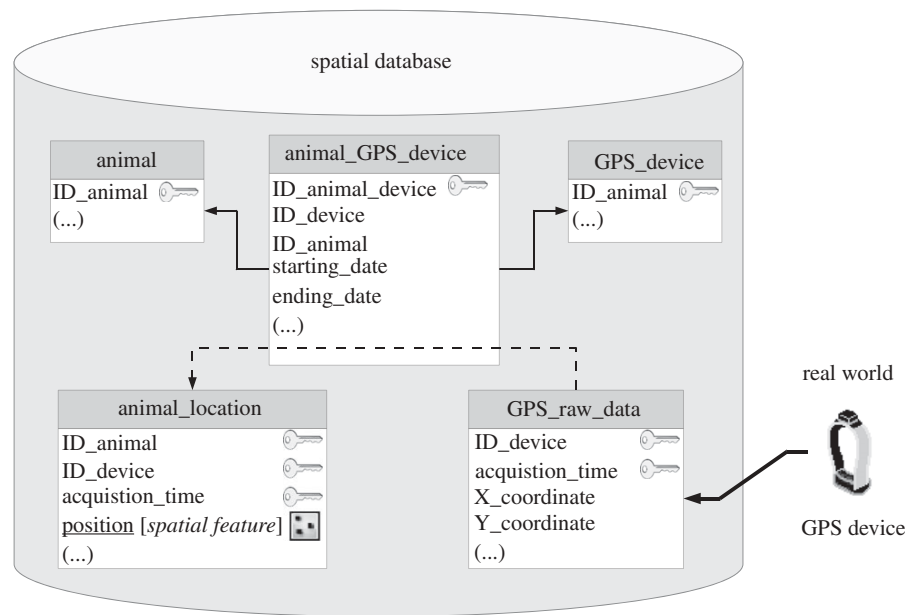


Abbildung 3.13: Generelle Repräsentation eines Schemas für ein Datenbankmodell zugeschnitten für Wildtierdaten, aus Urbano *et al.* (2010).

Tabelle der Datenbank geladen (GPS\_raw\_data). Die Geräte (GPS\_device) werden je einem Tier (animal) für eine bestimmte Zeitdauer (starting\_date, ending\_date) zugewiesen. Mit einem Skript (SQL 3.6) werden anhand der acquisition\_time und der ID\_device in GPS\_raw\_data die rohen GPS-Daten über die Tabelle animal\_GPS\_device dem Tier zugewiesen, welches das konkrete Halsband gerade im Moment der GPS-Messung trug. Anhand der Koordinaten wird ein räumliches Objekt erstellt und mit ID\_animal, ID\_device und der acquisition\_time in die Tabelle animal\_location eingefügt (gestrichelter Pfeil). Dadurch werden Positionen nicht nur als Paare von Nummern (Koordinaten), sondern als komplexe Objekte (*spatial data type* SDT) angehäuft. Zusätzlich wird noch die Zeit des Updates (eine Art *transaction time*) vermerkt und die Höheninformation zu jedem *fix* gespeichert. Weitere Daten, welche mit dem GPS-Gerät erzielt werden (wie z.B. horizontale oder vertikale Ungenauigkeiten, Batteriestatus, TimeToFirstFix, Daten aus Pedometer oder Accelerometer, usw.) können durch kleine Modifikationen im Skript ebenfalls in Tabelle animal\_location übernommen werden.

```
INSERT INTO animal_location (id_device, id_animal, acquisition_time, height_gps,
                             update_time, the_geom)
SELECT a.id_device, a.id_animal, r.acquisition_time, r.height, localtimestamp,
       ST_SetSRID(ST_MakePoint(r.longitude, r.latitude), 4326)
FROM animal_GPS_device a, GPS_raw_data r
WHERE r.id_device = a.id_device
AND
  (((r.acquisition_time) >= a.start_time
  AND
  (r.acquisition_time) <= a.end_time))
OR
  (((r.acquisition_time) >= a.start_time
  AND
  (r.end_time) Is Null));
```

SQL 3.6: Zuweisung der rohen GPS-Daten an Individuen im ISAMUD-System.

Über Trigger werden die Punktgeometrien mit Kontextinformationen wie Bodenbedeckung, Wetterdaten Tag/Nacht-Zeiten usw. verschnitten und die relativen Attribute aktualisiert. Die Verknüpfung mit GRASS ermöglicht über ein Skript die Verschneidung der einzelnen GPS-Messpunkte mit Rasterebenen, z.B. aus Fernerkundungsdaten. Oftmals sind diese jedoch entweder hoch aufgelöst und teuer oder mit ungenügender Auflösung und dafür günstig oder sogar gratis. In Urbano *et al.* (2010) findet sich eine Auflistung von Fernerkundungsdaten zu ökologischen Indikatoren.

Ein anderes Werkzeug, welches mehr auf die Analyse und weniger das Management der Bewegungsdaten fokussiert, ist das *R-package adehabitat* von Calenge *et al.* (2009). Trajektoriendaten können nach diversen Eigenschaften untersucht und dann auch visualisiert werden. Ein Screenshot der Klasse *ltraj* aus diesem Paket wird in Abbildung 3.14 gezeigt. Es werden jedoch vor allem Möglichkeiten für eine geometrische Analyse gegeben und Kontextinformationen zur Umwelt oder Daten zum Verhalten der einzelnen Tiere werden nicht beachtet. Das bei Biologen und Ökologen beliebte *adehabitat-package* kann auch im ISAMUD-Gefüge eingesetzt werden und mit dessen Vorteilen kombiniert werden.

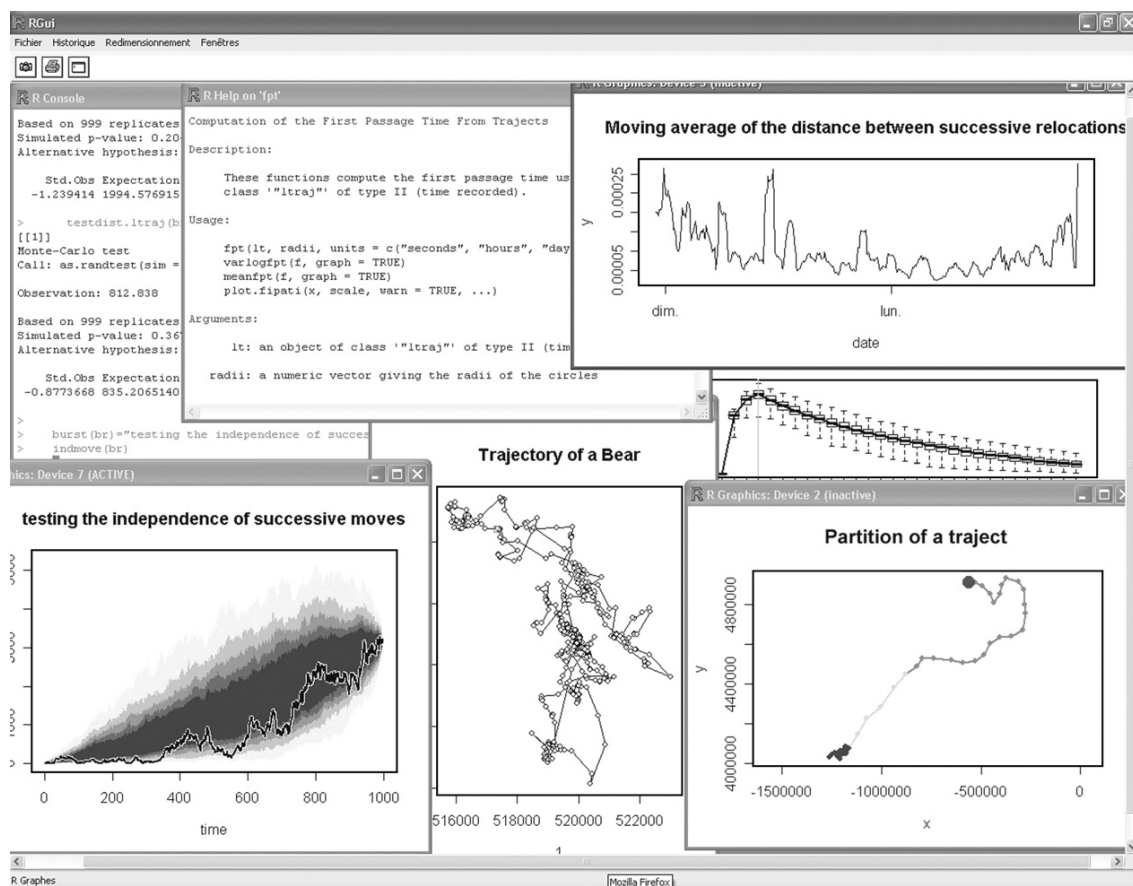


Abbildung 3.14: GUI (*graphical user interface*) von *R* mit Klasse *ltraj*, aus Calenge *et al.* (2009).

Um Wissen über die ursprünglichen, regional und zeitlich beschränkten Forschungsprojekte hinaus zu generieren, werden zunehmend studien- und länderübergreifende Projekte lanciert. Ein Beispiel für ein solches *longterm monitoring*-Programm ist das European

ROe DEER Information System (EURODEER<sup>11</sup>), wo die ISAMUD-Datenstruktur angewendet. Mit dem Zusammentragen von Bewegungsdaten zu Rehwild aus diversen Quellen soll diese Spezies in einem biologischen, ökologischen sowie evolutionären Kontext studiert werden um ein komplettes Bild dieses Tieres zu erstellen. Die Daten werden einem immer breiteren Spektrum von Forschern und auch der Öffentlichkeit zur Verfügung gestellt. In diesem Bereich haben Kranstauber *et al.* (2011) mit *Movebank*<sup>12</sup> eine Pionierleistung erbracht. Dort können Forscher mit einem relativ simplen Webinterface Bewegungsdaten austauschen um damit Analysen zu machen. Auf dem *Google Maps*-Hintergrund können die einzelnen Datensätze auch zuerst einmal dargestellt werden. Es existieren bereits viele Datensätze, welche per Registrierung zugänglich sind (siehe Punkte in Abbildung 3.15) und die Datenbank wächst weiter. Das darunter liegende konzeptuelle Datenmodell ist ähnlich dem in ISAMUD aufgebaut und verwendet die Begriffe *animal*, *tag*, *tag deployment*, *observations*, *sensor* und *other measurements* (Details in Kranstauber *et al.* (2011)).

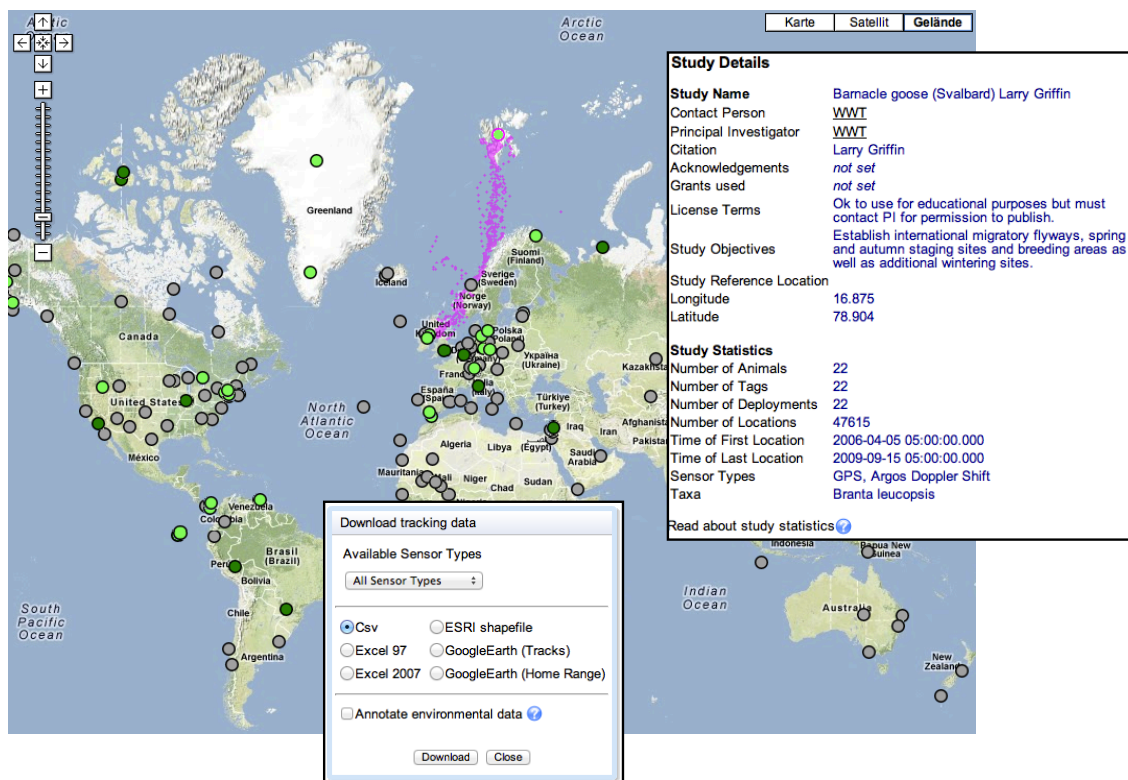


Abbildung 3.15: Ausschnitte von *Movebank* mit den zahlreichen Bewegungsdatensätzen, repräsentiert durch grüne Punkte. Einfaches GUI für Sichtung (Punkte in pink) und Download der Bewegungsdatensätze. Quelle: *movebank.org*

Ein ähnliches Projekt ist WRAM (*Wireless Remote Animal Monitoring*<sup>13</sup>), wo Positionsdaten zu 1446 Tieren aus 12 Spezies und 4 Ländern gespeichert sind (Stand: November 2011). Insgesamt können somit rund 70 Mio. Datenpunkte über einen einzelnen Zugriff abgefragt werden, ohne dass dieselbe Query in verschiedenen Datenbanken mit unterschiedlichen

<sup>11</sup><http://sites.google.com/site/eurodeerproject/> 04.12.2011

<sup>12</sup><http://www.movebank.org/> 05.01.2012

<sup>13</sup><http://www-wram.slu.se> 05.01.2012

Benutzer-IDs und variierenden Output-Tabellen eingespeist werden muss. Die Datenbank beherrscht 10 Datenformate und 6 verschiedene Sensortypen (GPS, VHF, usw.). Den Bedarf zu Kollaborationen zwischen einzelnen Projekten sieht auch Vesco *et al.* (2011), welcher verschiedene Datensätze von lokalisierten Infektionskrankheiten mit *Microsoft Access* zusammentrug. Um doppelte Einträge zu vermeiden und eine gute Datenqualität zu behalten, sei jedoch eine Überwachung der Uploads unbedingt notwendig.

*Oracle* als einer der führenden Anbieter für Datenbanklösungen bietet mit der räumlichen Erweiterung *Spatial* und der Möglichkeit zur Verbindung mit *ArcGIS* eine gute und komplette Lösung für Datenmanagement und -analyse. Für Analysen von Tierbewegungen existieren zusätzliche Erweiterungen für den *Spatial Analyst* wie z.B. *Animal Movement*<sup>14</sup> vom *United States Geology Survey* (USGS). Diese wird jedoch nicht mehr aktualisiert. Eine ziemlich aktuelle Liste von *spatial information systems* bzw. Werkzeuge für Datenmanagement und deren Analyse findet sich in Beyer *et al.* (2010).

Die Verwendung von proprietärer Software wie *Oracle* oder *ArcGIS* zieht gewöhnlich hohe Lizenzkosten mit sich. Die begrenzten finanziellen Mittel von biologischen und ökologischen Forschungsprojekten können eher bei der Software als bei der Hardware, beispielsweise GPS-Halsbändern, gespart werden. Im Bereich Wildtierforschung zeigt sich ein Trend zu kostenlosen Software-Paketen (Calenge *et al.*, 2009; Urbano *et al.*, 2010) und diverse Projekte setzen voll und ganz auf *Open Source* Lösungen, so auch ISAMUD oder das *Ani-move*-Projekt<sup>15</sup>.

### 3.8 Forschungslücken

Gewisse Konzepte der Verhaltensforschung an Tieren in freier Wildnis wie z.B. das der *home range* sind fast so alt wie die Ökologie selbst, was ungefähr der Zeit Darwins Mitte des 19. Jahrhunderts entspricht (Kie *et al.*, 2010). Die Art und Weise jedoch wie diese erfasst und analysiert werden, hat sich in den letzten 35 Jahren mit dem Aufkommen moderner Fernmesstechniken wie VHF und vor allem GPS verändert (Tomkiewicz *et al.*, 2010). Was nach wie vor fehlt ist eine in der Ökologie einheitliche Terminologie für die Beschreibung von Bewegungen und je nach Spezies gelten andere Anhaltspunkte für das Sammeln der Daten, was auch mit den Einschränkungen von Seiten der Technologie zusammenhängt: Beschränkte Batterielaufzeit führt zum Zielkonflikt zwischen langen Messperioden mit wenigen *fixes* und kurzen Perioden mit zeitlich hoch aufgelösten Positionsdaten. Trotz gleicher Tierart und gleichem Sensortyp können völlig unterschiedliche Parameter verwendet und damit die Bewegungsdaten ungewollt beeinflusst werden.

Auch Hebblewhite und Haydon (2010) sehen das Sammeln von räumlich und zeitlich hoch aufgelösten Bewegungsdaten von Tieren und die Analyse durch fortgeschrittene, quantitative Methoden nicht nur als Segen für die ökologische Forschung und sind dementsprechend kritisch eingestellt. Durch die hohen Kosten der Positionierungstechniken müsse die Anzahl der zu untersuchenden Tiere reduziert werden, was unter Umständen die statistische Signifikanz gefährden könnte. Vor allem sprechen die Autoren die Tatsache an, dass die (räumlich und zeitlich) fein aufgelösten Bewegungsdaten nicht mit entsprechenden Ressourcen- sowie Verhaltensdaten verglichen werden, obwohl gerade dort Fortschritte in der bewegungsökologischen Forschung erwartet werden. Auch Kie *et al.* (2010) vertreten

<sup>14</sup>[http://alaska.usgs.gov/science/biology/spatial/gistools/index.php/animal\\_mvmt.htm](http://alaska.usgs.gov/science/biology/spatial/gistools/index.php/animal_mvmt.htm) 05.01.2012

<sup>15</sup><http://www.faunalia.it/animove> 05.01.2012

die Meinung, dass noch so hoch aufgelöste VHF- oder GPS-Daten ohne zusätzliche Angaben zu Kontext (Habitate, Ressourcen, natürliche Feinde, Erinnerung, ...) unbrauchbar sind.

Nicht nur in Biologie und Ökologie werden Bewegungsdaten meist bloss anhand ihrer geometrischen Charakteristiken untersucht. Obwohl auch Dodge *et al.* (2008) Bewegung immer als zweckbestimmt in einem geographischen Umfeld ansehen, wird die Information zum Kontext in der Analyse meist nicht miteinbezogen. Der Fokus in der GIScience liegt im Bereich der *moving point objects*, wobei auch hier keine einheitliche Terminologie zu Bewegungen sowie deren Eigenschaften und Parameter besteht.

Neben den unterschiedlichen Erfassungstechniken für die Bewegungsdaten besteht eine Vielzahl von unterschiedlichen Analysemethoden zu den einzelnen biologischen Konzepten wie *home range*, Verteilung oder Ressourcenverbrauch, unzählige statistische Methoden und Algorithmen werden präsentiert. Die komplexeren Verfahren aus der Theorie, welche Modelle von Tierbewegungen miteinbeziehen, haben das Potential um neue Einsichten in fein aufgelöste Muster der Habitatwahl zu gewinnen. Solche Ansätze benötigen wegen der grossen und komplexen Datenmenge generell hohe Rechenleistung und vor allem selbstgeschriebene, massgeschneiderte Algorithmen (Fieberg *et al.*, 2010).

Es bestehen zwar einige Programme für ökologische Analysen (ausführliche Liste vom *Illinois Natural History Survey* INHS auf <http://nhsbig.inhs.uiuc.edu/>), doch es scheint sich kein Standard durchzusetzen und Datenmanagement wird nicht als sehr wichtig betrachtet. Obwohl gerade für Langzeitstudien oder eine Zusammenarbeit mehrerer Universitäten solche Datennormen von grosser Wichtigkeit sind und ein gemeinsames Wirken sowie neue Erkenntnisse erst ermöglichen, wurde dies vernachlässigt. Es werden simple, jedoch weitläufig brauchbare Datenstrukturen benötigt, welche zudem den Einbezug von forschungs- oder gattungsspezifischen Analysewerkzeugen erlauben.

Rund um die Forschung an *Spatio-Temporal* sowie *Moving Objects Databases* werden zwar diverse Datenmodelle für die Speicherung von sich bewegendenden Objekten präsentiert, diese kommen jedoch kaum über das konzeptuelle Stadium hinaus. Es werden zusätzliche Schritte benötigt, nämlich die Entwicklung und das Testen von weiteren Datenbankprototypen, um detaillierte Anforderungen und Forschungsrichtungen zu ermitteln (Pelekis *et al.*, 2004).

Für die Speicherung von Bewegungsdaten aus der Wildtierforschung muss deshalb auf handelsübliche Software zurückgegriffen werden, wobei im Moment relationale Datenbanksysteme das grösste Potential aufweisen (Urbano *et al.*, 2010) und mit dem flexibel einsetzbaren ISAMUD-Gefüge eine interessante Datenstruktur-‘Vorlage‘ vorliegt.

Die Kombination von automatisch bezogenen GPS-Bewegungsdaten, Kontextinformationen, z.B. durch Fernerkundung und Verhaltensdaten aus traditioneller Feldarbeit, wie sie Hebblewhite und Haydon (2010) fordert, ist auch genau das Ziel des *Inkawu Vervet Projects*. Neue wissenschaftliche Erkenntnisse sollen durch Zusammenführen der verschiedenen Datenquellen gewonnen werden. Im Moment sind diese Daten teils noch lose und nicht miteinander verknüpft, weshalb sie und für eine Analyse manuell zusammengeführt werden müssen, was zeitaufwändig ist und zudem eine Fehlerquelle darstellt.

Deshalb würde ein ausgeklügeltes Datenbankmanagementsystem mit der Verbindung der hier wichtigsten Elemente Bewegung, Ressourcen (entspricht der Umwelt) und Verhalten der Tiere die Arbeit der Forscher immens vereinfachen und zudem die Fehlerquote bei der noch komplizierten Analyse verkleinern. Die Grundlage dafür ist ein gut durchdachtes Daten-Konzept, welches in Zusammenarbeit mit den Forschern erarbeitet werden muss. Gerade die Vernetzung von verschiedenen raum-zeitlichen Elementen stellt in kommer-

ziellen Datenbankmanagementsystemen (DBMS) eine Herausforderung dar und benötigt geschickte Datenstrukturen, da die Systeme eigentlich nicht für solche Anwendungen entwickelt wurden. Es stellt sich die Frage, wie die Daten sinnvoll gespeichert und inwieweit die für das Forschungsprojekt notwendigen Analysen mit einer solchen Lösung durchgeführt werden können.

# Kapitel 4

## Methodik

In den folgenden Kapiteln wird das Vorgehen festgehalten, wie die angestrebten Ziele erreicht werden sollen und eine kritische Auseinandersetzung mit den Forschungsfragen ermöglicht werden kann.

In einem ersten Schritt soll wie bei einem normalen Datenbankentwurf eine Anforderungsanalyse des Inkawu Vervet Projektes durchgeführt und damit das konzeptuelle Modell erstellt werden. In einem weiteren Schritt soll gezeigt werden, was mit den vorhandenen Daten durch ein geschicktes Verwenden der raum-zeitlichen Elemente möglich wäre. Anhand einiger Beispiele wird in ausgewählten Bereichen die mögliche Datenstruktur aufgezeigt und mit einer räumlichen Datenbank eine mögliche Implementierung vorbereitet.

### 4.1 Anforderungsanalyse

Die erste Phase eines Datenbankentwurfs ist meist sehr zeitaufwändig, für den Erfolg einer Datenbank jedoch von sehr grosser Bedeutung, denn die Korrektur eines Anforderungsfehlers ist meist viel teurer als die Korrektur eines Implementierungsfehlers (Elmasri und Navathe, 2009; Teorey *et al.*, 2011). Da der Prozess der Datenbankerstellung meist nicht von den Auftraggebern, sondern von externen Fachpersonen durchgeführt wird, ist ein Miteinbezug der zukünftigen Nutzer und damit deren vertieftes Wissen vom Anwendungsgebiet unerlässlich.

Die Grundziele der Anforderungsanalyse formuliert Teorey *et al.* (2011) wie folgt:

- 1) Abgrenzung der Datenanforderungen in Bezug auf grundlegende Datenelemente.
- 2) Ausführliche Beschreibung der Datenelemente und der Beziehungen untereinander, damit die Anforderungen modelliert werden können.
- 3) Bestimmung der beabsichtigten Typen von Transaktionen, welche in der Datenbank ausgeführt werden sollen, sowie die Interaktionen zwischen den Transaktionen und den Datenelementen.
- 4) Definition von Einschränkungen betreffend Performance, Integrität, Sicherheit sowie administrativen Auflagen der Datenbank.
- 5) Angabe der Einschränkungen bezüglich Design und Implementierung (spezifische Technologie, Hardware / Software, Programmiersprachen, Richtlinien, Normen oder externe Schnittstellen).

6) Umfassende Dokumentation des Vorgehens.

Die Punkte 4) und 5) dieser eher generell verfassten Ziele wurden im Laufe dieser Anforderungsanalyse nicht gross beachtet, da diese fest in die Richtung der Implementierung gehen und der Schwerpunkt auf dem konzeptuellen Modell liegt. Die Erfassung und Analyse der Anforderungen nach Elmasri und Navathe (2009) zeigt das grundlegende Vorgehen auf:

- 1) Identifikation der Anwendungsbereiche und Benutzergruppen, Kontaktpersonen für nachfolgende Schritte der Erfassung und Spezifikation von Anforderungen wählen.
- 2) Vorhandene Dokumentation zu Anwendungen sichten und analysieren, z.B. Handbücher, Formulare, Berichte und Organisationsdiagramme prüfen, um Einfluss auf Erfassung und Spezifikation der Anforderungen zu ermitteln.
- 3) Aktuelle Betriebsumgebung und geplante Nutzung der Informationen untersuchen: Analyse der Transaktionsarten und ihrer Häufigkeit sowie Informationsfluss innerhalb des Systems.
- 4) Kontaktpersonen können interviewt werden, um den Informationswert und -gehalt in Bezug auf Prioritäten festzustellen.

Spezifisch auf das Inkawu Vervet Projekt angewendet gilt es die übergeordneten Anforderungen an die Datenbank und die Eingrenzung des Anwendungsbereiches zu bestimmen, die Analyse der Datenstruktur und deren möglichen Inhalte zu ermitteln, sowie die möglichen Anwendungen auszumachen, welche mit der Datenbank durchgeführt werden sollen. Die Informationen zu Beginn einer Anforderungsanalyse sind meist informell, unvollständig, inkonsistent und teilweise sogar unrichtig (Elmasri und Navathe, 2009), ein Grossteil der Arbeit ist deshalb die Umwandlung dieses ersten Anforderungskatalogs in eine Spezifikation der Anwendung als Ausgangspunkt für den konzeptuellen Entwurf. Aufgrund des kleinen Anfangswissens über das Inkawu Vervet Projekt war das Vorgehen der Anforderungsanalyse deshalb wie folgt:

Zuerst wurden alle möglichen Dokumente gesichtet, welche von den Forschern vorgelegt wurden: das *data collection protocol* mit den zu erfassenden Daten, das Forschungskonzept mit den Forschungsfragen des Projektes, mehrere Projektberichte mit Informationen zu Hintergründen und Methodik der Forscher. Aufgrund dieser Dokumente und der jetzigen Struktur der Daten wurde ein ausführlicher Anforderungstext erstellt, mit welchem dann Rücksprache mit der Kontaktperson Dr. Erik Willems geführt wurde. Damit konnten Fehler gefunden bzw. die Richtigkeit der erfassten Anforderungen bestätigt und Ergänzungen vorgenommen werden.

Zusätzlich wurden projektspezifische Abfragen für die Datenbank (*queries*) erstellt, da die gewünschten Queries die Art von Information repräsentieren, welche die Nutzer erlangen möchten. Im Hinblick auf den Schwerpunkt dieser Arbeit wurde der Fokus vor allem auf raum-zeitlich relevante Abfragen gelegt. Als Vorlage dient die in Kapitel 3.6.2 erstellte Kategorisierung nach Yuan und McIntosh (2002). Vorschläge von meiner Seite wurden nach Rücksprache mit den Forschern verfeinert und schlussendlich ausgewählt.

## 4.2 Datenbankentwurf

Aufgrund der Anforderungsanalyse wird nachfolgend die Datenbank entworfen. Unterteilt in einen konzeptuellen und logischen Entwurf wird dabei eine mögliche Implementierung vorbereitet und der Grundstein für eine raum-zeitliche Anreicherung gelegt.



### 4.2.1 Konzeptueller Entwurf

Ein konzeptuelles Modell bietet laut Khatri *et al.* (2004) Notations- und Formalismusköglichkeiten um eine Beschreibung der realen Welt auf höchster Ebene zu erstellen - unabhängig von Implementierungsdetails.

Anhand der Resultate der Anforderungsanalyse wird deshalb als nächster Schritt ein konzeptuelles Schema entworfen, welches DBMS-unabhängig formuliert wird. Das Ziel ist, die vollständige Kenntnis der angestrebten Datenbankstruktur abzubilden, inklusive Beziehungen und Einschränkungen. Dieses Schema ist ausserdem ein gutes Kommunikationsmittel zwischen Datenbanknutzern, Designern sowie Analysten und ist verständlicher als DBMS-spezifische Datenmodelle. Von Vorteil wird deshalb ein solches Schema in Diagrammform erstellt und soll leicht interpretierbar sein. Möglichkeiten dafür sind z.B. das *entity relationship model* (ER-Modell) bzw. *Enhanced entity relation model* (EER-Modell) sowie *Universal Modeling Language* UML oder *Object Modeling Technique* OMT (Elmasri und Navathe, 2009).

Das ER bzw. EER sind konzeptuelle Modelle auf der logischen Ebene und beschreiben einen Ausschnitt der realen Welt mit Hilfe eines Diagramms und dazugehörigen Erklärungen. Die Anforderungen der Nutzer werden in ausführlich beschriebene Entitätstypen, Beziehungen und Einschränkungen umgewandelt. Das EER erweitert das Standard ER um einige zusätzliche Konzepte aus der semantischen Datenmodellierung, womit Eigenschaften und Einschränkungen einer Datenbank präziser abgebildet werden können. Dazu gehören die Konzepte der Super- und Subklassen, Klasse/Subklasse-Beziehungen, Typenvererbung, Generalisierung / Spezialisierung und Kategorien.

UML und OMT sind Objektmodellierungsmethoden, welche vor allem für den Softwareentwurf entwickelt wurden, in der Datenbankmodellierung jedoch immer häufiger genutzt werden. Gewisse Teile dieser Methoden ähneln den EER-Modellen, wobei die Terminologie jedoch oft unterschiedlich ist. Entitätstypen im ER (bzw. EER) entsprechen Klassen in UML, eine Entität entspricht einem Objekt.

Die konzeptuelle Datenbankmodellierung ermöglicht es den Designern, sich auf die wesentliche Spezifikation der Eigenschaften der Datenbank zu konzentrieren und implementierungstechnische Aspekte vorerst auszublenden. Es empfiehlt sich, den konzeptuellen Entwurf mit den Endnutzern ausgiebig zu besprechen, um sicherzustellen, dass alle Anforderungen erfüllt sind und sich keine widersprüchlichen oder falschen Anforderungen eingeschlichen haben (Elmasri und Navathe, 2009).

In dieser Arbeit wird mit dem EER gearbeitet, da gewisse Bereiche nur mit den Konzepten von Spezialisierung, Generalisierung und Kategorien reflektiert werden können und ich bereits über ein wenig Erfahrung mit dieser Lösung besitze.

### 4.2.2 Logisches Datenbankmodell

Im logischen Datenbankentwurf werden die Entitätstypen und Beziehungen aus dem ER als Vorbereitung zur physischen Datenbank in Tabellenform überführt. Dies wird meist unabhängig vom geplanten DBMS, jedoch abhängig vom angestrebten Datenmodell durchgeführt (Elmasri und Navathe, 2009). In meinem Fall wird mit dem relationalen Datenbankmodell gearbeitet. Es wird jedoch nicht das komplette ER ins logische Modell überführt,

sondern nur einige Relationen aus ausgewählten Bereichen, wo Designentscheidungen erklärt und begründet werden.

Gewöhnlich wird in diesem Schritt auch noch eine Normalisierung durchgeführt, um Redundanzen, Einfüge-, Modifikations- und Löschanomalien zu vermeiden. Bei gut ausgearbeiteten ERs ist dies meist nicht nötig und wird in dieser Arbeit aus Zeitgründen nicht erledigt.

### 4.3 Analyse der raum-zeitlichen Informationsstruktur

Der Datenbankerstellungsprozess soll an dieser Stelle nicht nach Elmasri und Navathe (2009) weitergeführt, sondern für eine Analyse der erstellten Entitäten und Relationen nach raum-zeitlichen Strukturen unterbrochen werden. Die derzeit bestehende Datenstruktur soll im Bereich der räumlichen, zeitlichen und raum-zeitlichen Komponenten durchleuchtet werden, um mögliche Schwachstellen zu detektieren. Die erstellten Relationen mit Raum- oder Zeitinformation sollen hervorgehoben und deren Eigenschaften genau analysiert werden. Das Ziel ist eine Verbesserung oder Vereinfachung des Handlings mit den räumlichen, zeitlichen sowie raum-zeitlichen Elementen und vor allem der Bewegungsdaten der *vervets* im darauffolgenden Kapitel.

### 4.4 Raum-zeitliche Anreicherung

Hier soll aufgezeigt werden, wie mit den detektierten Schwächen in Bezug auf raum-zeitliche Elemente einfacher oder besser verfahren werden kann. Ausserdem sollen mögliche raum-zeitliche Analysen vorgezeigt werden, um das Potential dieser Daten zu verdeutlichen. Anhand einiger Beispiele wird aufgezeigt, was im raum-zeitlichen Bereich alles möglich wäre. Die Beantwortung gewisser, im Zuge der Anforderungsanalyse ausgearbeiteten Queries, ist mit einem Datenmodell beruhend auf der Basisversion des ERs nämlich nur eingeschränkt möglich.

Um die Möglichkeiten für einen Einbezug von mehr raum-zeitlichen Informationen darzustellen, muss das konzeptuelle Modell mit zusätzlichen Elementen erweitert werden. Allfällige automatisierte Skripts, Trigger und SQLs sollen aufzeigen, was mit einer möglichen Implementierung ungefähr möglich wäre. Um die SQLs auch zu testen, wird eine relationale Datenbank mit einem kleinen Datensatz erstellt. Mit Visualisierungen und Screenshots soll das raum-zeitliche Potential der Beispiele verdeutlicht werden und eine einfache, jedoch praktische Lösung für das Datenmanagement von Bewegungsdaten aus der Wildnis präsentiert werden.

## Kapitel 5

# Anforderungsanalyse

Der erste Schritt innerhalb der Analyse der Anforderungen, die Festlegung der Ansprechpartner und Nutzergruppen war im Voraus klar. Erik Willems als Postdoktorand und Projektkoordinator war meine Kontaktperson für die nachfolgenden Schritte.

Durch ihn entstand Zugang zu den vorhandenen Dokumenten (Forschungskonzept für Festlegung, was mit Datenbank möglich sein soll inklusive Forschungsfragen, *data collection protocol* mit den zu enthaltenden Daten, Projektinformationen zu Methodik). Dazu erhielt ich das *Access-File*, um die Daten-Struktur (jedoch noch ohne Daten) zu studieren. Mit Hilfe von diversen Sitzungen wurden die Anforderungen immer vertiefter analysiert und damit das konzeptuelle Modell (Kapitel 6.1) iterativ verfeinert und verbessert.

### 5.1 Forschungskonzept Inkawu Vervet Monkey Project

Die Ergebnisse aus dem Studium des Forschungskonzeptes vom Inkawu Vervet Projekt sind die Forschungsfragen, welche mit der angestrebten Datenbank beantwortet werden sollen:

- 1) Kann der Grad an Konflikten zwischen Gruppen quantifiziert werden? Dazu soll mit verschiedenen Parametern die Benutzung des Raums durch die Tiere gemessen und analysiert werden. Zu diesen statistischen Werten gehört das Mass an Überschneidung der verschiedenen Aktionsradien (*proportion of home range overlap*) und der Verteidigungsindex (*defendability index*). Ausserdem interessiert die Frequenz und Natur der Interaktionen, sowie die Konsistenz von räumlichen Mustern zwischen zwei ausgesuchten Gruppen. Sind anhaltende Unterschiede (Annäherung, Vermeidung) bei gewissen Gruppen auszumachen?
- 2) Variieren die Konflikte zwischen Gruppen in Raum und Zeit? Mit Hilfe raum-zeitlicher Variationen im Verhalten der Tiere in den Konflikten können z.B. die umkämpften Ressourcen identifiziert werden (Zugang zu Nahrung, Wasser, Zufluchtsorte, Paarungsgelegenheiten). Dies ist wesentlich für die Unterscheidung von verschiedenen Konflikt-Kategorien und für die Voraussage, wer wo teilhaben wird. Es sollen gewisse Hypothesen zum geschlechterspezifischen Verhalten der Tiere in Konflikten analysiert werden.
- 3) Ist das Ausmass der Konflikte zwischen zwei Gruppen abhängig von spezifischen Charakteristiken? Dazu gehören demographische, ökologische, genetische und soziale Einflüsse, aber auch die Interaktionsgeschichte eines 'Gruppen-Paares'.
- 4) Haben Konflikte Auswirkungen auf die individuelle Fitness? Existiert ein Anreiz für Individuen in einer sozialen Einheit, um mit kollektivem Handeln (*collective action*) den

Aktionsradius zu verteidigen? Dazu werden Konfliktanalysen mit Fitnessparametern wie Nahrungsaufnahme, Geburtenrate, Überlebenszahlen oder auch physischen Massen verglichen.

- 5) Kann das Mass an Kooperation innerhalb von Gruppen das Resultat eines Konfliktes zwischen Gruppen beeinflussen? Die Antwort auf die Frage, ob Individuen den Ausgang von Konflikten durch effektive Zusammenarbeit zu ihren Gunsten verändern können, entwickelt sich bereits in den Punkten 2) und 3), wo verschiedene Kategorien von Konflikten definiert werden.

Anhand erster Tests der Forscher mit den GPS-Daten ergaben sich unterschiedliche Werte von *home range overlaps* und auch grosse Variationen in der Natur der Interaktionen zwischen Gruppen. Die Forscher sind deshalb zuversichtlich, dass sicher die Beantwortung der Forschungsfragen 1-3 anhand der Datenlage möglich sei. Für die restlichen zwei Fragen sind grosse Datenmengen und vor allem auch schwierig zu erzielende Masse, wie z.B. zu Gruppen-internen Kooperationen, nötig.

Die Wertung dieser Forschungsziele nach Priorität mit Fokus auf raum-zeitliche Informationen wird in Kapitel 5.4 angesprochen.

## 5.2 Struktur der existierenden Daten

Das *data collection protocol* (siehe Abbildung 5.1) ist als Vorbereitung für die Beobachter im Feld gedacht, damit die Methodik der Datenerfassung immer gleich bleibt. Es zeigt auch die zu speichernden Elemente, welche bereits ziemlich strukturiert sind und erraten lassen, welches Entitätstypen und Attribute werden könnten. In der dazugehörigen *Access-Datenbank* (Abbildung 5.2) konnte die Struktur noch besser studiert werden. Man musste jedoch aufpassen, um nicht einfach bereits die verwendete Struktur zu übernehmen, denn nicht alles war aus Datenbanksicht optimal gelöst.

**Between Group Encounters**  
(Form: IVP AlIOcc BGE)

**Focal group:**  
Name of group on which data were being collected (Lookup list: GroupID)

**Encounter group<sup>5</sup>:**  
Name of group encountered (Lookup list: GroupID)

**Start time:**  
Time at which individuals of the two groups were first seen to be within 50m of each other

**Start location:**  
Location where individuals of the two groups were first seen to be within 50m of each other

**At onset, focal group was:**  
Select the predominant activity of most individuals within focal group at onset

**Overall nature:**  
Select a description that best reflects overall character of BGE (Lookup list: BGE-Nature)

Abbildung 5.1: Ausschnitt vom *data collection protocol* des Inkawu Vervet Projektes, Teil des Beschriebs von Gruppenzusammenstössen (*between group encounters*).

RecordID	<input type="text" value="0"/>
UnitID	<input type="text" value="0"/>
UserName	<input type="text" value="No one"/>
TimeStamp	<input type="text" value="21.12.2011 13:48:15"/>
Focal group	<input type="text"/>
Encounter group	<input type="text"/>
Start time	<input type="text"/>
Start location	<input type="text"/>
At onset, focal group was	<input type="text"/>
Overall nature	<input type="text"/>
Group cohesion focal group	<input type="text"/>
Group cohesion encounter	<input type="text"/>
Cooperation within focal gr	<input type="text"/>
Cooperation within other g	<input type="text"/>
Individuals from different g	<input type="text"/>
N Individuals actively invol	<input type="text"/>
Age-sex class most active:	<input type="text"/>
Up to 5 most active ind. fo	Adult males
Up to 5 most active ind. ot	Sub-adult males
	Juvenile males
End Time	Adult females
End location	Sub-adult females
	Juvenile females
Clear winner of encounter:	<input type="text"/>
'Winner' Group	<input type="text"/>
Detailed description	<input type="text"/>

Abbildung 5.2: Ansicht des Formulars, wie *between group encounter* im Feld gespeichert werden; gewisse Elemente können aus einem Dropdown-Menü ausgewählt werden, andere müssen explizit eingegeben werden. Die Beobachtungen werden dann in die *Access*-Datenbank übertragen, wo die Zeilen aus der Formular-Sicht dann als Spalten fungieren und die einzelnen *records* als Zeilen dargestellt werden.

Anhand dieser und weiterer Dokumente, in denen das Projekt eingehend beschrieben wurde, erstellte ich einen zusammenfassenden Anforderungstext mit allen vorkommenden Elementen. Der vollständige Anforderungstext befindet sich im Anhang A, nachfolgend ist ein kleiner Ausschnitt vorgelegt, wo auch schon die Einschränkungen möglicher Eingabewerte vermerkt sind:

*If a Between Group Encounter (BGE) occurs, the observed as well as the encountered group is saved, start/end time, predominant activity at start (either moving, resting, feeding or social; no details) and nature of BGE (affiliative, aggressive, neutral, ignore), estimated group cohesion (10m, 20, 50, 100, 100+) of both groups and cooperation formed within both groups (yes/no). The general behaviour of individuals from different groups (multi-selection: feed in same patch, allogroom, play, engage in sexual behaviour, scream, displace, chase, hit, bite) is chosen. The number of actively involved individuals (N), their AgeSexClass (adult, sub-adult, juvenile males/females) and if possible the 5 most active monkeys of both groups are noted. Was there a clear winner group (yes/no), which one? A detailed description is noted using ethogram stenography. Start/end location is provided separately through a handheld GPS by the person who notes the BGE.*

Der komplette Anforderungstext diene als Grundlage, um mit Dr. Willems Rücksprache zu halten und die für die Datenbank bestimmten Elemente auszumachen. In den Gesprächen (siehe nächstes Kapitel) wurden die einzelnen Bereiche vertieft diskutiert und Prioritäten ausgemacht.

GPS-Rohdaten werden etwa einmal im Monat getrennt nach den Halsbändern im Feld heruntergeladen. Es werden dann Files mit der ID des GPS-Empfängers (z.B. 1296) kreiert, ein Ausschnitt daraus sieht dann ungefähr so aus:

```
ACC,1396,29.12.2010,We,11:58:00,2064,1593,1780,2096,1541,1792,...
```

```
GPS,1396,29.12.2010,We,12:00:01,31.1842245,-28.0073448,675.6,3,A,1,
137,12,29.12.2010,We,12:00:50,3662,18,0.600000,77.987840,6.160000,13.824000
```

```
ACC,1396,29.12.2010,We,12:02:00,1909,1626,1654,1912,1656,1642,1922,...
```

Da Accelerometer-Messungen alle zwei und GPS-*fixes* alle 30 Minuten gemacht werden, hat es vielmehr ACC-Daten, welche zudem noch viel umfassender sind. Da im Projekt die Accelerometer-Daten im Moment nicht gebraucht werden, sind die Werte partiell ausgeblendet. In einem ersten Schritt müssen diese ACC- von den GPS-Messungen getrennt werden, was mit einem entsprechenden Programm bewerkstelligt wird. Die GPS-Daten werden nachfolgend mit einer spezifischen Software in eine *Access* Datenbank importiert, wobei die von Kommas separierten Werte pro *fix* als folgende Attribute gespeichert werden:

Art der Daten (in unserem Falle GPS)

Anzahl Tage seit dem 04. März 2007

Datum im Forschungsgebiet (muss festgelegt werden)

Wochentag im Forschungsgebiet

Zeit im Forschungsgebiet

Geographische Länge (*Longitude*)

Geographische Breite (*Latitude*)

Höhe über Ellipsoid [m]

Art des *fixes*: 3 = 3D *fix*, 2 = 2D *fix*

Status: A = Position und Zeit innerhalb der Genauigkeitsmaske, B = nur Zeit und

Wochennummer gültig, C = nur Wochennummer gültig

Identifizierungssignal der GPS-Uhr (*GPSFlag*)

GPS *noiselevel*

TTFF (*Time To First Fix*), gebrauchte Anzahl Sekunden zum ersten *fix*

Datum des *fixes* (nach GPS-Zeit)

Wochentag (nach GPS-Zeit)

GPS-Zeit (entspricht ungefähr der UTC/Greenwich-Zeit)

Batteriespannung

Temperatur [°C] (nicht kalibriert und deshalb ziemlich ungenau)

Geschwindigkeit [m/s]

Richtung (*heading*) [Grad]

Genauigkeit der Geschwindigkeit [m/s]

Horizontale Genauigkeit des *fixes* [m]

Die einzelnen *fixes* sind also nicht nummeriert, sondern werden anhand des Datums und der Zeit auseinandergehalten. Zudem werden zwei Zeiten geliefert, in weiteren Schritten muss also keine Umrechnung in andere Zeitzonen mehr gemacht werden. Je nach Managementsystem der Daten ist jedoch ein Umrechnen in andere Formate notwendig, da im rohen GPS-File das Datum und die Zeit getrennt vorliegen. Zur einfacheren Handhabung und Analyse der Daten bietet sich ein Format wie `'dd/mm/yyyy hh:mm:ss'` an. In vielen DBMS können Datentypen wie *date* oder ähnliche Alternativen verwendet werden, was den Gebrauch von Zeiten grundsätzlich erleichtern sollte, da damit Berechnungen möglich sind.

### 5.3 Sitzungen mit Kontaktperson

In den fünf Sitzungen mit Erik Willems, dem Projektkoordinator, wurde neben den einzelnen Elementen für die Datenbank auch vor allem das Vorgehen der Forscher im Feld und die Prioritäten betreffend Forschungsfragen (aus Kapitel 5.1) besprochen.

Das methodisches Vorgehen der Forscher im Feld an dieser Stelle umfassend zu beschreiben, wäre zu umfangreich und nicht produktiv. Ein Teil davon ist im Anforderungstext im Anhang enthalten, auf die wichtigsten Punkte möchte ich aber trotzdem kurz eingehen:

Ungefähr alle 4-5 Tage geht eine kleine Gruppe von PhD- und MSc-Studenten (meist 2 Personen) ins Feld und sucht die zugeteilte Gruppe von *Vervet Monkeys* anhand von VHF-Transmittern. Immer zur vollen und halben Stunde wird ein sogenannter Gruppen-Scan durchgeführt, wobei für eine Dauer von 10 Minuten immer ein Individuum für 1 Minute beobachtet wird. Auf einem PDA werden die Forscher durch eine Serie von *forms* geführt und können dort die gemachte Beobachtungen festhalten. Beim SCAN sind eher gruppenspezifische Elemente gefragt und durch die Auswahl der 10 Affen soll auch die Gesamtsituation der Gruppe repräsentiert werden.

Zwischen den SCANS werden FOCAL-Beobachtungen gemacht, d.h. ein einzelnes Individuum wird für 10 Minuten verfolgt und jede Minute wird dessen Verhalten notiert. Dabei sollen für ein Individuum für jede der vier vordefinierten Zeitperioden nicht nacheinander folgende Beobachtungen vorliegen. Das bedeutet, dass ein einzelnes Tier an mehreren Tagen überwacht werden muss.

Nicht zu steuern sind die Ereignisse, welche über den ganzen Tag verteilt plötzlich auftreten können und sogleich festgehalten werden sollen (*OnOccurrence*). Dazu gehören Alarmrufe, Paarungsverhalten, Konkurrenzkämpfe und Zusammenstöße mit anderen Gruppen. Das vorgefertigte Formular auf dem PDA muss ausgewählt und dementsprechend ausgefüllt werden.

Einmal täglich wird der sogenannte Log ausgefüllt, eine Art Logbuch zum Festhalten von grundlegenden Informationen. Dazu gehören z.B. die Namen der Beobachter oder das Auto. Wichtiger sind jedoch die festgehaltenen Änderungen bezüglich der Gruppenkomposition, welche auch an dieser Stelle gespeichert werden und nur einmal pro Tag registriert werden.

Die GPS-Empfänger für die regelmässigen Messungen werden jeweils dem dominanten Weibchen angelegt, da diese die Gruppen nie wechseln. Ein Emigrieren in eine andere Gruppe würde einem Selbstmord gleichkommen. Somit stehen für jede Gruppen regelmässige GPS-Messungen zur Verfügung, welche alle Monate durch die Forscher im Feld heruntergeladen werden. Dies funktioniert bis auf eine Distanz von ungefähr 500m.

Generell wurde aus den Gesprächen auch klar, dass dem raum-zeitlichen Bereich bei der Modellierung des jetzigen Systems wenig Bedeutung zukam. Und dies obwohl die Forschungsfragen (in Kapitel 5.1) fast alle einen räumlichen Bezug besitzen. Ich bekam einige räumliche Analysen zu sehen, in welchen die Bewegungen der Gruppen visualisiert waren und wie aus den Rohdaten Gruppenkonflikte mit einem Java-Skript detektiert wurden. Im Gespräch kam zum Ausdruck, dass das Abfragen der Kontextinformation an den festgestellten *between group encounters* technisch nicht möglich sei. Gerade um Rückschlüsse auf die umkämpften Ressourcen zu ziehen und verschiedene Kategorien von Zusammenstößen (wie z.B. Konflikte um Nahrung oder Sexualpartner) auseinanderzuhalten, ist das Wissen um die Kontextinformation unerlässlich. Das Inkawu Vervet Projekt besitzt grundsätzlich verschiedene Kontextinformationen (Vegetationskarte, *food availability*- sowie *habitat productivity*-Raster), doch im benutzten System sind diese nicht miteinander vernetzt und können nur beschränkt gemeinsam untersucht werden.

## 5.4 Abfragen

Die in Kapitel 3.6.2 vorgenommene Klassifizierung nach Attribut-, räumlichen, zeitlichen und raum-zeitlichen Queries wird hier auf projektspezifische Beispiele angewendet. Die Query-Beispiele sollen als Diskussionsgrundlage für die gewünschten bzw. benötigten raum-zeitlichen Informationen in einem zukünftigen System dienen. Als Teil der Anforderungsanalyse wurden die Beispiele vor der Erstellung des konzeptuellen Modells ausgearbeitet und repräsentieren damit nicht unbedingt die später ausgearbeitete Situation der Datenstruktur. Denn je nach Modellierung oder sogar je nach Speicherungs- und Abfragezeitpunkt kann die Zuweisung der Queries variieren, was in der Einteilung hier nicht berücksichtigt wurde. Es ging auch vor allem darum, anhand dieser Abfragen die Anforderungen und Wünsche der Forscher im raum-zeitlichen Bereich zu identifizieren.

**Attribut Abfragen** Wann wurde Affe B geboren? Welche Gruppe wurde am längsten untersucht? Wie viele Affen besitzt Gruppe 3 jetzt?

**Einfache räumliche Abfrage** Wo ist der Aussenposten XY der Forschungsstation? Wo gibt es Savanne im Forschungsgebiet?

**Räumliche Abgrenzungsabfrage** Welche Gruppen befanden sich mindestens einmal in der Nähe (100m) des Flusses X? In welcher Art Vegetation ist die Dichte der Vervet Monkeys am grössten?

**Räumliche Beziehungsabfrage** Welche Gruppe ist am nächsten zu Gruppe 3? Welche Art Vegetation grenzt ans Gewässer?

**Einfache zeitliche Abfrage** Wann war die grösste Dürreperiode? Wie viele Affen zählte Gruppe 1 am 23.04.2010?

**Zeitliche Abgrenzungsabfrage** Mit welchen Durchschnittsgeschwindigkeiten sind die Tiere während dem Tag unterwegs? In welchem Monat gibt es am meisten Geburten / Tode?



**Zeitliche Beziehungsabfrage** Korrelieren Ruhezeiten der Tiere mit extremen Wetterereignissen? Gibt es weniger / mehr Konflikte während der Paarungszeit?

**Einfache raum-zeitliche Abfrage** Wann war Affe A / Gruppe 1 zuletzt an Position XY (z.B. Wasserstelle)? Wo war Gruppe 3 am 1. Mai 2011?

**Raum-zeitliche Abgrenzungsabfrage** Welche Gruppe reist über die grössten Distanzen im März 2010? Wie ändert sich die Reviergrösse (*home range*) der Gruppe 1 vom Sommer zum Winter?

**Raum-zeitliche Verhaltensabfrage** Sind die Bewegungen der Gruppen 1 und 2 unabhängig voneinander oder gibt es Ähnlichkeiten (folgt eine Gruppe der anderen, mit / ohne Verzögerung oder meiden sie sich)? Wie viele Konflikte gibt es zwischen den Gruppen 3 und 6? Gibt es zwischen genetisch nahen Gruppen weniger Konflikte? Haben sich die Lebensräume in den letzten 2 Jahren verändert? Wo und um wie viel überschneiden sich die Lebensräume? Gibt es sich wiederholende Bewegungsmuster, welche Rückschlüsse auf Verhaltensweisen zulassen?

**Raum-zeitliche Beziehungsabfrage** Wie passen sich die Gruppen den Lebensräumen der Raubtiere an? Sind saisonale, wie z.B. nahrungstechnisch bedingte, Variationen in den Routen erkennbar, welche durch die variierend vorherrschende Vegetation erklärbar sind? In welchem Kontext (z.B. Vegetationstyp) gibt es die meisten Konflikte?

Wie im vierten Vorgehenspunkt der Anforderungsanalyse erwähnt, wurden im Hinblick auf eine Anreicherung der Möglichkeiten mit raum-zeitlichen Informationen die Prioritäten des Forschungsteams bezüglich Raum-Zeit-Bereich ausfindig gemacht. Im Gespräch wurden die Gebiete festgelegt, auf welche der Schwerpunkt im konzeptuellen Modell sowie dem angestrebten Datenbank-Prototypen gelegt werden soll:

Anhand der regelmässigen GPS-Daten sollen die Zusammenstösse von Gruppen automatisch ausfindig gemacht werden, um die *between group encounters* zu quantifizieren. Dies entspricht ungefähr der Forschungsfrage 1 des Inkawu Vervet Projektes aus Kapitel 5.1. Damit sollen statistisch repräsentative Zahlen zur Frequenz von Gruppenkonflikten erzielt werden, da die Forscher nicht rund um die Uhr bei allen Gruppen sein können. Die Erfassung 'aller' Zusammenstösse bereitet die weiteren Forschungsfragen (v.a. 2 und 3) vor, da anhand dieser Basisdaten weitere Analysen gemacht werden können.

Der zweite Schwerpunkt ist die Anreicherung von Ereignissen mit Kontextinformation. Es soll möglich sein, jeden Punkt mit seiner zugehörigen Umwelt in Bezug zu setzen. Die Gruppenkonflikte sollen mit der Vegetationskarte oder auch zeitlich variierender Informationen zu Nahrungsverfügbarkeit verbunden werden, um Rückschlüsse auf die umkämpften Ressourcen machen zu können (Forschungsfrage 2 des Projektes). Zudem soll dieser Bezug auch für die anderen räumlich sowie zeitlich lokalisierten Ereignisse wie Alarmrufe, Paarungs- und Konkurrenzverhalten ermöglicht werden.



## Kapitel 6

# Datenbankentwurf

### 6.1 Konzeptueller Entwurf

In diesem Kapitel wird die Wahl der Entitäten, Relationen, Attribute und Einschränkungen erklärt, indem einige Beispiele ausgesucht werden. Dies geschieht anhand einzelner Ausschnitte des *entity relationship models*, das vollständige ER befindet sich im Anhang B.

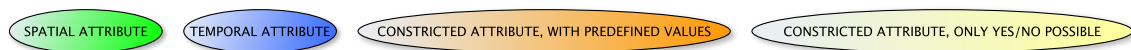


Abbildung 6.1: Farbcode der Entitäten und Relationen im *entity relationship model*.

Von nun an werden im Text zur Vereinfachung Namen von Entitäten und Relationen in serifenloser Schrift (**PERSON**) und Attribute als Kapitälchen (**NAME**) geschrieben. Die Sprache der Elemente ist Englisch, da das internationale Forscherteam vorwiegend damit kommuniziert. Um das Lesen des ERs zu vereinfachen, wurden die Attribute verschieden eingefärbt (Abbildung 6.1): grün sind räumliche Attribute, blau bedeutet zeitliches Attribut und orange wie gelb sind Attribute mit eingeschränktem Wertebereich, wobei letztere nur ja/nein sein können. In Abbildung 6.2 z.B. kann in Attribut **CHANGE\_KIND** nur zwischen den Werten 'immigration, emigration, disappearance, appearance' ausgewählt werden. In einer Datenbank würde unter Umständen ein weiterer Entitätstyp erstellt, als visuell einfachere Variante wurde es jedoch im ER als Attribut belassen.

Die Grundstruktur des ERs besteht aus den Entitätstypen **MONKEY** sowie **MONKEY\_GROUP**, welche mit den verschiedenen Beobachtungen bzw. Ereignissen verknüpft werden können. Es befinden sich mehrere *personal digital assistants* (PDAs) als auch Personen im Einsatz. Um die Eindeutigkeit der *records* von Beobachtungen zu gewährleisten, werden die Formulare für Beobachtungen (**LOG**, **SEXUAL\_BEHAVIOUR**, **BETWEEN\_GROUP\_ENCOUNTER**, **WITHIN\_GROUP\_COMPETITION**, **ALARM\_CALL**, **PHYSICAL\_MEASUREMENT** sowie **REGULAR\_OBSERVATION**) als schwache Entitätstypen von PDA modelliert. Dadurch wird sichergestellt, dass der Primärschlüssel (*primary key*, PK) als Kombination der automatisch vergebenen fortlaufenden Beobachtungs-ID des jeweiligen PDAs und der zugehörigen **PDA\_ID** einzigartig ist. Im Inkawu Vervet Projekt wird für den Import aller *forms* jeweils ein Programm namens *Pendragon*<sup>1</sup> verwendet, welches automa-

<sup>1</sup><http://pendragonsoftware.com/> 14.01.2012

tisch eindeutige Primärschlüssel vergibt. In meiner Modellierung wird dies nicht beachtet, um eine möglichst system-unabhängige Lösung zu erstellen.

Die räumlichen Daten sind zum jetzigen Zeitpunkt von den anderen Elementen getrennt, und werden in der Basisversion des ERs auch so dargestellt. Bei allen *OnOccurrence*-Beobachtungen, wo mit Hilfe eines separaten *handheld*-GPS-Gerätes eine Ortung vorgenommen wird, repräsentiert ein gestricheltes räumliches Attribut eine vorhandene Position. Diese räumlichen Daten müssen manuell zugewiesen werden und sind nicht über Beziehungen verbunden. Im weiteren Verlauf der Arbeit werden alternative Strukturen für das Handling von Bewegungsdaten präsentiert und diskutiert.

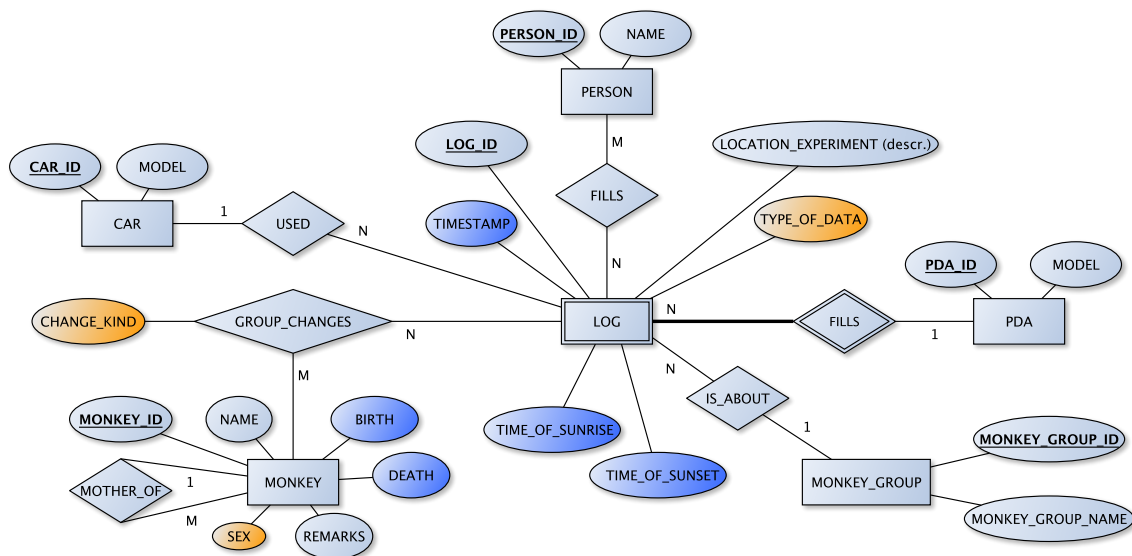


Abbildung 6.2: ER-Struktur für LOG und wie Änderungen der Gruppenkomposition gespeichert werden könnten (Tabelle GROUP\_CHANGES).

Im LOG (Abbildung 6.2) werden die Aktivitäten einer Gruppe von Forschern (meist zwei Personen) an einem Tag vermerkt. Da mehrere PDAs im Einsatz sind, wird LOG wegen oben genanntem Grund als schwacher Entitätstyp von PDA modelliert. Weiter wird bei einem neuen Log automatisch ein Zeitstempel (TIMESTAMP) beigefügt, bei welchem vor allem das Datum entscheidend und die Zeit eher unwichtig ist. Die beobachtete Gruppe (Beziehung zu MONKEY\_GROUP), Arten von aufgenommenen Daten sowie weitere Daten werden protokolliert.

Eine Änderung in der Gruppenkombination wird auch jeweils einmal täglich innerhalb des Logs vermerkt, wobei das Individuum und die Art Änderung (*disappearance*, *emigration*, usw.) gespeichert werden. Da ein einzelnes Tier nur eine Änderung pro Tag und damit pro Log durchlaufen kann, schlage ich einen N:M-Beziehungstyp GROUP\_CHANGES zwischen LOG und MONKEY vor (Abbildung 6.2). Der Primärschlüssel ist folglich eine Kombination von LOG\_ID, PDA\_ID und MONKEY\_ID und jede Veränderung in der Gruppenzusammensetzung kann somit registriert werden (siehe auch logisches Modell im nächsten Kapitel und in Abbildung 6.5). Über die Beziehung zum Log ist für jede Änderung ein Zeitstempel und die Gruppe bekannt, was für zukünftige Analysen interessant ist (siehe auch *view* GROUP\_CHANGES\_VIEW in Abbildung 6.6). Dass der Zeitpunkt der Veränderung (*valid time*) eine Ungenauigkeit von bis zu 5 Tagen aufweisen kann, je nachdem wann

die Forscher die Gruppe besuchten, ist nicht weiter schlimm, da für Analysen zur demographischen Gruppenzusammensetzung grössere Zeiträume entscheidend sind. Um bereits vorhandene Daten zu Gruppenkomposition hier einzubinden, könnte neben den Arten von Veränderungen in *CHANGE\_KIND* (*imigration, appearance*, usw.) ein neuer Wert eingeführt werden, z.B. 'is\_in'. Spätere Veränderungen werden dann wie besprochen aufgenommen. Das Geburts- und Todesdatum eines Affen wird ebenfalls im Log verzeichnet und während dem Import der Daten in die Datenbank dem jeweiligen Individuum zugewiesen.

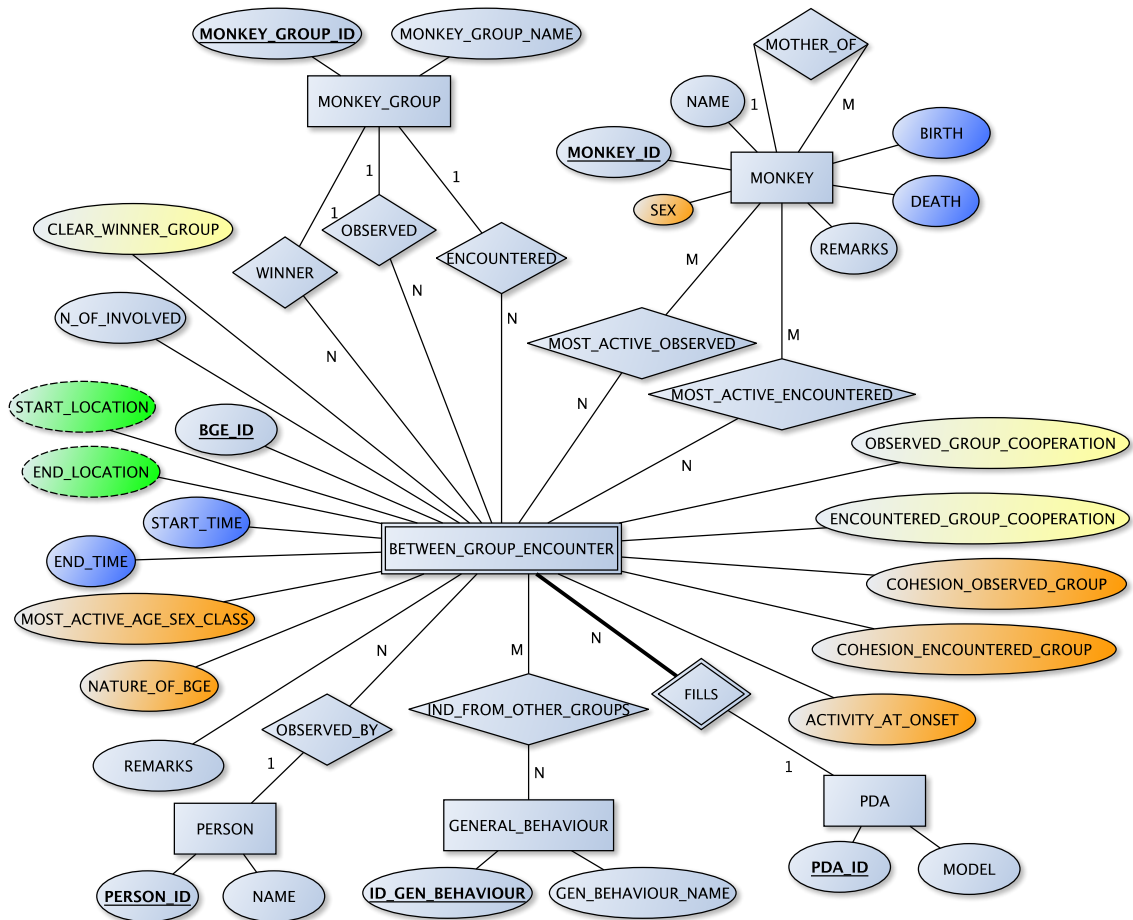


Abbildung 6.3: ER-Struktur der Gruppenzusammenstösse (*BETWEEN\_GROUP\_ENCOUNTER*).

Falls Forscher anwesend sind, werden Konflikte zwischen Gruppen bei ihrem Auftreten protokolliert, teilweise jedoch auch gefilmt und die Details im Nachhinein vermerkt. Der Entitätstyp *BETWEEN\_GROUP\_ENCOUNTER* (siehe Abbildung 6.3) stellt dabei als schwacher Entitätstyp von PDA das Zentrum dar, mit diversen Attributen sowie Beziehungen zu anderen Entitätstypen. Die teilnehmenden und eine allfällig gewinnende Gruppe sowie die fünf aktivsten Individuen pro Gruppe werden über Beziehungen zu *MONKEY\_GROUP* respektive *MONKEY* modelliert. Start- und Endzeiten sowie die Positionen beim Beginn und Ende eines Zusammenstosses werden als zeitliche respektive räumliche Attribute modelliert. Es gilt jedoch zu beachten, dass die Positionsangaben mit einem separaten *handheld*-GPS erfasst und damit auch getrennt gespeichert werden. Dies wird im ER ver-

deutlich, indem die räumlichen Attribute mit gestricheltem Rand dargestellt sind. Es soll jedoch gezeigt werden, dass die Daten trotz nicht vorhandener direkter Verbindung für eine Positionierung vorhanden sind. Damit zu einem Gruppenzusammenstoß mehrere generelle Verhaltensweisen von Individuen der beiden Gruppen gewählt werden können, wird eine N:M-Beziehung (IND\_FROM\_OTHER\_GROUPS) zu der Auswahl GENERAL\_BEHAVIOUR gebildet.

Bei der Anforderungsanalyse fiel auf, dass für die regelmässigen Beobachtungen (FOCAL und SCAN) viele gemeinsame Attribute und Beziehungen bestanden (siehe vollständiges ER im Anhang B). Um dies vereinfacht im *entity relationship model* darzustellen, wurde vom *enhanced ER* das Prinzip der Generalisierung angewandt. Mit dem Entitätstyp **REGULAR\_OBSERVATION** wurden die beiden verschiedenen Beobachtungsformen zusammengefasst. An dieser sogenannten Superklasse wurden die gemeinsamen Attribute und Beziehungen angehängt, während individuelle Attribute den Subklassen **FOCAL** und **SCAN** beigelegt wurden. Da nur entweder die eine oder andere Beobachtungsform auftreten kann, trifft die *Disjunktheit*-Einschränkung zu (Elmasri und Navathe, 2009).

Dasselbe Prinzip wird beim Entitätstyp **ACTIVITY** angewendet: Bei der Wahl der Aktivität eines Tieres muss zwischen **FEEDING**, **MOVING**, **RESTING** und **SOCIAL** selektiert werden, was wieder eine Disjunkteinschränkung mit sich zieht, da die Entität höchstens in einer Subklasse Mitglied sein kann. Die Subklasse **SOCIAL** wird weiter in **AFFILIATIVE** oder **AGGRESSIVE** unterteilt, wo je verschiedene Verhaltensmuster vermerkt sind. Weil die Subklassen **AFFILIATIVE** und **AGGRESSIVE** die gleiche Superklasse (**REGULAR\_OBSERVATION**) besitzen, spricht man hier von einer Hierarchie (Elmasri und Navathe, 2009).

Eine Generalisierung von **SEXUAL\_BEHAVIOUR**, **BETWEEN\_GROUP\_ENCOUNTER**, **WITHIN\_GROUP\_COMPETITION** und **ALARM\_CALL** unter **ON\_OCCURRENCE\_OBSERVATION** wurde verworfen, da nicht sehr viele gemeinsame Attribute und Beziehungen vorhanden sind. Immerhin die Beziehung zu **PDA** und **PERSON** sowie die Attribute **TIME-STAMP**, **HANDHELD\_GPS\_FIX**, **REMARKS** und **ID** wären gleich gewesen.

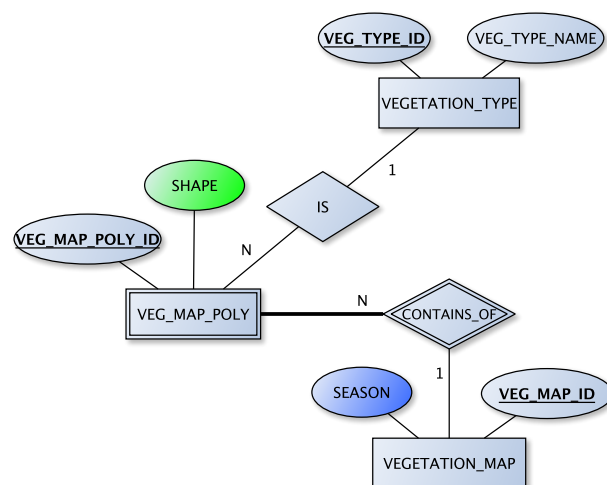


Abbildung 6.4: Mögliche ER-Struktur der Vegetationskarte, wobei die Speicherung und Abfrage verschiedener Jahreszeiten zu verschiedenen Jahren möglich wäre.

Die räumlichen Information liegen im *Inkawu Vervet Project* getrennt von den restlichen Daten vor und sind deshalb auch im ER vom Rest separiert (siehe Anhang B). Im Moment liegt die Information zur Vegetation noch als statische Karte ohne zeitliche Variation vor. Mit dem Entitätstyp **VEGETATION\_MAP** wird jedoch bereits die Möglichkeit gegeben, mehrere verschiedene Vegetationskarten zu speichern und abzufragen. Abbildung 6.4 zeigt die Modellierung mit einzelnen Polygonen (**VEG\_MAP\_POLY**) verschiedenen Vegetationstyps (**VEGETATION\_TYPE**), aus welchen eine **VEGETATION\_MAP** zusammengesetzt sein kann. Damit besteht die Möglichkeit, einerseits unterschiedliche Jahreszeiten, andererseits auch verschiedene Jahre abzuspeichern.

## 6.2 Logisches Datenmodell

Um gewisse Bereiche des ERs zu erklären, wird für einige ausgewählte Entitäten und Relationen das logische Datenmodell erstellt. Dabei werden Tabellenstrukturen anhand von vordefinierten Überführungsregeln nach Elmasri und Navathe (2009) abgeleitet. In den abgebildeten Tabellen repräsentiert dabei der Titel den Namen der Relation, die Spalten sind Attribute oder Fremdschlüssel (*foreign key* FK) zu anderen Tabellen. Dunkelgrau unterteilt ist jeweils der Primärschlüssel einer Relation und kann auch aus mehreren Spalten bestehen.

MONKEY						PDA			MONKEY_GROUP	
MONKEY_ID	NAME	SEX	BIRTH	DEATH	...	PDA_ID	MODEL_NR	REMARKS	MONKEY_GROUP_ID	NAME
2	AF001	FEMALE	12.04.2009	NULL	...	345	XCD3044	NULL	1	ANKASSE
1	AM001	MALE	23.04.2008	NULL	...	132	XCD3044	NULL	2	LEMON TREE
4	AM002	MALE	02.05.2010	NULL	...	368	XCD3044.DF	NULL	4	ROCK

PERSON			LOG					
PERSON_ID	NAME	...	LOG_ID	PDA_ID (FK)	TIMESTAMP	MONKEY_GROUP_ID (FK)	PERSON_ID (FK)	...
14	JACK	...	34	345	12.04.2009 - 16:48:34	1	43	...
29	ERIK	...	35	345	13.04.2009 - 17:03:23	2	43	...
43	TINA	...	45	132	02.05.2009 - 11:17:48	1	14	...

GROUP_CHANGES			
LOG_ID (FK)	PDA_ID (FK)	MONKEY_ID (FK)	CHANGE_KIND
34	345	7	EMIGRATION
34	345	1	DISAPPEARANCE
45	132	4	IMMIGRATION

Abbildung 6.5: Logisches Modell um LOG und GROUP\_CHANGES mit einigen fiktiven Datenwerten.

Abbildung 6.5 zeigt die im vorherigen Kapitel abgebildete Modellierung rund um den LOG und GROUP\_CHANGE (Abbildung 6.2). Die Tabellen MONKEY, PDA, MONKEY\_GROUP und PERSON besitzen nur einen Fremdschlüssel, während ein Eintrag (*record*) im schwachen Entitätstyp LOG durch die Verbindung von dessen eigener (LOG\_ID) sowie der PDA\_ID einzigartig wird. Wendet man die Überführungsregeln auf den N:M-Beziehungstyp GROUP\_CHANGES an, so ergibt sich der Primärschlüssel aus drei Fremdschlüsseln (LOG\_ID, PDA\_ID, MONKEY\_ID). Dies macht Sinn, denn eine einmalige Änderung in der Gruppenkomposition besteht aus einer Verbindung von LOG und MONKEY. Ein Log behandelt

immer eine einzelne Gruppe und darin kann ein Individuum nicht am selben Tag verschwunden oder aufgetaucht sein.

Beim Ausfüllen des LOGs wird automatisch eine fortlaufende ID, die ID des aktuellen PDAs sowie ein Zeitstempel protokolliert und der Beobachter wählt seinen Namen (bzw. seine PERSON\_ID über den Fremdschlüssel) aus. Die beobachtende Gruppe wird ebenfalls über eine Fremdschlüsselbeziehung vermerkt. Es können pro Log mehrere Änderung in der Gruppe festgehalten werden, in Abbildung 6.5 kann man erkennen, dass für den Log mit der LOG\_ID 34 und PDA\_ID 345 zwei verschiedene Einträge vorhanden sind. Einerseits ist Affe Nr. 7 ausgewandert und andererseits das Individuum Nr. 1 verschwunden.

Die Tabelle GROUP\_CHANGES allein gibt jedoch nicht sehr viel her, erst über die Beziehungen entstehen Informationen. Kombiniert man die Relation nun mit einigen Attribute aus den verbundenen Tabellen, so kann man eine Datenbankansicht (*view*) zu den Gruppenänderungen erstellen (GROUP\_CHANGES\_VIEW in Abbildung 6.6). Damit soll bereits der physische Datenbankentwurf beziehungsweise die Analyse ein wenig vorgezogen werden. Mit einem simplen SQL können darin nun alle Einträge zu einer bestimmten Gruppe (hier Gruppe 1) abgefragt werden:

```
SELECT g.monkey_id, g.change_kind, g.timestamp
FROM GROUP_CHANGES_VIEW g
WHERE g.group_id='1'
```

SQL 6.1: Auflistung aller Veränderungen einer ausgewählten Gruppe aus GROUP\_CHANGES\_VIEW.

Es könnten auch die Veränderungen aller Gruppen innerhalb eines gewissen Zeitraums (wie hier das Beispiel mit dem ganzen Monat Mai im Jahr 2009) interessieren:

```
SELECT g.monkey_id, g.group_id, g.change_kind, g.timestamp
FROM GROUP_CHANGES_VIEW g
WHERE g.timestamp BETWEEN to_date('2009/05/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss')
AND to_date('2010/05/31 23:59:59', 'yyyy/mm/dd hh24:mi:ss')
```

SQL 6.2: Abfrage zu den Veränderung aller Gruppen innerhalb einer bestimmten Zeitperiode aus GROUP\_CHANGES\_VIEW.

GROUP\_CHANGES\_VIEW

LOG_ID (FK)	PDA_ID (FK)	MONKEY_ID (FK)	CHANGE_KIND	TIMESTAMP (FK)	GROUP_ID (FK)	PERSON_ID (FK)	...
34	345	7	EMIGRATION	12.04.2009 - 16:48:34	1	43	...
34	345	1	DISAPPEARANCE	12.04.2009 - 16:48:34	1	43	...
45	132	4	IMMIGRATION	02.05.2009 - 11:17:48	1	14	...

Abbildung 6.6: Mögliche *View* von GROUP\_CHANGES, mit ausgewählten Attributen bzw. Beziehungen.

Natürlich können noch weitere Attribute beigezogen werden, die Datenbankansicht aus Abbildung 6.6 ist nur ein Beispiel um die Funktionalität rund um die Relationen LOG und GROUP\_CHANGES zu demonstrieren. Theoretisch wäre auch ein SQL möglich, welcher die Anzahl Individuen der Gruppen automatisch für jeden Zeitpunkt herauslesen kann, indem 'emigration' beispielsweise als -1 und 'immigration' als +1 zählt. Dazu ist aber fundiertes SQL-Wissen nötig.



Zur Vervollständigung ist in Abbildung 6.7 noch das logische Modell für die Relationen rund um die Beobachtungen von Gruppenzusammenstößen (**BETWEEN\_GROUP\_ENCOUNTER**) dargestellt. Auch wenn die Relationen erstmals erschreckend chaotisch erscheinen, sollte damit das Verständnis des in Abbildung 6.3 gezeigten ERs besser werden. Denn mit dieser Struktur werden die Daten möglichst redundantfrei gespeichert und es wird durch zahlreiche Primär- und Fremdschlüsselbeziehungen sichergestellt, dass nur in Frage kommende Werte eingefügt werden können. Beispielsweise können nur vorhandene GruppenIDs als **OBSERVED**, **ENCOUNTERED** oder **WINNER** eingefügt werden. Da **BETWEEN\_GROUP\_ENCOUNTER** ein schwacher Entitätstyp ist, wird der Primärschlüssel aus **PDA\_ID** sowie **BGE\_ID** zusammengesetzt. Weil mehrere generelle Verhaltensweisen aufgezählt werden können, wird die N:M-Beziehung aus dem ER in eine eigene Relation **IND\_FROM\_OTHER\_GROUPS** überführt, mit insgesamt drei Fremdschlüsseln aus den teilnehmenden Tabellen. Dasselbe passiert mit der Aufzählung der fünf aktivsten Tiere, eine eindeutige Entität wird über drei PKs definiert.

Als Beispiel soll der *encounter* vom 17.04.2009 mit **PDA\_ID** 345 und **BGE\_ID** 56 untersucht werden: Während Gruppe 1 von den Forschern begleitet und untersucht wurde, traf man um 16:48 auf Gruppe 4. Das generelle Verhalten der Individuen war 'play' und die Natur des Zusammenstoßes 'neutral'. Die aktivsten Individuen der beobachteten Gruppe waren 8 und 14, von der angetroffenen Gemeinschaft 3 und 5. Nach rund 20 Minuten trennten sich die beiden Gruppen wieder, ohne dass jemand 'gewonnen' hatte. Wenn man die Abbildung 6.3 untersucht, so erkennt man noch viel mehr Attribute, welche für einen einzelnen Konflikt aufgenommen werden. Aus Platzgründen wurde hier im logischen Modell jedoch darauf verzichtet, beim Erstellen der physischen Datenbank müssten natürlich alle Attribute erstellt werden.

MONKEY

MONKEY_ID	NAME	SEX	BIRTH	DEATH	...
2	AF001	FEMALE	12.04.2009	NULL	...
1	AM001	MALE	23.04.2008	NULL	...
4	AM002	MALE	02.05.2010	NULL	...

PDA

PDA_ID	MODEL_NR	REMARKS
345	XCD3044	NULL
132	XCD3044	NULL
368	XCD3044.DF	NULL

MONKEY\_GROUP

MONKEY_GROUP_ID	NAME
1	ANKASSE
2	LEMON TREE
4	ROCK

PERSON

PERSON_ID	NAME	...
14	JACK	...
29	ERIK	...
43	TINA	...

GENERAL\_BEHAVIOUR

G_B_ID	G_B_NAME
1	PLAY
2	CHASE
3	HIT

MOST\_ACTIVE\_OBSERVED

PDA_ID (FK)	BGE_ID (FK)	MONKEY_ID (FK)
345	56	8
345	56	14
132	26	45

IND\_FROM\_OTHER\_GROUPS

PDA_ID (FK)	BGE_ID (FK)	G_B_ID (FK)
345	56	1
345	57	5
132	48	4

MOST\_ACTIVE\_ENCOUNTERED

PDA_ID (FK)	BGE_ID (FK)	MONKEY_ID (FK)
345	56	3
345	56	5
132	26	34

BETWEEN\_GROUP\_ENCOUNTER

PDA_ID (FK)	BGE_ID	OBSERVED (FK)	ENCOUNTERED (FK)	WINNER (FK)	START_TIME	END_TIME	NATURE_OF_BGE	...
345	56	1	4	NULL	17.04.2009 - 16:48:34	17.04.2009 - 17:08:14	NEUTRAL	...
345	57	5	3	NULL	28.04.2009 - 11:18:12	28.04.2009 - 12:11:56	IGNORE	...
132	26	3	2	3	02.05.2009 - 11:17:48	02.05.2009 - 13:34:23	AGGRESSIVE	...

Abbildung 6.7: Logisches Modell der Relationen rund um **BETWEEN\_GROUP\_ENCOUNTER** mit einigen fiktiven Datenwerten.

Die ER-Modellierung der Vegetationskarte mit unterschiedlichen Zeitständen aus Abbildung 6.4 entspricht dem logischen Modell in Abbildung 6.8. Die Tabelle `VEG_MAP_POLY` als schwacher Entitätstyp von `VEGETATION_MAP` beinhaltet die räumlichen Informationen als Polygone, welche noch mit dem entsprechenden Vegetationstyp verlinkt sind. Ein Polygon kann also nur eingefügt werden, wenn ein zugehöriger Zeitstand der Vegetationskarte vermerkt wird. Eine solche Modellierung hat den Nutzen, dass in einer Abfrage ein ausgewählter Zeitstand der Vegetationskarte direkt als eine einzige Entität adressiert werden kann und die räumliche Information in separater Form vorhanden sind. Von Vorteil wäre gegebenenfalls sogar eine Modellierung von `VEG_MAP_POLY` als schwacher Entitätstyp von beiden anderen Relationen, damit beim Vegetationstyp auch nur bereits vorhandene Kategorien eingefügt werden können.

Im gezeigten Beispiel würde die Vegetationskarte vom Sommer 2009 aus zwei Polygonen (345 und 346) bestehen, welche vom Vegetationstyp 'riverine forrest' respektive 'woodland' sind. Für den Stand vom Frühling 2010 wird nur ein 'grassland'-Polygon angezeigt.

VEGETATION_MAP		VEGETATION_TYPE		VEG_MAP_POLY			
VEG_MAP_ID	SEASON	VEG_TYPE_ID	VEG_TYPE_NAME	VEG_MAP_POLY_ID	VEG_MAP_ID (FK)	VEG_TYPE_ID (FK)	SHAPE
1	SUMMER 2009	1	RIVERINE FORREST	345	1	1	GEOMETRY
2	FALL 2009	2	GRASSLAND	346	1	4	GEOMETRY
5	SPRING 2010	4	WOODLAND	132	5	2	GEOMETRY

Abbildung 6.8: Logisches Modell der Relationen rund um die zeitliche variierenden Vegetationsinformation im Vektorformat (`VEGETATION_MAP`) mit einigen fiktiven Datenwerten.

## Kapitel 7

# Situation der raum-zeitlichen Information im Inkawu Vervet Project

In diesem Kapitel wird die Situation dargestellt, wie die räumlichen, zeitlichen sowie raum-zeitlichen Informationen derzeit aufgenommen und gehandhabt werden.

### 7.1 Räumliche Informationen

Einen ausschliesslich räumlichen Aspekt besitzen derzeit die Daten der Vegetationskarte, es gibt nur einen Stand und die unterschiedlichen Vegetationstypen *riverine forest*, *grassland*, *thicket*, *woodland*, usw. haben noch keine zeitlich variierenden Informationen. Das Ziel des Inkawu Vervet Projektes ist jedoch der Einbezug von zeitlich variierender Kontextinformation auch bei der Vegetationskarte, weshalb der Entitätstyp `VEGETATION_MAP` trotz noch nicht vorhandenen Daten bereits so modelliert wurde (siehe Abbildung 6.4).

Weitere im Projekt aufgenommenen Daten, welche ausschliesslich einen räumlichen Bezug ohne Zeitinformation besitzen, gibt es nicht ausdrücklich. Man könnte z.B. ein Strassen-Netzwerk, Ortschaften oder auch den Standort der Forschungsanstalt dazu zählen, doch zum Zeitpunkt dieser Arbeit existierten keine solchen Daten.

### 7.2 Zeitliche Informationen

Als Daten mit ausschliesslich zeitlichem Aspekt wird der LOG einmal pro Tag gemacht (falls überhaupt Beobachtungen getätigt wurden) und mit einem automatischen Zeitstempel versehen. Dabei werden auch Änderungen in der Gruppenzusammensetzung (Tabelle `GROUPCHANGE`) vermerkt und mit dem im Log gemachten Zeitstempel versehen. Damit besitzen Veränderungen von Gruppen immer eine zeitliche Information und können in der Gruppengeschichte eingeordnet werden.

Es wäre ein ziemlich grosser Zufall, den genauen Zeitpunkt einer Änderung (wenn z.B. ein Tier emigriert) in der Gruppenkomposition festzuhalten, da die Forscher nicht rund um die Uhr bei allen Gruppen sein können. Deshalb reicht es auch, beobachtete Veränderungen in der Gruppenzusammensetzung einmal pro Tag festzuhalten. Dass diese Änderungen teils mehrere Tage zurückliegen können, ist laut Forscher nicht sehr schlimm, da eine (Un-)Genauigkeit von ein paar Tagen in den Analysen über die demographische Zusammensetzung der Gruppen wenig ausmache.

Die regelmässig gemachten Beobachtungen (*RegularObservations*) von Gruppen und Individuen (SCAN zur vollen und halben Stunde für je 10 Minuten, FOCAL dazwischen für je 10 Minuten) werden ebenfalls mit einem automatischen Zeitstempel pro Messung (alle Minuten) versehen. Der Beginn eines Gruppen-Scans ist jeweils zur vollen und halben Stunde, damit der automatische GPS-*fix* mehr oder weniger zutreffend ist. Zusätzlich werden die Forscher angewiesen, beim Protokollieren von SCANS im PDA manuell die Zeit auf die nächste halbe Stunde zu runden, um die Funktion von gewissen Algorithmen zu ermöglichen. Die SCAN-Daten können über den Tag gesehen als regelmässig betrachtet werden, da eine Gruppe jedoch nur alle 4-5 Tage (oder noch weniger) beobachtet wird, kann man die Daten gerade so gut auch als unregelmässig ansehen.

Neben dem Übertragen eines Zeitstempels werden die FOCAL Beobachtungen zusätzlich in eine der vier über den Tag festgelegten Zeitzonen eingeteilt, um das Beobachten eines einzelnen Individuums über den Tag gleichmässig zu verteilen.

### 7.3 Raum-Zeitliche Informationen

GPS-Daten werden einerseits über den regelmässigen Download der Halsband-Messungen, als auch durch die Forscher über *handheld*-Geräte gemacht. Die Daten haben mit Koordinaten und Zeitstempel jeweils einen räumlichen sowie zeitlichen Aspekt.

Über die Halsband-GPS-Geräte wird alle 30 Minuten jeweils zur vollen und halben Stunde während der aktiven Periode der Tiere (ändert monatlich mit den Jahreszeiten, z.B. 05:00-18:00 im Mai, 05:30-17:30 im Juni) automatisch ein *fix* gemacht, welche rund alle Monate einmal heruntergeladen werden. Da die Tiere in der Nacht schlafen und sich deshalb nicht bewegen, hat man aus Energiespargründen diese begrenzte Messperiode gewählt. Die Zuweisung der rohen GPS-Daten zu den Gruppen erfolgt manuell und es wird nur der Zeitpunkt des *fixes* gespeichert (= *valid time*), die *transaction time* wird trotz meist grosser Verzögerung beim Importieren ignoriert. Es werden jedoch mit der GPS-Zeit (entspricht ungefähr UTC/Greenwich-Zeit) und einem Zeitstempel mit vordefiniertem Wert (also Lokal-Zeit) zwei verschiedenen Zeiten (*multiple times*) importiert.

Neben dem GPS-Empfänger tragen die einen Tiere auch noch einen Accelerometer, der die Bewegungen der Affen misst. Die Daten werden gleichzeitig mit den GPS-Daten heruntergeladen und sind eine Art raum-zeitliche Daten, mit zeitlicher Auflösung von 2 Minuten und einer Rauminformation in einer etwas speziellen Form. Neben der grossen Menge und Komplexität dieser Daten (siehe Seite 46) sind sie ausserdem sehr schwierig zu entschlüsseln, da bei den Affen die Devices immer verrutschen. So könnte eine interpretierte Bewegung des Tieres nach links eigentlich eine Bewegung nach rechts sein, je nachdem wo der Accelerometer gerade am Hals platziert ist.

Auch die *handheld*-GPS Daten, welche zu den unregelmässig auftretenden Beobachtungen gemacht werden, besitzen bereits einen Zeitstempel und natürlich die Koordinaten der *fixes*. Die Positionen müssen dann manuell über Gruppe und Zeit den einzelnen Ereignissen zugewiesen werden, in der Datenstruktur besteht noch keine direkte Verbindung zwischen den *handheld*-GPS-Daten und den auftretenden Ereignissen, was im ER gut ersichtlich wird.

Das Zusammentreffen von zwei verschiedenen Gruppen (*between group encounter*) kann als Event bezeichnet werden, da das Ereignis eine Start- und Endposition sowie auch Start und Endzeit besitzt. Ein solcher Gruppenkonflikt ist somit ein klassischer raum-zeitlicher Event mit einer Ausdehnung sowohl in Raum als auch in Zeit.

Die anderen unregelmässig auftretenden Ereignisse (*OnOccurrence*) wie Alarmrufe, sexuelles Verhalten und Konflikte innerhalb von Gruppen haben nur einen einzelnen Zeitstempel und einen mit *handheld*-GPS separat aufgenommenen GPS-Messpunkt. Laut der Kontaktperson werden teilweise Beobachtungen ohne GPS-Daten aufgenommen, da dies in der Hektik von Alarmrufen oder Konflikten vergessen werden kann.

Aus Satellitenbildern werden ungefähr monatliche *food availability*- und *habitat productivity*-Raster erstellt. Obschon dies eigentlich Werte mit kontinuierlichen Änderungen sind, werden sie als *Snapshots* behandelt, d.h. es wird immer ein ganzer Layer pro Zeitpunkt gespeichert.

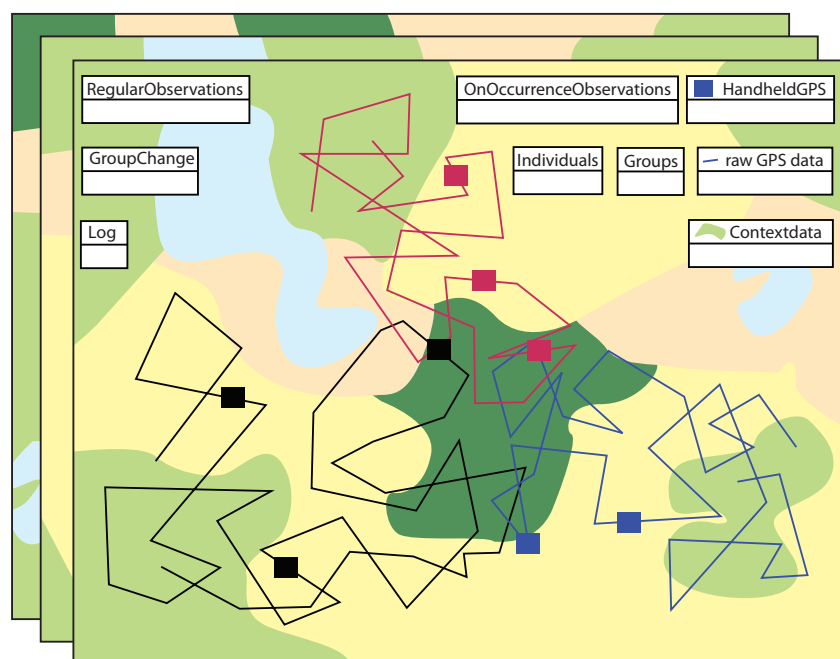


Abbildung 7.1: Schematische Darstellung, welche Art räumlichen, zeitlichen und raum-zeitlichen Daten im Inkawu Vervet Project alles vorhanden sind. Gruppen sind durch verschiedene Farben dargestellt, die weissen Rechtecke repräsentieren Tabellen. Regelmässige sowie unregelmässige Beobachtungen bei auftretenden Ereignissen können nur bei Anwesenheit der Forscher aufgenommen werden, bei den regelmässigen Beobachtungen wird keine räumliche, sondern nur die zeitliche Information gespeichert. Im Gegensatz dazu sind Positionsangaben von jeder Gruppe mit einer zeitlichen Granularität von 30 Minuten vorhanden. Die Kontextinformationen ändern zum Teil ebenfalls im Laufe der Zeit.

Die schematische Darstellung in Abbildung 7.1 zeigt die im Projekt involvierten räumlichen, zeitlichen und raum-zeitlichen Daten. An ausgesuchten Tagen werden regelmässige Beobachtungen (*RegularObservations*), Angaben zu Gruppenänderungen (*GroupChange*), der Log sowie innerhalb dieser Perioden auftretende Ereignisse (*OnOccurrence*) festgehalten. Bei unregelmässig auftretenden Ereignissen wird zudem mit einem *handheld*-GPS-Gerät eine Ortung vorgenommen und separat gespeichert; reguläre Beobachtungen besitzen keine Rauminformation. Dies wird dadurch veranschaulicht, indem die regulären Beobachtungen als eigenständige Tabelle und *OnOccurrence*-Beobachtungen neben der

*HandheldGPS*-Tabelle dargestellt wird. Regelmässige GPS-Messungen (eigentlich Punkte, hier als Trajektorien dargestellt) werden von Individuen erfasst, repräsentieren jedoch auch die zugehörige Gruppe. Dazu kommen noch zeitlich variierenden Kontextinformationen in Form von Polygonen und Raster.

## Zusammenfassung

Grundsätzlich kann gesagt werden, dass vielfältige Daten mit unterschiedlichen Raum- und Zeitinformationen vorliegen, jedoch ist die Handhabung dieser mit einem grossen Arbeitsaufwand verbunden. Einerseits ist dies zeitaufwändig, vor allem aber ist es eine mögliche Fehlerquelle. Durch eine geschickte Modellierung und den Einbezug von automatisierten Abläufen könnte dieses Fehlerpotential vermindert werden.

Im Projekt gibt es sehr viele räumliche und zeitliche Informationen, das Potential dieser Daten wird jedoch noch nicht voll ausgenutzt. Gerade die Vernetzung der verschiedenen Elemente ist mager und verhindert dadurch, dass grundsätzlich einfache Abfragen nur bedingt oder über grosse Umwege (d.h. komplizierte Queries) möglich sind.

Gerade die GPS-Halsband-Daten sind z.B. nicht direkt mit den Daten zu Verhaltensbeobachtungen verknüpft, welche überhaupt keine räumlichen Informationen besitzen. Um die in 5.1 besprochenen Forschungsfragen des Projektes zu beantworten, sind diese Verbindungen jedoch unerlässlich. Fragen nach der Kontextinformation eines gewissen Punktes sollten nicht über kompliziertes Exportieren und Importieren von einzelnen Datensets gelöst werden, sondern über gezielte Abfragen an ein miteinander verknüpft, räumliches Datenset.

Im nächsten Kapitel wird auf die angesprochene Problematik eingegangen und es werden einige Beispiele präsentiert, wie mit diesen reichhaltigen Bewegungsdaten anders und womöglich auch effizienter verfahren werden könnte.

## Kapitel 8

# Raum-zeitliche Anreicherung

In den folgenden Kapiteln wird auf ausgesuchte Beispiele fokussiert, um das 'räumliche Potential' der im Inkawu Vervet Projekt gesammelten Daten zu zeigen. Es wurde auf die in Kapitel 5.4 ausgewählten Abfragen eingegangen, d.h. es soll eine automatische Erkennung von *between group encounters* (8.3) sowie eine Anreicherung der räumlichen Objekte mit dem zeitlichen ändernden Kontext versucht werden (8.4). Zusätzlich soll in 8.1 der Prozess des GPS-Datenimports vereinfacht, und in 8.2 die bisher nicht-räumlichen Daten der regulären Beobachtungen mit Positionsinformationen angereichert werden.

Mit Hilfe von schematischen Darstellungen, *entity relationship models*, Tabellen und SQLs (*Structured Query Language*) wird auf die genannten Beispiele eingegangen. Um die SQLs auch zu testen, wurden die Relationen in einer Oracle Datenbank erstellt, da ich bereits über ein wenig Erfahrung mit dieser Datenbanksoftware besitze. Als räumlicher Datentyp wurde *SDO\_Geometry* verwendet, die Standard-Option der räumlichen Erweiterung Oracle Spatial, welche räumliche Abfragen auch direkt per SQL erlaubt. Es wurden nur ausgewählte Attribute für die Relationen implementiert und mit einem kleinen Set an Mockup-Daten gefüllt, um die Funktionalität der SQLs zu testen.

### 8.1 Automatische Zuweisung der rohen GPS-Daten an Individuen

Wie in Kapitel 7 beschrieben, werden derzeit die monatlich heruntergeladenen GPS-Rohdaten im Inkawu Vervet Projekt manuell den Tieren zugewiesen und in Tabellen übertragen. Dieser Weg der GPS-Dateningegration ermöglicht zwar die visuelle Kontrolle, ob einzelne *fixes* mehrmals integriert werden, er birgt aber auch Gefahren. Denn mit der Verwechslung eines GPS-Gerätes werden die Daten den falschen Individuen bzw. Gruppen zugeordnet. Ausserdem wird für den manuellen Einlesungsprozess Zeit benötigt, welche bei wichtigeren Dingen, wie z.B. der Analyse der Daten eingesetzt werden könnte. Ausserdem kann die Überprüfung nach Duplikaten von einem Computer schneller und je nach Modellierung auch ohne Fehler vorgenommen werden.

Um die rohen GPS-Daten zu integrieren, wird in ISAMUD (siehe Kapitel 3.7.2) das in Abbildung 3.13 dargestellte Modell verwendet. Die Zuordnung der rohen GPS-Daten wird über eine zusätzliche Tabelle ermöglicht, in welcher die Geräte-ID, die ID des tragenden Individuums sowie Start- und Endzeit der Trageperiode gespeichert werden.

Diese Variante könnte in etwa auch als Lösung im Inkawu Vervet Projekt angewendet werden, denn die Voraussetzungen dazu bestehen. Die rohen GPS-Daten, welche ungefähr alle Monate heruntergeladen werden, besitzen neben den Koordinaten auch die ID des zugehörigen GPS-Gerätes und einen Zeitstempel (GPS-Zeit bzw. UTC/Greenwich Time). Ich schlage deshalb vor, eine Tabelle **WEARS** zu erstellen (Abbildung 8.1), in welcher eine Zuweisung der Tiere zu den vorhandenen GPS-Geräten erfolgt. Darin sollen neben der **MONKEY\_ID** und der **DEVICE\_ID**, die Start- und falls bereits vorhanden, die Endzeit (ansonsten wird NULL eingesetzt) enthalten sein.

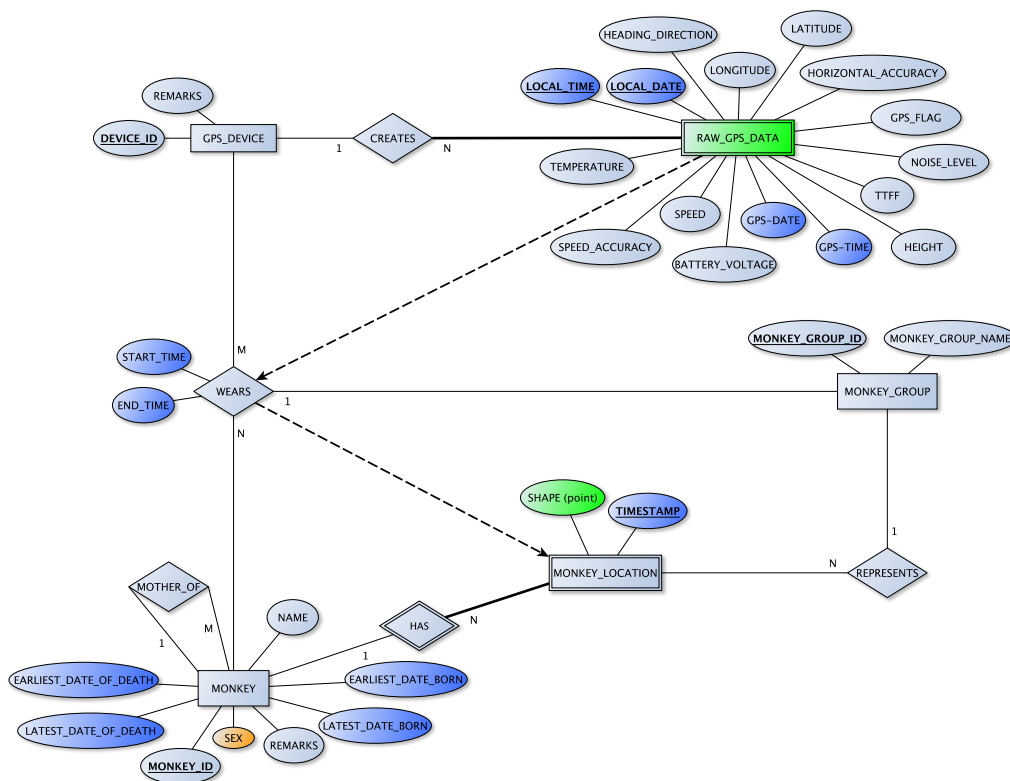


Abbildung 8.1: ER-Struktur, wie rohe GPS-Daten automatisch den Affen zugeordnet werden und räumliches Objekt in Tabelle **MONKEY\_LOCATION** erstellt wird.

Für jeden *fix* wird ein räumliches Punktojekt in Tabelle **MONKEY\_LOCATION** mit Zeitangabe sowie Beziehungen zum zugehörigen Affen und der Gruppe erstellt (durch gestrichelten Pfeil in Abbildung 8.1 angedeutet). Da für jeden Affen nur eine Positionsangabe pro Zeitpunkt möglich ist, wird **MONKEY\_LOCATION** als schwacher Entitätstyp von **MONKEY** modelliert und damit ein Kontrollmechanismus eingeführt. Es wird dabei der lokale Zeitstempel des GPS-Gerätes verwendet, somit wird keine Umrechnung mehr benötigt. Je nach Wunsch könnten weitere Attribute von **RAW\_GPS\_DATA** (z.B. SPEED, SPEED ACCURACY, HEADING DIRECTION, TEMPERATURE, ...) zusätzlich eingefügt werden.

Diese Schritte werden nun auf der kleinen Prototyp-Datenbank mit einem kleinen Mockup-Datenset durchgeführt. Der erste Schritt des Datenimportes, die Trennung von Accelerometer- und GPS-Daten, sowie das Einfügen in die Datenbank, wird hier nicht behandelt. Es wird davon ausgegangen, dass die rohen GPS-Daten bereits in aufbereiteter Form in der Datenbank (Tabelle **RAW\_GPS\_DATA**) zur Verfügung stehen.



Mit dem SQL 8.1 funktioniert das Einlesen der GPS-Daten automatisch, es werden jeweils gleich räumliche Datentypen (*spatial data types*) aus den geographischen Längen- und Breitenangaben erstellt. Zusätzlich wird über die Tabelle WEARS sichergestellt, dass die einzelnen Messungen den richtigen Individuen zugeordnet werden. Auch wenn ein Halsband nicht mehr funktioniert und ausgewechselt wurde, können die Daten ohne Änderungen im Vorgehen in die Datenbank aufgenommen werden, die Tabelle WEARS muss einfach auf dem neusten Stand gehalten werden. Falls ein Tier mit GPS-Halsband sterben sollte, kann anhand der nicht mehr ändernden Position das Halsband lokalisiert, entfernt und an einem anderen Individuum festgemacht werden. Auch hier muss dann einfach WEARS aktualisiert werden. Zudem werden nur gültige GPS-Daten, d.h. mit GPS-FLAG=1 in die Tabelle MONKEY\_LOCATION aufgenommen. Der SQL funktioniert auf dem Testdatensatz mit 33 rohen GPS-Positionen (davon zwei mit GPS-FLAG=0), es ergibt 31 Einträge in MONKEY\_LOCATION.

```
INSERT INTO MONKEY_LOCATION(monkey_id, REPRESENTS_GROUP, Time, Shape)
SELECT w.monkey_id, w.represents_group, r.local_date, MDSYS.SDO_GEOMETRY(
2001, NULL, MDSYS.SDO_POINT_TYPE(r.longitude,r.latitude,r.height), NULL, NULL)
FROM RAW_GPS_DATA r, wears w
WHERE r.gps_flag='1'
AND r.gps_device_id=w.gps_device_id
AND ((r.local_date >= w.start_time) AND (r.local_date <=w.end_time))
OR ((r.local_date >= w.start_time) AND (w.end_time IS NULL)))
```

SQL 8.1: Zuweisen der rohen GPS-Daten aus RAW\_GPS\_DATA an Individuen und Erstellen von Punktdaten in Tabelle MONKEY\_LOCATION.

Dieser SQL kann nach dem Laden von neuen rohen GPS-Daten in RAW\_GPS\_DATA einfach wiederholt werden und neue MONKEY\_LOCATIONS werden automatisch erstellt. Falls für ein Individuum für den exakt gleichen Zeitpunkt mehrere Locations vorliegen sollten, gibt es eine Fehlermeldung wegen einem Primärschlüsselkonflikt (Kombination von Zeitstempel und MONKEYID darf nur einmal vorkommen).

Abbildung 8.2 zeigt wie der SQL 8.1 funktioniert: Der *fix* aus RAW\_GPS\_DATA mit der Position (13,13) und einer Höhe von 652m wird über die GPS\_DEVICE\_ID=1386 und den Zeitstempel '10.05.2010 15:30' mit der Tabelle WEARS dem Individuum AF001 sowie der Gruppe 1 zugewiesen. Mit diesen Angaben wird in MONKEY\_LOCATION ein neues räumliches, dreidimensionales Objekt mit dem *SDO\_Geometry Type* erstellt. Damit werden die bisher rohen und losen GPS-Daten semantisch angereichert und erhalten eine Verbindung zu den übrigen Daten.

Diese Informationen wären grundsätzlich auch über einfachere SQLs abfragbar, müssten jedoch oft wiederholt werden, was einen hohen Zeitaufwand bedeuten würde. Die rohen GPS-Daten als auch die Tabelle WEARS bleiben mit dieser Lösung weiter in der Datenbank bestehen und werden mit den weiteren Messungen jeweils ergänzt. Durch die Tabelle MONKEY\_LOCATION sind die Positionsdaten als räumliche Datentypen bereits den Individuen zugewiesen, können viel einfacher abgefragt werden und würden (abhängig von Update-Frequenz) immer relativ aktuell die Position der Tiere mit GPS-Halsband verraten. Wenn man aus bestimmtem Grund noch Zugriff auf die Original-Daten haben muss, so kann man dies für ausgewählte Zeitperioden, GPS-Halsbänder oder (durch primäres Konsultieren von WEARS) Individuen immer noch per SQL 8.2 tun. Es werden hier jedoch nur die rohen räumlichen Daten, d.h. die geographische Länge und Breite abgefragt.

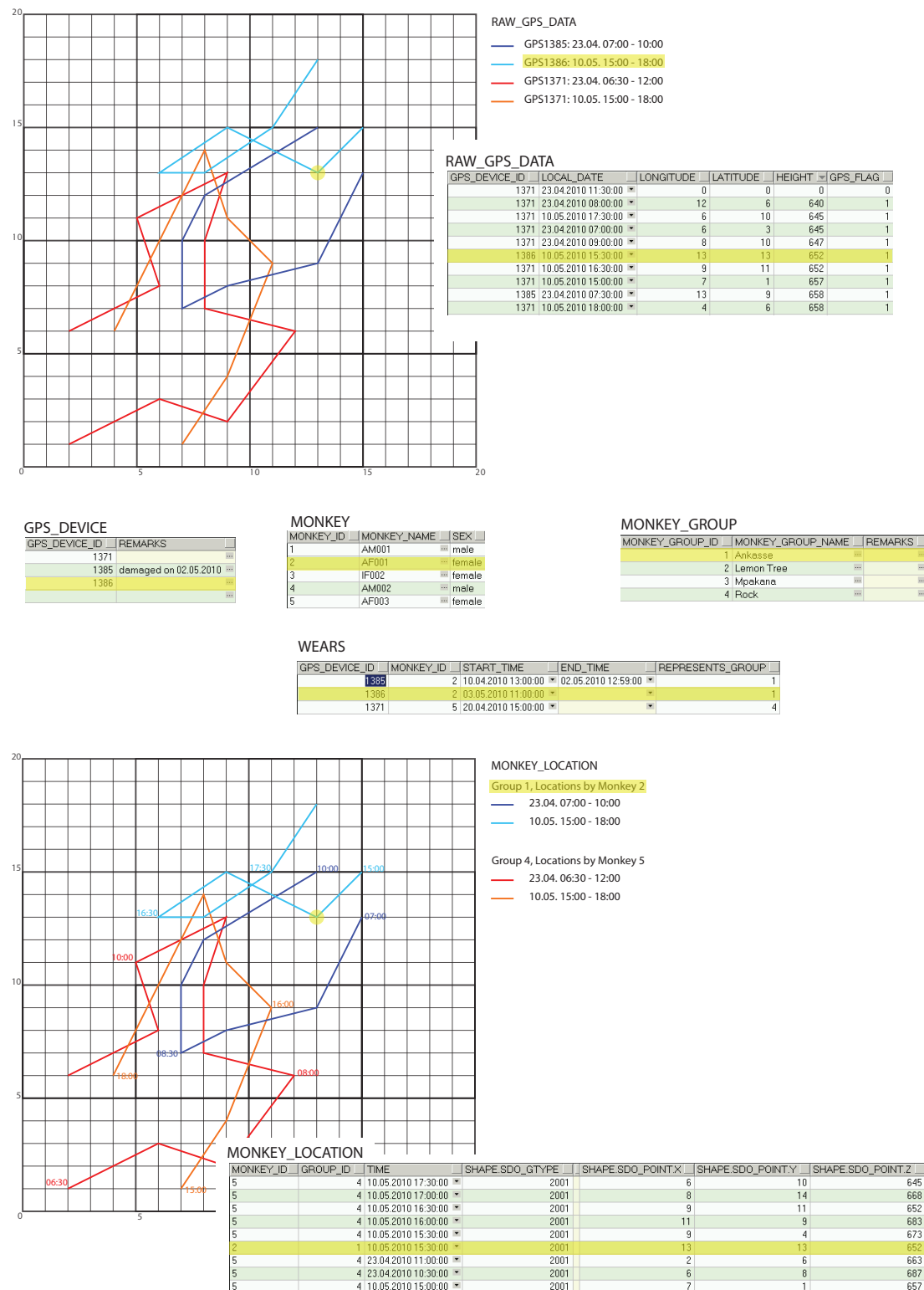


Abbildung 8.2: Automatische Zuweisung der rohen GPS-Daten aus Tabelle RAW\_GPS\_DATA an die Individuen. Über die Tabelle WEARS wird mit dem SQL 8.1 aus den Paaren geographischer Länge/Breite ein räumliches Objekt für den zugehörigen Affen aus MONKEY erstellt, mit der Gruppe verknüpft und in MONKEY\_LOCATION eingefügt. Zusätzlich können auch noch weitere Attribute aus den rohen Daten angehängt werden.

```
SELECT r.longitude, r.latitude, r.local_date
FROM RAW_GPS_DATA r
WHERE r.gps_device_id = '1385'
AND local_date BETWEEN to_date('2009/05/10 12:00:00', 'yyyy/mm/dd hh24:mi:ss')
AND to_date('2010/05/10 20:00:00', 'yyyy/mm/dd hh24:mi:ss')
```

SQL 8.2: Abfrage von geographischer Länge und Breite der rohen GPS-Daten.

Mit einer solch einfachen Abfragemöglichkeit per SQL 8.2 besteht eine Kontrolle für allfällig eingeschlichene Fehler, welche mit einer Automatisierung durch Skripte durchaus anfallen können. Falls bei der Analyse irgendwelche Ausreisser oder unstimmige Werte erscheinen, so kann man dem mit solch einfachen Queries auf den Grund gehen.

### 8.1.1 Handhabung der Gruppeninformation

Jede Gruppe unter Beobachtung besitzt mindestens ein Individuum, welches ein GPS-Halsband trägt und womit regelmässig immer zur vollen und halben Stunde Messungen gemacht werden. Die GPS-Geräte werden jeweils den Weibchen angehängt, da diese die Gruppen nie wechseln. Wie in Kapitel 8.1 bereits erwähnt wurde, ist die Grundidee der Zuweisung von rohen GPS-Positionen vom ISAMUD Modell. Das Modell ist jedoch nicht auf Gruppen ausgelegt, es wird nur auf die einzelnen Tiere Bezug genommen.

Da im Inkawu Vervet Projekt die Gruppen meist sowieso nur durch die GPS-Daten eines einzelnen Individuums repräsentiert werden, könnte man einfach ein Individuum einer Gruppe gleichsetzen. Um aber für Fälle vorzusorgen, wo mehrere Individuen gleichzeitig für dieselbe Gruppe GPS-Daten sammeln, oder wo ein Tier stirbt und das GPS-Gerät ersetzt wird, schlage ich das Attribut REPRESENTS GROUP (bzw. Beziehung zu MONKEY\_GROUP) in Tabelle WEARS vor (siehe Abbildung 8.1). Damit sind die Punktdaten von Tabelle MONKEY\_LOCATION immer einer Gruppe zugewiesen.

Um die Wege bzw. *trajectories* von Gruppen zu analysieren, kann MONKEY\_LOCATION nach TIMESTAMP geordnet und darin eine Gruppe selektiert werden. Auch beim Einsatz mehrerer GPS-Geräte pro Gruppe ist dieser Weg möglich. Im Laufe dieser Arbeit wurde oftmals das Erstellen von Gruppen-Trajektorien diskutiert und diverse Entwürfe erstellt und wieder verworfen. Beim Erstellen von Linienobjekten pro Gruppe und Tag verliert man die Zeitinformation und alle anderen Attribute pro *fix* und kreiert dabei nur Redundanz. Mit einem Entitätstyp GROUP\_TRAJECTORY (siehe Abbildung 8.3), bei welchem für jeden Tag pro Gruppe eine Beziehung zu den einzelnen *fixes* in MONKEY\_LOCATION erstellt werden, kreiert man nur eine komplizierte Struktur. Zusätzlich sind es redundante Informationen, da die Daten auch sonst per SQL abgefragt werden können. Da die Informationen auch anders gut abgerufen werden können, wurde dies weggelassen.

Dadurch, dass die Geometrie in MONKEY\_LOCATION als Punkte gespeichert ist, bleibt die Zeitinformation (und allfällige zusätzliche Attribute) für jeden *fix* erhalten. Zu Analysezwecken können Trajektorien immer noch als Linienobjekte erstellt werden, um Redundanzen zu vermeiden. Diese werden jedoch nicht explizit modelliert und gespeichert. Trajektorien können als eine abgeleitete Grösse betrachtet werden, denn diese können unterschiedlich eingeteilt werden: Tages- oder Monatstrajektorien, Morgen- oder Abendtrajektorien einer Gruppe usw. Deshalb wurde keine fixe Gruppentrajektorie modelliert, die Frage, wo z.B. Gruppe 1 am 23. April 2010 war, kann relativ einfach über SQL 8.3 beantwortet werden.

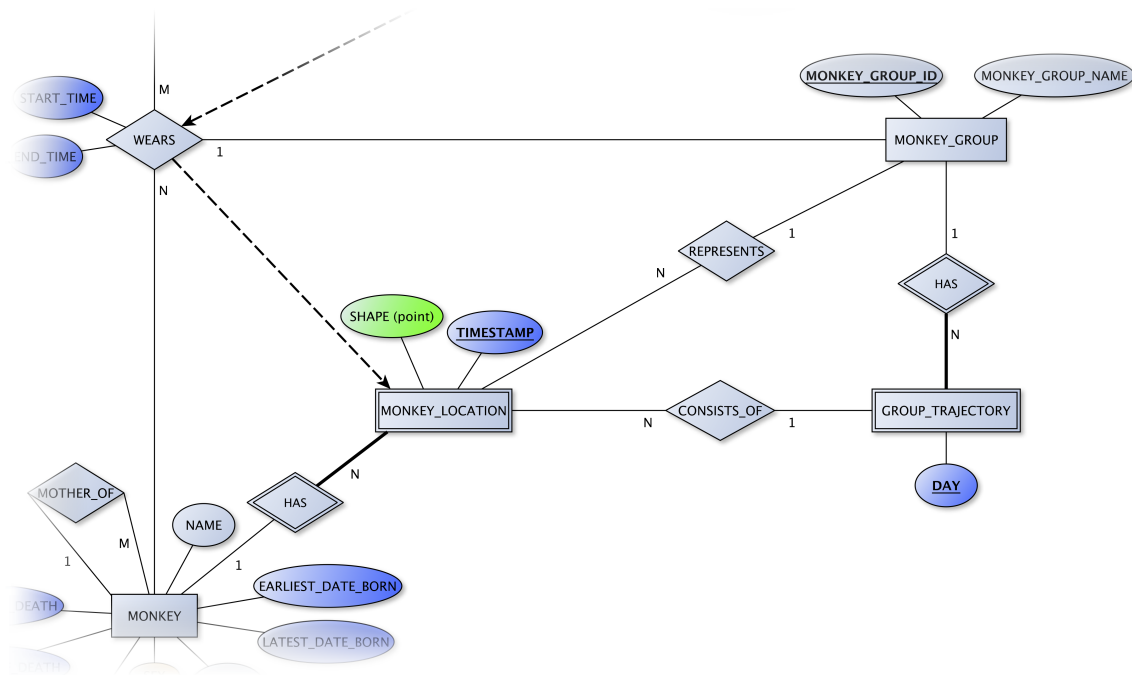


Abbildung 8.3: ER-Struktur mit Entitätstyp `GROUP_TRAJECTORY`, welche nicht verwendet wurde, da wegen zwei schwachen Entitätstypen komplizierte und wegen Redundanz auch unnötige Beziehungen entstanden wären.

```
SELECT m.represents_group, m.time, m.shape
FROM MONKEY_LOCATION m
WHERE m.represents_group = '1'
AND m.time BETWEEN to_date('2010/04/23 00:00:00', 'yyyy/mm/dd hh24:mi:ss')
AND to_date('2010/04/23 23:59:59', 'yyyy/mm/dd hh24:mi:ss')
```

SQL 8.3: Trajektorie einer Gruppe an einem bestimmten Datum (hier 23.04.2010).

Mit SQL 8.3 werden alle Punkte der Gruppe vom abgefragten Datum ausgewählt und stehen für weitere Analysen wie z.B. die Erstellung einer *home range* zur Verfügung.

## 8.2 Automatische Lokalisierung der regelmässigen Beobachtungen

Die regelmässigen Beobachtungen, also *SCAN* oder *FOCAL* haben keine räumlichen Informationen, jedoch immer Angaben zu Gruppe und Zeit. Da die Positionen aller Gruppen regelmässig aktualisiert und in `MONKEY_LOCATION` gespeichert werden (siehe Kapitel 8.1), kann eine Lokalisierung über die Gruppen- und Zeitangabe vorgenommen werden. Damit können regelmässigen Beobachtung im Raum fixiert werden, was zusätzliche Abfragen wie z.B. zu Kontextinformationen (in Kapitel 8.4 behandelt) erlaubt. Die Lokalisierung dieser Beobachtungen kann über mehrere Wege erfolgen, z.B. unter Annahme von linearen Bewegungen zwischen den einzelnen *fixes* oder mit Einbezug von anderen Bewegungsmodellen. Mit Interpolation zwischen den einzelnen *fixes* könnte dann eine Positionsangabe extrahiert werden. Der Einfachheit halber wurden in meinem kleinen Datensatz jedoch nur

die bereits bestehenden Geometrien aus `MONKEY_LOCATION` verwendet. Bei der Zuweisung wird die Punktgeometrie desjenigen *fixes* eingesetzt, welche zeitlich am nächsten bei der Beobachtung ist und die gleiche Gruppeninformation besitzt. Damit werden einerseits die Beobachtungen der Gruppe und nicht dem Individuum zugeordnet, und andererseits nicht wirklich auf den Meter genau gespeichert, da auf den nächstmöglichen Punkt zurückgegriffen wird.

```
UPDATE OBS_SCAN
SET shape = (
SELECT MONKEY_LOCATION.shape
FROM MONKEY_LOCATION
WHERE OBS_SCAN.GROUP_ID = MONKEY_LOCATION.REPRESENTS_GROUP
AND OBS_SCAN.DATE_AND_TIME = MONKEY_LOCATION.TIME) --same timestamp
```

SQL 8.4: Einfügen der Punktgeometrie aus `MONKEY_LOCATION` für Messzeitpunkte in `OBS_SCAN`, mit exakt gleicher Gruppen- und Zeitinformation.

Mit SQL 8.4 werden jeweils nur diejenigen `SCANS` bzw. `FOCALs` lokalisiert, welche exakt den Zeitstempel eines GPS-*fixes* besitzen. Da damit nur sehr wenige Beobachtungen eine Raum-Information bekämen, wurde der SQL ein wenig abgeändert. Da die GPS-*fixes* regelmässig alle 30 Minuten stattfinden, kann die Geometrie desjenigen Messpunktes eingefügt werden, welche eine Zeitdifferenz von kleiner als 15 Minuten und natürlich die gleiche Gruppeninformation besitzt:

```
UPDATE OBS_SCAN
SET shape =
(SELECT "shape"
FROM ( SELECT o.scan_id as "scan_id", o.group_id as "group", o.date_and_time,
m.time, m.shape as "shape", m.represents_group as "shapegroup",
sqrt(power((o.date_and_time-m.time),2)) as "timediff"
FROM MONKEY_LOCATION m, OBS_SCAN o
WHERE o.group_id = m.represents_group)
WHERE "timediff" < 0.01041666 --smaller than 15min (1 day = 1)
AND "scan_id" = OBS_SCAN.scan_id)
```

SQL 8.5: Einfügen der Punktgeometrie aus `MONKEY_LOCATION` für Messzeitpunkte in `OBS_SCAN`, mit Gruppen- und Zeitinformation < 15min; Auswahl des zeitlich nächsten Punktes, also keine Interpolation. Resultat in Abbildung 8.4

Somit wird die Punktgeometrie aus `MONKEY_LOCATION` in `OBS_SCAN` eingefügt, welche einerseits die gleiche GruppenID und andererseits eine möglichst kleine Differenz zwischen den Zeitstempeln besitzt. Mit meinen einfachen Rahmenbedingungen funktioniert SQL 8.5 sehr gut und die `SCAN`-Beobachtungen werden mit einer Punktgeometrie ergänzt. In Abbildung 8.4 wird die Beobachtung mit `SCAN_ID` = 2 und einem Zeitstempel von 07:31 der `MONKEY_LOCATION` mit Zeitstempel 07:30 zugewiesen. Mit SQL 8.5 wird die Geometrie direkt in `OBS_SCAN` eingefügt.

Es muss jedoch beachtet werden, dass dieser SQL unter gewissen Umständen nicht mehr funktioniert. Und zwar dann, falls mehrere *shapes* in der `WHERE`-Klausel '`timediff`' < 15min in Frage kämen. Dies könnte passieren, falls ein GPS-*fix* z.B. um 12:03 und der nächste um 12:30 gemacht würde. Falls nun ein `SCAN` mit dem Zeitstempel 12:17 lokalisiert werden sollte, so gibt es die Meldung *single-row subquery returns more than one row*. Dasselbe Problem würde auftreten, falls mehrere Individuen für eine Gruppe GPS-Messungen erfassen. Dann kämen mit Sicherheit immer mehr als ein *shape* in Frage und man müsste den SQL 8.5 umschreiben, damit '`timediff`' minimal wird.

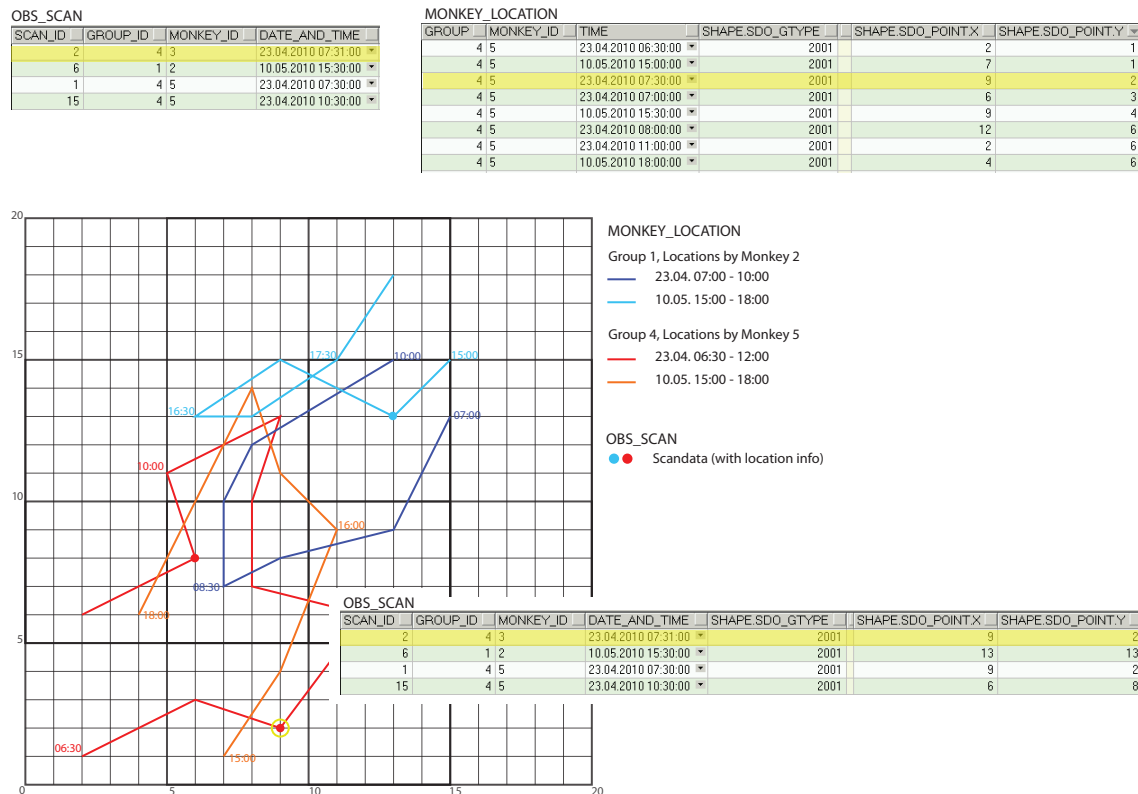


Abbildung 8.4: Zuweisung von Rauminformation an reguläre Beobachtung OBS\_SCAN anhand von Gruppe und Zeit. Beobachtung Nr. 2 vom 23.04.2010 um 07:31 wird über Gruppeninfo = 4 der MONKEY\_LOCATION mit Koordinaten 9,2 (23.04.2010 07:30) zugeordnet und in OBS\_SCAN eingefügt.

Ein Problem ist natürlich die Ungenauigkeit, welche sich durch die Übernahme der 'ungefähren' Position einschleicht. Auch wenn sich die Affen nicht sehr schnell bewegen, können falsche Positionen zugewiesen werden, was sich dann immer weiter in die Analyse (z.B. Kapitel 8.4 zu Kontextinformationen) ziehen kann. Man muss jedoch beachten, dass mit Hilfe von Bewegungsmodellen oder Interpolation die Position auch nur angenähert wird und sich Fehler einschleichen können.

Man muss noch anmerken, dass mit dieser Methode redundante Informationen gespeichert werden, was sicher nicht optimal ist und nicht gerade dem Sinn einer Datenbank entspricht. Denn wegen einer etwas komplizierten Datenstruktur (PK von MONKEY\_LOCATION besteht aus MONKEY\_ID und TIMESTAMP) wird nicht etwa eine Beziehung von SCAN zu MONKEY\_LOCATION erstellt, sondern die SHAPE-Information kopiert. Es gilt jedoch auch abzuwägen, inwiefern die bereitstehende SHAPE-Information von Vorteil sein kann im Gegensatz zu einer zusätzlichen Query im Prozess einer Analyse. Eine reine SQL-Query mit SELECT ohne UPDATE könnte zwar die Information zur Geometrie für weitere Queries abfragen und Redundanz würde vermieden. Ich sehe jedoch im Vorhandensein der Raum-Information direkt in der Tabelle der Beobachtungen einen grossen Vorteil, da die Abfragen einfacher zu formulieren sind. Dies wird umso wichtiger, wenn nicht die bereits vorhandenen GPS-Messpunkte übernommen werden, sondern mit komplizierteren Methoden eine Interpolation zwischen Punkten vorgenommen wird. Oder falls mehrere Af-

fen Positionsdaten für eine Gruppe aufnehmen und eine Interpolation zwischen mehr als zwei Messpunkten betätigt wird. Im Zuge von weiteren Analysen kann die Tabelle direkt und ohne *joins* angesprochen werden.

Auch die Beobachtungen zu den unregelmässig auftretenden Ereignissen (*OnOccurrence*) könnten mit Hilfe der Gruppeninformation sowie ihrem Datum- und Zeitstempel einer Gruppentrajektorie zugewiesen werden. Diese besitzen normalerweise eine Positionsangabe, doch wird diese separat mit einem *handheld*-GPS aufgenommen und muss manuell im Nachhinein dem Ereignis beigelegt werden. Falls in diesem Prozess etwas schief läuft oder die Forscher in der Hektik eines solchen auftretenden Ereignisses keine GPS-Messung machen, könnte ein Update der *shape*-Information per SQL 8.5 eventuell hilfreich sein. Dies wäre jedoch eher eine Notfalloption, denn anders als bei *SCANS*, sind die anderen Beobachtungen nicht auf die vollen und halben Stunden festgesetzt. Das heisst, dass die übernommene Position ziemlich fest abweichen könnte; auch Erik Willems sieht ausser bei *Scans* eine Notwendigkeit zur Interpolation zwischen den regelmässigen GPS-Daten.

### 8.3 Analyse von Gruppen-Trajektorien

Da die Forscher nur zeitlich begrenzt bei den verschiedenen Gruppen sein können, wird nur ein Teil der *between group encounters* aufgenommen. Doch gerade diese Vorfälle könnten anhand ihrer geometrischen und zeitlichen Eigenschaften erkannt werden. Wie aus Forschungsfragen 1 und 2 (Kapitel 5.1) und auch den ausgearbeiteten Abfragen (Kapitel 5.4) hervorgeht, liegt ein grosses Interesse darin, dass diese automatisch detektiert und quantifiziert werden können.

Dazu sollen die Punktgeometrien einer Gruppe (hier Nr. 4) mit denjenigen aller anderen Gruppen verglichen werden. Die Paare von Punkten müssen innerhalb einer gewissen Distanz (hier im Mockup-Datensatz 2m) sowie Zeitperiode (hier: 30min) liegen:

```
SELECT "foc_group", "foc_time", "x1", "y1", "enc_group", "enc_time", "x2", "y2",
      "group_dist"
FROM   (SELECT m2.represents_group as "foc_group", m2.time as "foc_time",
              m2.shape.sdo_point.x as "x1", m2.shape.sdo_point.y as "y1",
              m3.represents_group as "enc_group", m3.time as "enc_time",
              m3.shape.sdo_point.x as "x2", m3.shape.sdo_point.y as "y2",
              SDO_GEOM.SDO_DISTANCE(m2.shape, m3.shape, 0.005) as "group_dist"
        FROM monkey_location m2, monkey_location m3
        WHERE m2.represents_group='4' AND m3.represents_group <> '4')
WHERE "group_dist" < '2' --distance is smaller than 2(m)
AND ("foc_time" - "enc_time") <= '0.020833333'
AND ("enc_time" - "foc_time" ) <= '0.020833333' --time difference smaller than 30min
ORDER BY "foc_time";
```

SQL 8.6: Detektion von Gruppenzusammenstössen, Resultat siehe Abbildung 8.5.

Im doch ziemlich kleinen Datensatz mit 33 rohen Positionsangaben (davon zwei ungültig) funktioniert der SQL sehr gut und liefert die Konfliktorte (siehe Abbildung 8.5). Als Resultat werden nur die Gruppennummern, ihre Position (inkl. Zeit) sowie die Distanz zwischen den beiden Gruppen angegeben. Der Start- und Endpunkt des Konflikts muss aus dem Resultat herausgelesen (und allenfalls interpoliert) werden. Der SQL sollte jedoch auch mit einem grösseren Datensatz funktionieren, eventuell müsste man den Output ein wenig anpassen, da dieser schnell einmal unübersichtlich werden könnte.



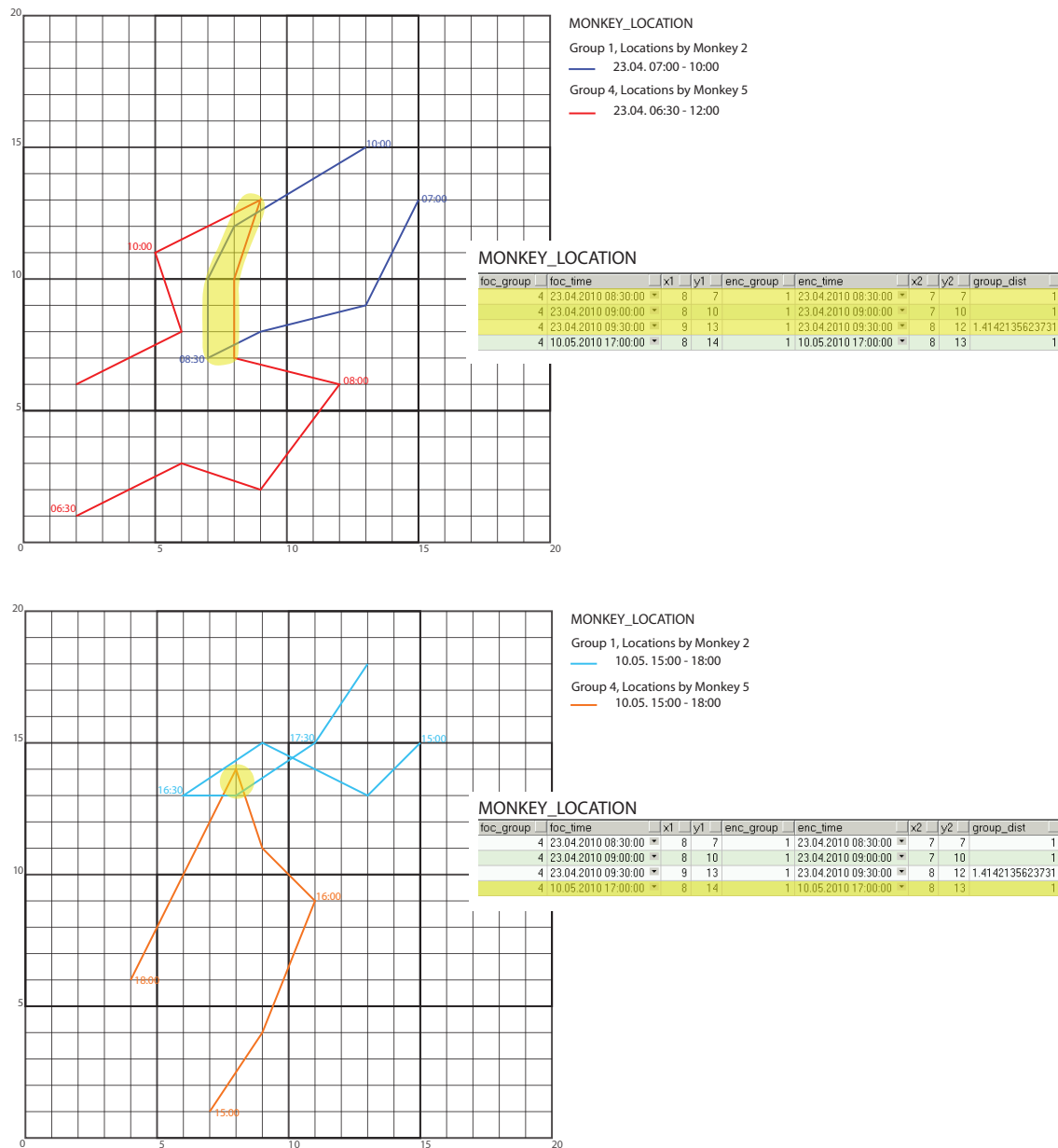


Abbildung 8.5: Detektion der Gruppenzusammenstößen aus Tabelle MONKEY\_LOCATION als Resultat von SQL 8.6. Man erkennt, dass sich der Konflikt am 23. April über mindestens eine Stunde hinwegzog, während am 10. Mai nur ein kurzes Abtasten stattgefunden zu haben scheint.



Im SQL 8.6 wurde nur nach Konflikten mit Gruppe 4 gesucht, da der Datensatz nur aus zwei Gruppen besteht, wurden auch 'alle' Konflikte gefunden. Um das Skript auf grössere Datensätze anzuwenden, müsste durch alle Gruppen gegangen und im Nachhinein Duplikate entfernt werden. Alternativ könnte man die eine WHERE-Klausel abändern, und zwar anstatt `WHERE m2.represents_group='4' AND m3.represents_group <> '4'`:

```
WHERE m2.represents_group <> m3.represents_group
```

Doch auch damit ergeben sich Duplikate, welche entfernt werden müssen. Gerade um nach statistisch repräsentativen Mustern zwischen Gruppen zu suchen, ist dies unerlässlich. Denn dort würde die Anzahl von Konflikten die Situation nicht naturgetreu wiedergeben, was nicht erwünscht ist. Doch dieser SQL zur Detektion von Gruppenkonflikten innerhalb der Tabelle `MONKEY_LOCATION` zeigt, dass mit räumlichen Datentypen und Queries relativ einfach in der Datenbank nach räumlichen Mustern gesucht werden kann.

Um Trajektorien von Gruppen auf komplexere (Gemeinsamkeits-) Muster zu untersuchen, ist ein grösseres Verständnis von räumlichen SQLs vonnöten. Um die Detektion der *between group encounters* beispielsweise abhängig zur Vegetation zu machen (in Steppen gilt Konflikt falls innerhalb 100m, im Wald 50m), wäre ein ziemlich verschachteltes Skript notwendig. Auch um beispielsweise einen schönen Output mit Start- und Endzeit sowie Start- und Endposition zu generieren, braucht es zusätzliche SQL-Kenntnisse. Sehr gut wäre das automatische Update einer schon vorhandenen `BETWEEN_GROUP_ENCOUNTER`-Tabelle, um eine Vernetzung mit den visuellen Beobachtungen der Forscher zu erstellen. Solche Automatisierungen fordern jedoch sehr fundiertes SQL- und Datenbankwissen.

Mit SQL 8.6 können jedoch schon repräsentative Daten herausgelesen werden. So z.B. welche Gruppen wie oft *encounters* miteinander haben oder, mit ein wenig Mehraufwand, wie lange diese dauern. Überall wo die Forscher auch anwesend waren, können Informationen zum Ablauf des BGEs manuell beigezogen werden. Gründe für die Konflikte können dann z.B. mit dem in 8.4 behandelten Punkt beantwortet werden. Zusätzliche Schlüsse könnten Verwandtschaftsinformation liefern, welche jedoch noch nicht implementiert wurden.

Mit komplexeren Algorithmen wäre die Detektion komplizierterer raum-zeitlicher Muster aus den Trajektorien möglich: 'Folgen sich einzelne Gruppen?', 'Meiden sie sich?', Das Berechnen von *home ranges* oder ähnlichen anderen Massen wäre einiges einfacher, da die Bewegung und damit die Zeitinformation dort keine so grosse Rolle spielt. Für das Berechnen von jahreszeitlich bedingten Territorien können einfach die Punkte einer Gruppe innerhalb einer gewissen Zeitspanne ausgewählt und mit den in Kapitel 3.6.1 angesprochenen Methoden analysiert werden.

## 8.4 Anreicherung mit Kontextinformation

Antworten auf Fragen wie 'In welcher Vegetation ruhen sich die Tiere aus und wo essen sie?', oder 'Hat die Sichtbarkeit einen Einfluss auf das *'resting'*-Verhalten?', sagen viel über das Verhalten der Tiere aus und welchen Einfluss die Umwelt darauf hat. Grundsätzlich wächst der Stellenwert solcher Fragen in der ökologischen Forschung, da auf diese Weise das Verhalten eines Tieres spezifischer untersucht werden kann (siehe Kapitel 3.6.1). Diese Art Fragen können in vielen Fällen durch ein Verschneiden von lokalisierten Beobachtungen (Kapitel 8.2) mit Kontextinformationen beantwortet werden. Eher auf eine

Query umformuliert, würden die oben genannten Fragen in etwa so lauten: 'Wie ist der vorherrschende Vegetationstyp, in welcher die Aktivität 'feeding' ausgeübt wird; oder 'Ist bei niedriger Sichtbarkeit die detaillierte Aktivität von 'resting' eher 'vigilant' und sind die Augen eher geöffnet;

Kontextinformationen, wie die Vegetationskarte, monatliche *food availability*- und *habitat productivity*-Werte, sowie Daten zu den Sichtweiten in den verschiedenen Vegetationen zu unterschiedlichen Jahreszeiten, liegen zum Teil schon vor. Auch stehen zahlreiche Daten zu Beobachtungen und zum Verhalten der Tiere, teils mit räumlichen und oder zeitlichen Informationen zur Verfügung. Der grosse Vorteil eines räumlichen DBMS ist auch hier die Vernetzung der unterschiedlichen Elemente – auch über die Rauminformation. Dies ermöglicht die Beantwortung der im oberen Abschnitt gestellten Fragen nach dem häufigsten Vegetationstyp bei der Aktivität Nahrungsaufnahme ('feeding'), vorausgesetzt die Lokalisierung der regulären Beobachtungen ist wie in Kapitel 8.2 durchgeführt worden. Mit dem SQL 8.7 können nun ausgewählte Daten oder auch der gesamte Datensatz abgefragt werden. Durch die Kontextinformation, welche bei der Durchführung der Aktivität vorherrschte, können Rückschlüsse zum Verhalten im Bezug zur Umwelt gezogen werden.

```
SELECT o.scan_id, o.monkey_id, o.date_and_time, o.group_id, o.activity,
t.veg_type_name, v.poly_id
FROM OBS_SCAN o, vegetation_map v INNER JOIN vegetation_type t
ON v.veg_type = t.veg_type_id
WHERE SDO_ANYINTERACT (v.shape, o.shape)
= 'TRUE'
ORDER BY o.scan_id ASC
```

SQL 8.7: Suche von Kontextinformation bei OBS\_SCAN-Daten je nach Aktivität. Die Resultate werden in Abbildung 8.6 dargestellt.

Da es im Mockup-Datensatz nur vier SCAN-Einträge hat, gibt es auch nur vier Resultate aus der Query (siehe Abbildung 8.6), was nicht gerade Rückschlüsse von der Aktivität zur vorherrschenden Vegetation erlaubt. Doch es zeigt, dass mit einem simplen SQL verschiedene in der Datenbank gespeicherte Geometrien (Punkte und Polygone) miteinander verglichen werden können.

Die Modellierung der Vegetationskarte, wie es in Abbildung 6.4 dargestellt ist, erlaubt zudem die Abfrage nach zeitlich variierender Kontextinformation. Es ist jedoch nicht möglich, eine 'richtige' raum-zeitliche Query zu stellen, wo sich die Jahreszeit und/oder das Jahr der Vegetationskarte sich dem Beobachtungszeitpunkt des SCANS anpasst. Vielmehr muss die WHERE-Klausel noch ergänzt werden, z.B. mit:

```
AND VEGETATION_MAP.season = 'spring2010'
```

Das Problem ist dann jedoch, dass die SCAN-Daten zuerst ausgewählt werden müssen, bevor sie mit der passenden Vegetationskarte verschnitten werden können. Auch dies sollte in der WHERE-Klausel erfolgen, alle Scans vom Frühling 2010 könnten z.B. mit folgendem SQL ausgewählt werden:

```
AND OBS_SCAN BETWEEN to_date('2010/03/21 00:00:00', 'yyyy/mm/dd hh24:mi:ss')
AND to_date('2010/06/20 23:59:59', 'yyyy/mm/dd hh24:mi:ss')
```

Der SQL wird damit zusehends komplizierter und benötigt diverse Zusatzangaben, um korrekt ausgeführt zu werden. Denn das System erkennt nicht selbstständig, dass eine Position zu einem bestimmten Zeitpunkt mit der damals dort herrschenden Kontextinformation verschnitten werden soll.

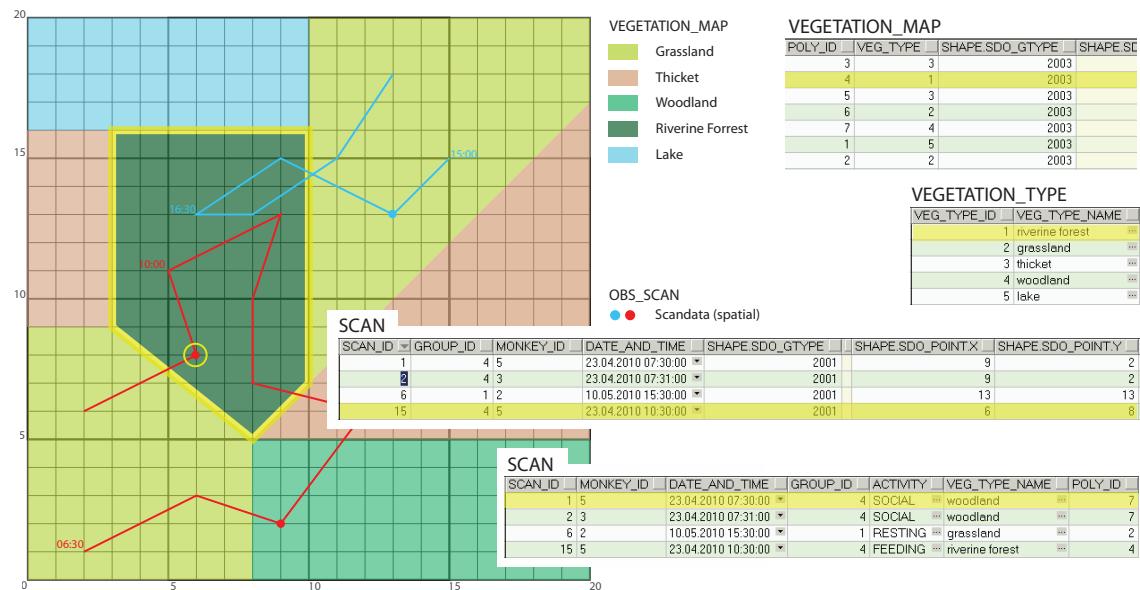


Abbildung 8.6: Kontextinformation bei OBS\_SCAN-Daten. Die Beobachtung mit der SCAN\_ID 15 wird über SQL 8.7 dem Polygon Nr. 4 der VEGETATION\_MAP zugewiesen, welches vom VEG\_TYPE 'riverine forrest' ist. Die Aktivität des Tieres war zum Zeitpunkt der Beobachtung 'social'. Mit mehr Einträgen könnte getestet werden, ob statistisch relevante Rückschlüsse von Aktivität zu Kontext bestehen.

Auch mit der Tabelle MONKEY\_LOCATION kann die Kontextinformation verschnitten werden, es kann ganz allgemein abgefragt werden, wie oft sich Vervet Monkeys in den verschiedenen Vegetationstypen aufhalten:

```
SELECT COUNT(*)
FROM MONKEY_LOCATION m, VEGETATION_MAP v INNER JOIN VEGETATION_TYPE v
ON v.veg_type = t.veg_type_id
WHERE SDO_ANYINTERACT (v.shape, m.shape)
= 'TRUE' AND t.veg_type_name = 'riverine forest';
```

SQL 8.8: Abfrage der MONKEY\_LOCATIONS nach Kontextinformationen

Wenn man mit SQL 8.8 nacheinander die verschiedenen Vegetationstypen durchgeht, so erhält man 15 locations in 'riverine forest', 12 in 'grassland', je 2 in 'thicket' und 'woodland' und nicht überraschend 0 in 'lake'. Diese noch ziemlich generelle Abfrage könnte zusätzlich verfeinert werden, um z.B. abzufragen, in welchem Vegetationstyp sich die meisten Affen zu gewissen Zeitperioden aufhalten.

Ähnliche Arten von Fragen könnte man auch bei ALARM\_CALL, SEX\_BEHAVIOUR, BETWEEN\_GROUP\_ENCOUNTER und WITHIN\_GROUP\_COMPETITION stellen. Aus Forschungsfrage 2 (Kapitel 5.1) wird klar, dass eines der Hauptziele des Inkawu Vervet Projektes das Erkennen der umkämpften Ressourcen von Konflikten zwischen Gruppen ist, um eine Unterscheidung von Konflikt-Kategorien zu ermöglichen. Bei beobachteten encounters zwischen Gruppen werden nicht nur ein, sondern zwei Punkte in Zeit und Raum festgehalten, nämlich den Beginn und das Ende eines Konfliktes. Bei automatisch detektierten between group encounters (Kapitel 8.3) variiert die Anzahl zwischen einem und (in meinem Datensatz) vier Punkten. Durch die Kombination der SQLs 8.6 und 8.7 bekommt man für jeden Punkt eines Konflikts (focal group und encounter group) die Kontextinformation:

```

SELECT "foc_group", "foc_time", "x1", "y1", "foc_veg", "enc_group", "enc_time",
"x2", "y2", "enc_veg", "group_dist"
FROM (SELECT "foc_group", "foc_time", "x1", "y1", "foc_veg", "enc_group",
"enc_time", "x2", "y2", "enc_veg", SDO_GEOM.SDO_DISTANCE("foc_shape",
"enc_shape", 0.005) as "group_dist"
FROM (SELECT represents_group as "foc_group", veg_type_name as "foc_veg",
m.shape as "foc_shape", m.time as "foc_time", m.shape.sdo_point.x as
"x1", m.shape.sdo_point.y as "y1"
FROM monkey_location m, vegetation_map v INNER JOIN vegetation_type t
ON v.veg_type=t.veg_type_id
WHERE SDO_ANYINTERACT(m.shape, v.shape) = 'TRUE'),
(SELECT represents_group as "enc_group", veg_type_name as "enc_veg",
m2.shape as "enc_shape", m2.time as "enc_time", m2.shape.sdo_point.x as
"x2", m2.shape.sdo_point.y as "y2"
FROM monkey_location m2, vegetation_map v2 INNER JOIN vegetation_type t2
ON v2.veg_type=t2.veg_type_id
WHERE SDO_ANYINTERACT(m2.shape, v2.shape) = 'TRUE')
WHERE "foc_group"='4' AND "enc_group" <> '4')
WHERE "group_dist" < '2' --distance is smaller than 2 (m)
AND ("foc_time" - "enc_time") <= '0.020833333' --time difference is <= 30min
AND ("enc_time" - "foc_time") <= '0.020833333' --time difference is <= 30min
ORDER BY "foc_time";

```

SQL 8.9: Detektierung der Gruppenzusammenstöße und zusätzlicher Verschneidung mit der Vegetationskarte, Resultat in Abbildung 8.7

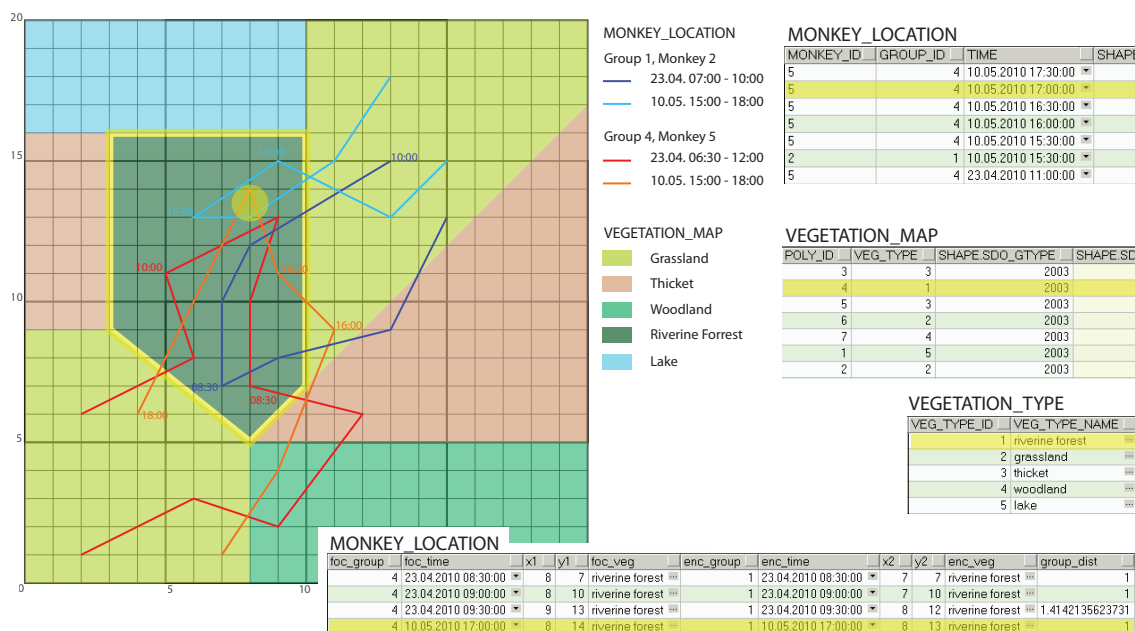


Abbildung 8.7: Resultat vom SQL 8.9, *between group encounter* mit Kontextinformation. Alle Informationen kommen aus den Tabellen **MONKEY\_LOCATION** und **VEGETATION\_MAP**. Es wird nur der kurze Konflikt vom 10. Mai mit Farbe hervorgehoben, welcher sich im Vegetationstyp 'riverine forrest' abspielte. Auf den *encounter* vom 23. April (siehe auch Abbildung 8.5) wird hier nicht weiter eingegangen.

Der Output von SQL 8.9 in Abbildung 8.7 weicht nicht gross vom Output der BGE-Detektion (Abbildung 8.5) ab und besitzt lediglich die Zusatzinformation zum Vegetationstyp. Aus den wenigen *encounters* des Mockup-Datensatzes könnte man nun schliessen, dass sich die Konflikte oft im Vegetationstyp 'riverine forrest' abspielen. Mögliche Gründe

könnten z.B. die reichlich vorhandene Nahrung und die vielen Schutzmöglichkeiten sein, um welche es sich zu kämpfen lohnt. Um statistisch signifikante Aussagen zu machen, werden mehr Daten sowie das Expertenwissen der Forscher benötigt.

Wenn die Tabellen zu Beobachtungen dieser Gruppenkonflikte vorliegen (im ER vorgesehen als `BETWEEN_GROUP_ENCOUNTER`) und zu Start- und Endpunkten aus den *hand-held*-GPS-Geräten bereits *shapes* eingefügt wurden, so funktioniert die Kontext-Suche wie in SQL 8.7, da die räumlichen Objekte bereits bestehen. Liegen nun 'nur' Werte von geographischer Länge und Breite vor, so sollten diese zuerst in *spatial objects* umgewandelt werden, um eine Verschneidung mit der Kontextinformation zu ermöglichen. Auch hier wurde nun die zeitliche Variabilität des Kontextes nicht beachtet und mit dem einzig erstellten Zeitpunkt der Vegetationskarte gerechnet. Wie bereits schon weiter oben in diesem Kapitel behandelt, muss man die zeitliche Variation in Oracle Spatial mit Zusätzen in der WHERE-Klausel lösen.



# Kapitel 9

## Diskussion

In den nächsten Kapiteln werden die in 1.2 gestellten Forschungsfragen sowie die angestrebten Ziele nochmals aufgenommen und gesammelt diskutiert. Dabei werden, während des methodischen Teils der Arbeit, aufgetauchte Schwierigkeiten sowie vorgeschlagene Lösungen aus der Literatur angesprochen. Zum Schluss werden noch einige mögliche raum-zeitliche Erweiterungen und Verbesserungen genannt, welche nicht detailliert bearbeitet wurden. Für das Inkawu Vervet Projekt könnten diese und die für eine allfällige Implementierung nötigen Schritte jedoch von Interesse sein.

### 9.1 Verbesserung des Datenmanagements durch Datenbankentwurf

*Wie kann der generelle Datenbankentwurf einem bereits bestehenden biologischen Forschungsprojekt mit räumlichen, zeitlichen und raum-zeitlichen Fragen helfen, das existierende Datenmanagement zu verbessern?*

Die Ausführung einer detaillierten Anforderungsanalyse hat die vorhandenen Daten sowie die Bedürfnisse des Inkawu Vervet Projektes offen gelegt. Der nachfolgende konzeptuelle Entwurf hat aufgezeigt, wie diese Daten mit einer relationalen Datenbankstruktur abgebildet werden könnten. Dabei wurde der Fokus auf eine stärkere Vernetzung der vorkommenden Daten und eine, mit eingeführten Einschränkungen durch Primärschlüssel und Wertebereiche einhergehende, Verminderung der Redundanz in der Verwaltung der Daten gelegt. Mit der vorgestellten Lösung könnte auch die Fehlerquelle beim Import der Daten reduziert werden, da jeweils nur in Frage kommende Datenwerte, wie beispielsweise vorhandene Gruppen-IDs akzeptiert werden. Die grösste Schwachstelle des bisherigen Datenmanagements im Forschungsprojekt – die fehlende Verknüpfung zu den räumlichen Daten – wurde beim Basis-ER jedoch nicht behandelt. Bei der Modellierung des konzeptuellen Designs wurde die Datenstruktur bewusst nicht zu stark verändert, um in der raum-zeitlichen Anreicherung in Kapitel 8 explizit auf das Potential der Bewegungsdaten einzugehen. Dort wurde gezeigt, wie auch die räumlichen, zeitlichen und raum-zeitlichen Entitäten fest in der Datenstruktur verankert und bei einer Umsetzung vernetzte Abfragen ermöglicht werden könnten.

Eine semantische Anreicherung der Trajektorien, wie z.B. eine Unterteilung in Tagespfade, wurde in 8.1.1 (Abbildung 8.3) kurz angesprochen und wegen geringem Nutzen oder sogar Anhäufung redundanter Informationen wieder verworfen. Mit einer Segmentierung der Trajektorien, wie dies beispielsweise Spaccapietra *et al.* (2008) in ihrem konzeptuellen

Modell (Kapitel 3.5.5) ansprechen, würde eine solch vertieftere Analyse ermöglicht. Für eine Implementierung in einem handelsüblichen DBMS ist dies leider in naher Zukunft noch unrealistisch.

Die in Kapitel 8.1 der raum-zeitlichen Anreicherung vorgeschlagene Datenstruktur mit `MONKEY_LOCATION` als schwacher Entitätstyp von `MONKEY` ist wegen dem zusammengesetzten Primärschlüssel nicht die optimalste Lösung. Die Herstellung einer Beziehung zu einer einzelnen Positionsinformation ist wegen dem Einbezug vom *date*-Datentyp im Primärschlüssel erschwert. Deswegen wurde der *shape* in Kapitel 8.2 kopiert, womit – wie auf Seite 70 angesprochen – redundante Informationen generiert werden. Es wäre deshalb von Vorteil, wenn in `MONKEY_LOCATION` jede Entität über einen einfachen PK angesprochen werden könnte, so wäre es beim oben genannten Beispiel einfach möglich, mit einem Fremdschlüssel auf die spezifische Ortsinformation hinzuweisen, anstatt die ganze *shape*-Information zu kopieren. Damit soll verdeutlicht werden, dass bereits in der Datenstruktur die Weichen für zukünftige Analysen gestellt werden und verschiedenen Optionen gegeneinander abgewägt werden müssen.

Auch die Vorstellung einiger, speziell für die konzeptuelle Modellierung von raum-zeitlichen Anwendungen entwickelten Methoden (Tryfona und Jensen, 2000; Parent *et al.*, 2006; Spaccapietra *et al.*, 2008) zeigt, dass die ersten Schritte eines Datenbankentwurfs von grosser Wichtigkeit sind und eine gut geplante Datenstruktur gerade in raum-zeitlichen oder bewegungsrelevanten Bereichen essenziell ist. Bei dieser Arbeit äusserte sich dies ebenfalls im Zeitumfang, welcher für die Anforderungsanalyse und den konzeptuellen Entwurf in diversen Iterationen benötigt wurde.

## 9.2 ER für Modellierung raum-zeitlicher Komponenten

*Ist das entity relationship model geeignet um die Datenstruktur eines bewegungsökologischen Forschungsprojektes mit räumlichen, zeitlichen und raum-zeitlichen Komponenten abzubilden und dessen Implementierung vorzubereiten?*

Die vorgeschlagene relationale Datenbankstruktur für das Inkawu Vervet Projekt wurde mit Hilfe des *entity relationship models* dargestellt. Die Modellierung mit diesem Werkzeug ist relativ einfach, was den Entwurfsprozess beschleunigt und den Fokus auf die angestrebte Datenstruktur anstatt auf ein komplexeres Modellierungs-Tool gelegt hat. Die leichte Verständlichkeit des ERs war auch in Gesprächen mit der Kontaktperson von grossem Vorteil, da die Elemente ohne Datenbankkenntnisse besprochen werden konnten. Wie Erik Willems meint, könnte das ER im Hinblick auf eine Implementierung zudem ein gutes Kommunikationsmittel zwischen den verschiedenen involvierten Forschungsrichtungen und Universitäten sein.

Für die Modellierung von räumlichen, zeitlichen und raum-zeitlichen Anwendungen ist das *entity relationship model* jedoch nur bedingt geeignet. Räumliche oder zeitliche Attribute können zwar entsprechend, z.B. mit einem Farbcode, hervorgehoben werden (siehe auch Abbildung 6.1), die Darstellung der Möglichkeiten im Raum-Zeit-Bereich ist aber beschränkt. Die Abbildung von räumlichen Beziehungen und Einschränkungen ist im ER nicht möglich, weshalb in Kapitel 8 die vorgeschlagene Datenstruktur implementiert und mit ausgesuchten Beispielen das raum-zeitliche Potential der Daten demonstriert wurde. In dieser 'raum-zeitlichen Anreicherung' wurde die *entity relationship model* -Struktur an gewissen Stellen ergänzt. Nicht mit dem ER visualisierbare Vorgänge wurden mit Skripts



bzw. SQLs, Datenbank-*screenshots* und Skizzen dargestellt, um die Beispiele nachvollziehbar zu machen. Es hat sich gezeigt, dass mit Hilfe dieser Methodik das Potential und Handling dieser raum-zeitlichen Daten durchaus verständlich gemacht werden kann.

Allerdings sind diese SQLs jeweils abhängig von gewählten Datenbanksystem (hier relationales DBMS *Oracle* mit räumlicher Erweiterung *Spatial* und dem räumlichen Datentyp *SDO\_Geometry*). Kaum werden andere Spezifikationen verwendet, können die SQLs nicht mehr 1:1 ausgeführt werden, weshalb diese Methode nicht sehr geeignet ist. Trotzdem bekommt der Leser einen vertieften Einblick in die Möglichkeiten der Daten, und andere Datentypen von weiteren Softwarelösungen, wie beispielsweise *ST\_Geometry* der Open Source Lösung *PostgreSQL / PostGIS*, sind sehr ähnlich und deren Funktionen unterscheiden sich auch nicht gross.

Eine alternative Methode für das Design des konzeptuellen Modells wären spezifisch für solche Anwendungen entwickelte Verfahren wie dasjenige von Tryfona und Jensen (2000). In dieser Erweiterung für das *entity relationship model* können räumliche, zeitliche und raum-zeitliche Aspekte semantisch dargestellt werden. Auch Parent *et al.* (2006) präsentieren mit MADS einen Ansatz für die konzeptuelle Modellierung von traditionellen als auch raum-zeitlichen Anwendungen. Die Stärken dieser Modelle liegen jedoch klar beim konzeptuellen Design, die Konzepte decken sich nicht mit vorhandenen Datenbanktechnologien (Pelekis *et al.*, 2004). Dadurch können zwar komplexere Beziehungen zwischen räumlichen und raum-zeitlichen Objekten dargestellt werden, entsprechende Software für eine Implementierung des erstellten Datenbankmodells fehlt dann aber. Da in dieser Arbeit eine anwendbare Lösung vorgestellt werden sollte, welche in dieser oder einer ähnlichen Art auch vom Inkawu Vervet Projekt implementiert werden könnte, wurde bewusst auf solche Methoden verzichtet. Durch die praxisnahe Modellierung mit dem 'einfachen' bzw. dem *enhanced ER* kann in einem ersten Schritt DBMS-unabhängig gearbeitet und bei raum-zeitlichen Elementen auf DBMS-spezifischen SQLs oder anderen Skripts zurückgegriffen werden. Wie in dieser Arbeit gezeigt wurde, können damit die Möglichkeiten von raum-zeitlichen Daten praxisnah für ein relationales Datenbanksystem dargestellt werden.

### 9.3 Handling von Bewegungsdaten in *spatial relational database management systems*

*Welches sind die Vorteile im Hinblick auf die Speicherung und Analyse von Bewegungsdaten aus der Wildtierforschung mit einem räumlichen, relationalen Datenbankmanagementsystem (spatial relational database management system SRDBMS) gegenüber einem file-basierten Ansatz?*

Rein vom technischen Standpunkt betrachtet, bieten sich im Datenmanagement diverse Vorteile mit einem relationalen Datenbanksystem (Beyer *et al.*, 2010; Urbano *et al.*, 2010). Die in Kapitel 3.4 auf Seite 10ff vorgestellten Anforderungen an Wildtiermanagementsysteme können durch relationale (oder objekt-relationale) DBMS mit geeigneten, räumlichen Erweiterungen theoretisch alle erfüllt werden. Was sehr verlockend erscheint, bedarf jedoch einer ausführlichen Planung und vertieften Kenntnissen im Umgang mit Datenbanken.

Im Erstellen der Beispiel-Datenbank mit ausgewählten Relationen (Kapitel 8) konnte die Erfüllung einiger dieser Anforderungen demonstriert werden. Gerade die geforderte Automatisierung von Arbeitsprozessen (Anforderung 3) kann anhand der Zuweisung der rohen

GPS-Daten an Individuen (Kapitel 8.1) oder der automatischen Ortszuweisung von Beobachtungen (Kapitel 8.2) gezeigt werden. Auch um SCAN-Beobachtungen der nächsten halben Stunde zuzuordnen (siehe Kapitel 7.2), könnte anstelle manueller Zuweisung eines Zeitwertes eine Automatisierung beim Datenimport vorgenommen werden. Damit kann einerseits Zeit gespart, andererseits durch eingefügte Kontrollmechanismen, wie Einschränkungen im Wertebereich oder Primärschlüssel in der Datenbankmodellierung, die Fehlerrate minimiert werden. Mit der Schaffung dieser Datenkonsistenz können die gemachten Daten über Jahre problemlos abgefragt werden und sind für Langzeitbeobachtungen geeignet (Anforderung 2).

Die Integration verschiedenster Daten (Punktdaten von regelmässigen GPS-Positionen der Tiere, vektorbasierte Kontextinformationen, grundlegende Daten wie z.B. vorhandene Individuen oder Gruppen, Beobachtungsdaten usw.) wurde veranschaulicht (Anforderung 9). Mit einem gut ausgearbeiteten Datenmodell können die verschiedenen Elemente aus diversen Datenquellen miteinander verbunden werden und bieten damit einen höchst vernetzten Datensatz, welcher jegliche Analysemöglichkeiten erlaubt. Durch diese Beziehungen zwischen den einzelnen Relationen kann ausserdem die Redundanz der Daten verkleinert werden, da Bezüge untereinander bestehen und diese nicht kopiert werden müssen.

Durch die Verwendung von räumlichen Datentypen und die damit verbundene Einkapselung der Geometrie werden räumliche Abfragen mit vordefinierten Operatoren in der Query-Sprache bereits im DBMS ermöglicht. Wegen den eingebetteten räumlichen Indizes können – wie in Kapiteln 8.3 und 8.4 gezeigt – auch Abfragen zu Geometrie und Topologie auf der Datenbank ausgeführt werden (Anforderungen 4 und 5, mehr dazu im nächsten Kapitel). Dies wurde in der *Oracle*-Datenbank anhand des Testdatensatzes demonstriert, wo die präsentierten SQLs ausführbar sind und die gewünschten Resultate liefern. Kapazität (Anforderung 1) und Effizienz waren in dieser Arbeit nicht gefragt und wurden auch nicht untersucht, theoretisch ist die Kapazität eines relationalen Datenbanksystems jedoch fast unlimitiert.

Bereits in den rohen GPS-Daten befinden sich zwei Zeitwerte und beide könnten mit einem angepassten Import-SQL in die Datenbank aufgenommen werden (Anforderung 6). Dies wird im Projekt jedoch als nicht wichtig erachtet und hier weggelassen. Im erstellten Datenbanksystem ist die Bearbeitung der Daten durch mehrere Nutzer möglich (Anforderungen 10 und 11), was für ein internationales Forschungsprogramm wie das Inkawu Vervet Projekt von grosser Bedeutung ist. Auch die Anbindung an ein Geographisches Informationssystem für einfachere Analysen wäre auch durchführbar. Dies wurde jedoch nicht getestet, wobei die Kombination von GIS mit DBMS eigentlich das ideale Informationssystem wäre (Urbano *et al.*, 2010).

Die Vorzüge von Datenbanken zeigen sich auch bei den im Moment existierenden Informationssystemen zur Speicherung und Analyse von Wildtierdaten wie z.B. ISAMUD (*Information System for Analysis and Management of Ungulate Data*), WRAM (*Wireless Remote Animal Monitoring*) oder *Movebank*. Diese sind alles relationale oder objektrelationale Datenbankmanagementsysteme mit zweckbestimmten räumlichen Tools (siehe auch Kapitel 3.7.2). Dies verdeutlicht die Stärken der relationalen Datenbanklösungen und unterstreicht die in dieser Arbeit gewählte Lösungsvariante mit *Oracle Spatial* als relationales DBMS mit räumlicher Erweiterung. Anforderung 13, Kosteneffektivität, kann mit meiner Datenbanklösung nicht erreicht werden, da Lizenzen von *Oracle Spatial* (ausser für Lehrzwecke) sehr teuer sind. Mit den vorhin erwähnten Lösungen können alle Anforderungen ebenfalls erfüllt werden, wobei die Software grösstenteils aus Open Source besteht und damit auch Anforderung 13 erfüllt werden kann.

## 9.4 Mögliche raum-zeitliche Fragen mit einem handelsüblichem *spatial relational database management system*

*Welche für ein ökologisches Forschungsprojekt interessanten raum-zeitlichen Fragen betreffend Bewegungen der Tiere können mit Hilfe eines handelsüblichen, räumlichen, relationalen Datenbankmanagementsystems (spatial relational database management system SRDBMS) untersucht werden?*

*Oracle Spatial* ist ein relationales DBMS mit räumlicher und nicht raum-zeitlicher Erweiterung. Es beinhaltet räumliche sowie zeitliche Datentypen, welche als voneinander getrennte Attribute agieren. Das bedeutet, dass die Raum- und Zeit-Attribute in Queries mit entsprechenden Operatoren durch separaten Bedingungen adressiert werden müssen. Bei zeitlich variierendem Kontext kann der passende Stand nicht durch einen einzigen Operator identifiziert werden, sondern muss in der Query ausdrücklich ausgewählt werden (siehe SQL 8.7 und folgende).

Um dies zu vermeiden und die Analyse zu vereinfachen, wird in ISAMUD jeder *fix* bereits beim Import in die Datenbank mit den zeitlich variierenden Kontextinformationen (*landcover*, Wetterdaten, Tag/Nacht-Zeiten, ...) verschnitten. Diese Lösung, auch wenn für spätere Untersuchungen sehr praktisch, generiert redundante Informationen, was eigentlich vermieden werden sollte. Doch es ist ein gangbarer Ausweg, um die Schwächen von derzeitigen Softwarelösungen zu umgehen.

In *Oracle* wird ausserdem die zeitliche Komponente nur dürftig unterstützt, es gibt zwar den Datentyp *date*, jedoch keine vordefinierten zeitlichen Operatoren wie die von Allen (1983) vorgestellten *after*, *during* usw. Um die Differenz zwischen zwei Entitäten zu ermitteln, müssen deren Zeitattribute jeweils subtrahiert und allenfalls zusätzlich die Vorzeichen entledigt werden. Es sind keine intuitiven Abfragen wie beispielsweise `TIMEDIFF is MIN(SCAN, MONKEY_LOCATION)` möglich und diese müssen mit umständlichen (und nicht 'perfekten') Lösungen umgangen werden (siehe z.B. SQL 8.5 in Kapitel 8.2). Unter diesen Umständen werden die in Kapitel 5.4 entworfenen raum-zeitlichen Abfragen zum Inkawu Vervet Projekt nach der Klassifikation von Yuan und McIntosh (2002) nachfolgend auf ihre mögliche Anwendung im erstellten Datenbanksystem untersucht:

Attributabfragen sind grundsätzlich immer lösbar, sie benötigen keine speziellen räumlichen oder zeitlichen Attribute beziehungsweise Operatoren und könnten auch in einem nicht-räumlichen DBMS ausgeführt werden. Räumliche Abfragen sind mit *Oracle* über Queries zufriedenstellend zu beantworten, da Buffer, Verschneidungen und diverse andere räumliche Funktionen relativ gut unterstützt werden. Zeitliche Abfragen sind wie bereits schon weiter oben erwähnt ein wenig mühsamer, da das Handling mit dem Datentyp *date* umständlich ist. Doch grundsätzlich ist auch hier die Beantwortung aller Query-Typen möglich, auch wenn die Formulierung der Abfragen teilweise gewöhnungsbedürftig ist. Einfache raum-zeitliche Abfragen, wo jeweils von der Raum- auf die Zeitinformation (oder umgekehrt) geschlossen wird, sind gut möglich, auch wenn die spezifischen Attribute getrennt gespeichert sind. Raum-zeitliche Verhaltensabfragen, wo beispielsweise innerhalb einer Zeitperiode die Positionsveränderung eines einzelnen Objektes untersucht werden (z.B. *home range* einer Gruppe im Sommer oder Winter), sind durchführbar. Eventuell müssen diese Abfragen über mehrere SQLs, mit Hilfe von *views* oder dann mit stark verschachtelten SQLs gelöst werden. Raum-zeitliche Verhaltens- und Beziehungsabfragen stellen wegen der höchsten Komplexität die grösste Schwierigkeit dar. Solche Abfragen sind fast nicht mehr in einem Schritt zu bewerkstelligen, da sie unweigerlich mehrere, in sich verschachtelte Unterabfragen (*subqueries*) nachziehen. Es ist jeweils einfacher, diese in

mehreren Schritten durchzuführen. Die Detektion von Gruppenzusammenstößen (Kapitel 8.3, SQL 8.6) und deren Anreicherung mit Kontext (Kapitel 8.4, SQL 8.9) zeigen, dass auch solch komplexere, raum-zeitlichen Queries in einem handelsüblichen DBMS per SQL möglich sind. Man sollte jedoch auch anmerken, dass sich ein GIS für solche Analysen und vor allem deren Visualisierung besser eignen würde.

## 9.5 Zielerreichung und Einschränkungen

Grundlegend muss festgehalten werden, dass der Zeitplan meiner Arbeit sich nicht gerade mit dem des Inkawu Vervet Projektes deckte und deshalb gewisse Einschränkungen von meiner Seite gemacht werden mussten. Die Durchführung der Anforderungsanalyse stellte insofern eine Herausforderung dar, als dass sie schwierig einzuschätzen war. Zeitlich zog sie sich in die Länge und während dem Erstellen des konzeptuellen Modells tauchten immer wieder Fragen auf. Im Voraus alle beabsichtigten Datenbanktransaktionen sowie alle Eventualitäten zu erfahren ist sehr schwierig und braucht wohl einiges an Datenbankerfahrung. Gerade das bereits bestehende Datenmanagement zu verstehen, ist wegen der Vielzahl an Prozessen, welche von der Datenerfassung bis zur Analyse ablaufen, sehr schwierig. Eine Anforderungsanalyse nach dem *Contextual Design*, wo die Datenbankentwickler die Abläufe im bestehenden Projekt durchmachen (Elmasri und Navathe, 2009), wäre in meinem Falle sicher sehr hilfreich gewesen. Die Zusammenarbeit mit dem Inkawu Vervet Projekt war mit Erik Willems zwar auf eine Person beschränkt, doch als Koordinator des Forschungsprojektes hatte er den wohl besten Überblick über das gesamte Projekt. Während der Anforderungsanalyse wurden auch die raum-zeitlichen Abfragen erstellt, welche in den Gesprächen mit der Kontaktperson als Anhaltspunkte dienten. Dies war eine gute Methode um festzustellen, welche Daten in welcher Form in die Datenbank implementiert werden sollten. Das erstellte ER widerspiegelt in seiner Endform eine Möglichkeit, wie die aktuell vorhandenen Daten in einem relationalen Modell wiedergegeben werden könnten, mit dem Aufzeigen der räumlichen, zeitlichen und raum-zeitlichen Informationen wurden Möglichkeiten hervorgehoben. Der Mehrwert für das Projekt liegt also nicht unbedingt im *entity relationship model* allein, sondern in der Kombination von ER und der raum-zeitlichen Anreicherung. Die relationale Datenbank wurde aufgrund meiner Vorkenntnisse mit *Oracle* und deren räumlichen Erweiterung *Spatial* erstellt. Es wurden darin nur Relationen für die Demonstration der Beispiele zur raum-zeitlichen Anreicherung erstellt, wobei diese mit einem sehr kleinen Datensatz gefüttert wurden. Die SQLs konnten also nicht explizit auf ihre Effizienz getestet werden. Bei einem grossen Datensatz, wie dies in der Bewegungsökologie wegen den tausenden von GPS-*fixes* üblich ist, würden stark verschachtelte Queries, wie z.B. derjenige für die Detektion der Gruppenzusammenstösse mit Kontextinformation (SQL 8.9), wohl einiges langsamer funktionieren.

## 9.6 Empfehlungen zur Implementation im Inkawu Vervet Projekt

Als weitere, in Kapitel 8 nicht besprochene, raum-zeitliche Anreicherung könnte die doch teils ziemlich grosse zeitliche Unsicherheit von gewissen Ereignissen festgehalten werden. Derzeit werden über den Log jeweils Änderungen in der Gruppenkomposition als auch Geburten und Tode von Individuen vermerkt und mit dem Zeit- bzw. Datumstempel vom LOG versetzt (siehe auch Abbildung 6.2). Eine Möglichkeit, wie sie auch in der ISAMUD-Struktur (Cagnacci und Urbano, 2008; Urbano *et al.*, 2010) angewendet wird, wäre bei

Geburten den Zeitstempel als *latest possible birth* zu speichern und über ein Skript (mit der Gruppeninformation und dem automatischen Zeitstempel aus LOG) den Zeitpunkt des letzten Logs herauszufinden und mit einem Tag addiert als *first possible birth* einzufügen. Genauso könnte auch mit den Veränderungen in der Gruppenzusammensetzung (siehe Kapitel 6.1) verfahren werden.

Momentan wird für die Beobachtungen von unregelmässig auftretenden Ereignissen jeweils mit einem separaten *handheld*-GPS-Gerät eine Lokalisierung vorgenommen. Diese muss in der Datenbank im Nachhinein manuell der Beobachtung zugewiesen werden, damit eine Position für die Beobachtung vorhanden ist. Es wäre jedoch wie für die regelmässigen GPS-Daten möglich (Kapitel 8.1), eine Automatisierung dieses eher mühsamen Importverfahrens zu erstellen. Dazu schlage ich folgende Datenstruktur vor: Beim Ausfüllen vom LOG wird jeweils vermerkt, welche `HANDHELD_GPS_ID` verwendet wird (in Abbildung 9.1 Beziehung LOG zu `HANDHELD_GPS_DEVICE`). Wenn nun beispielsweise ein Alarmruf bemerkt wird, soll neben dem Ausfüllen des `ALARM_CALL`-Formulars gerade noch schnell mit dem *handheld*-GPS-Gerät ein *fix* gemacht werden, wie dies auch jetzt bereits durchgeführt wird. Beim Import der Daten kann über die Gruppeninformation und den automatisch ausgefüllten Zeitstempel der zugehörige LOG und darin die benutzte `HANDHELD_GPS_ID` gefunden werden. Als nächstes soll dann derjenige *fix* aus `HANDHELD_GPS_LOCATION` in der `ALARM_CALL`-Relation eingefüllt werden, welcher den 'ungefähr gleichen' Zeitstempel besitzt. Der SQL wird sicher nicht einfach, da neben einigen *subqueries* Umrechnungen mit dem *date*-Datentyp notwendig sind, die SQLs 8.1 und 8.5 sollten dabei aber behilflich sein.

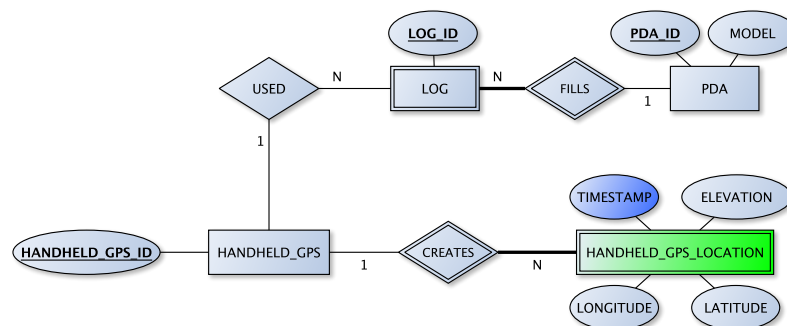


Abbildung 9.1: ER-Struktur für automatische Zuweisung der *handheld*-GPS-Daten.

Die Übernahme der Punktgeometrien aus `MONKEY_LOCATION` für `FOCAL` oder die auftretenden Ereignisse (*OnOccurrence*) mit Hilfe der Gruppen- und Zeitinformation (Kapitel 8.2) würde besser anhand einer Interpolation vorgenommen, wie Erik Willems anmerkte. Auch wenn 'nur' mit einem linearen Bewegungsmodell (Gerade zwischen einzelnen *fixes*) gearbeitet werde, so könnte die Fehlerabweichung von der tatsächlichen Position verkleinert werden. Die Entwicklung einer besseren Methode für die automatische Positionsanreicherung wäre auch für die `OBS_SCAN-records` anzustreben.

Die in Kapitel 8 und gerade eben angesprochenen Automatisierungen wären mit Hilfe von Triggers wahrscheinlich fast noch besser zu lösen, z.B. würde beim Import von neuen Daten in `RAW_GPS_DATA` automatisch der Prozess von SQL 8.1 ausgeführt werden. Ein solcher Datenbanktrigger könnte beispielsweise in dieser Form beginnen `CREATE OR REPLACE TRIGGER data_import, AFTER INSERT ON GPS_RAW_DATA, FOR EACH ROW ...` und automatisch weitere Triggers oder SQLs auslösen. Es sind ganze Ketten von automatischen Abläufen mit integrierten Kontrollmechanismen möglich, was sich bei immer wiederholen-

den Prozessen wie z.B. dem Datenimport sicher lohnen würde.

Die im Inkawu Vervet Projekt reichlich vorhandenen Kontextinformationen im Rasterformat wurden in der raum-zeitlichen Anreicherung nicht einbezogen. Ein nächster Schritt wäre sicher eine Integration dieser Daten, denn auch diese können indiziert, mit den weiteren Geometrien verschnitten und abgefragt werden, sowohl mit *Oracle Spatial* (*GeoRaster*<sup>1</sup>) als auch mit *PostgreSQL & PostGIS* (*WKT Raster*) (Obe und Hsu, 2011). Ebenso der Einbezug von Daten aus Experimenten und genetischen Analysen kann in der vorgeschlagenen relationalen Struktur verwirklicht werden. Gerade die Daten zu den genetischen Analysen sind jedoch sehr umfangreich und komplex mit Verlinkungen untereinander, weshalb das Tabellendesign im voraus gut abgeklärt werden sollte.

Für den Fall, dass die vorgeschlagene Datenstruktur vom Inkawu Vervet Projekt in dieser oder ähnlicher Form übernommen werden möchte, fallen weitere, im Bericht nicht besprochene Schritte an. Die Wahl des Datenbankmanagementsystems (DBMS) beeinflusst den weiteren Vorgang und sollte gut überdacht werden. Die Forschung an *moving objects databases* ist noch relativ jung und die vorgestellten MOD-Prototypen benötigen vertiefte Programmierkenntnisse sowie eine lange Einarbeitungszeit. 'Wahre' raum-zeitliche oder *moving objects* DBMS werden in naher Zukunft nicht auf dem Markt erscheinen, daher müssen sich biologische und ökologische Forschungsprojekte weiter auf räumliche DBMS ausrichten. Wegen eingeschränktem Budget würde eine Kombination von *Oracle Spatial* und *ArcGIS* eher nicht in Frage kommen, weshalb im Open Source Bereich nach Lösungen gesucht werden sollte. Die in Kapitel 3.7.2 vorgestellte Softwareplattform ISAMUD kann komplett mit frei zugänglichen Programmen erstellt werden und erfüllt die Anforderungen an ein Wildtier-Managementsystem (Kapitel 3.4) ebenfalls. Die Basis bilden das relationale DBMS *PostgreSQL* und die räumliche Erweiterung *PostGIS*, worauf die bereits im Inkawu Vervet Projekt verwendeten *R*-Skripts (evtl. mit ein paar kleinen Änderungen) ebenfalls ausgeführt werden können. Das Erstellen und Management der Datenbank kann entweder mit der Open Source Software *pgAdmin iii* oder auch mit dem bereits verwendeten *Microsoft Access* erledigt werden. Als Hilfe für eine Implementierung der ISAMUD-Struktur befinden sich in Obe und Hsu (2011) zahlreiche Tipps bezüglich der Open Source Lösungen *PostgreSQL* und *PostGIS* sowie auch SQLs für die Analyse. Die modulare Struktur von ISAMUD erlaubt es, einfachere Queries für grosse Datenmengen direkt auf der Datenbank abzusetzen und komplexere Analysen mit Visualisierungen beispielsweise mit *R* oder einem GIS durchzuführen. Die Möglichkeit zum Internet-Zugriff (*phpPgAdmin*, *MapServer*) auf die Datenbank ist auch gegeben, was gerade bei einem solch internationalen Forschungsprojekt von Vorteil sein kann.

Nach der Wahl des DBMS ist der nächste Schritt die Erstellung der physischen Datenbank aus dem konzeptuellen sowie logischen Modell und der Import der bereits vorhandenen Daten in das erstellte Datenbanksystem. Der Prozess des Datenimports sollte nicht unterschätzt werden, da diese teils in einer anderen Form vorliegen. Grundsätzlich ist der automatisierte Import von Daten in diversen Formaten möglich, wegen Primärschlüssel- und Fremdschlüsselbeziehungen muss jedoch auf die Reihenfolge geachtet werden. Das Ziel wären einheitliche und redundanzfreie Daten, welche über simple SQLs bis in die Anfänge des Projektes zurück abgefragt werden können. Die Verbindung mit einem GIS für komplexere Analysen und Visualisierungszwecke wäre empfehlenswert, obwohl mit dem bereits verwendeten *R* auch Darstellungen erstellt werden können. Allenfalls sollten für zukünftig einfachere Importprozedere der Daten aus dem Feld allenfalls die Formulare der PDAs und die Import-Skripts verändert werden.

---

<sup>1</sup>[http://docs.oracle.com/html/B10827\\_01/geor\\_intro.htm](http://docs.oracle.com/html/B10827_01/geor_intro.htm) 21.01.2012

# Kapitel 10

## Schlussfolgerung

Die Arbeit wird abgeschlossen, indem die durchgeführten Schritte geschildert und der wissenschaftliche Beitrag hervorgehoben wird. Im Ausblick sollen weitere raum-zeitliche Anreicherungsmöglichkeiten, die für eine Implementierung anfallenden Schritte, sowie denkbare Zukunftsaussichten in diesem Forschungsbereich genannt werden.

### 10.1 Zusammenfassung

Beim Management und der Analyse von Bewegungsdaten ist die Lücke zwischen der theoretischen Forschung und in der Praxis angewandten Softwarelösungen noch ziemlich gross. Raum-zeitliche oder *moving objects* Datenbanken existieren bis jetzt nur als Prototypen. Um sich bewegende Objekte effizient zu speichern und abzufragen muss auf kommerzielle Softwarepakete zurückgegriffen werden. Handelsübliche relationale Datenbankmanagementsysteme mit räumlichen Erweiterungen und zweckbestimmten Tools eignen sich am besten für die Forschung von Wildtierdaten (Urbano *et al.*, 2010; Beyer *et al.*, 2010).

Für ein bewegungsökologisches Forschungsprojekt wurde deshalb ein standardisierter Datenbankentwurfsprozess durchgeführt und die jetzige Datenstruktur auf räumliche, zeitliche und raum-zeitliche Komponenten untersucht. Mit einem handelsüblichen, relationalen Datenbanksystem (*Oracle Spatial*) wurde anhand eines kleinen Datensatzes aufgezeigt, wie verschiedene, im Projekt vorkommende Prozesse verbessert werden könnten. Es wurde eine Automatisierung des Importverfahrens von rohen GPS-Daten vorgeschlagen, wobei die Angaben zu geographischer Länge und Breite in räumliche Datentypen (*spatial data types*) umgewandelt werden, was eine Einkapselung der Geometrie bewirkt. Dadurch können mit räumlichen Operatoren und vorhandenen räumlichen Indizes die Daten direkt auf der Datenbank beispielsweise zu Topologie abgefragt werden. Dank zusätzlichen zeitlichen Attributen der Objekte konnten auch kompliziertere raum-zeitliche Abfragen beantwortet und so Zusammenstösse von verschiedenen Gruppen per SQL detektiert werden. Eine weitere Analysemöglichkeit wurde im Verschneiden von verschiedenen Geometrietypen demonstriert, indem Ereignisse in Raum und Zeit (Punkte) mit der entsprechenden Kontextinformation (Polygonen) versetzt wurden. Allgemein wurden die einzelnen Elemente in der Datenbank stärker vernetzt, was sogar eine Lokalisierung von ursprünglich nicht räumlichen Daten ermöglichte. Sofern die Daten denn vorhanden sind, können nun die meisten, von den Forschern gewünschten raum-zeitlichen Fragen direkt in der Datenbank beantwortet werden.

## 10.2 Wissenschaftlicher Beitrag

Mit der Durchführung einer Anforderungsanalyse und dem Erstellen des konzeptuellen Datenentwurfs für das Inkawu Vervet Projekt konnte aufgezeigt werden, wie die Datenstruktur für bewegungsökologische Forschungsprojekte mit dem *entity relationship model* nur bedingt wiedergegeben werden kann. Gerade für räumliche Elemente und topologische Beziehungen werden zusätzliche Strukturen benötigt, um die Möglichkeiten einer zukünftigen Datenbank darzustellen.

Bewegungsdaten, wie sie mit GPS-Empfängern erfasst werden – also als *fixes* mit  $x,y,(z)$  und einem Zeitwert – können in einem relationalen DBMS zweckmässig gespeichert werden. Die Implementierung mit *Oracle Spatial* hat gezeigt, dass die Umwandlung der rohen GPS-Daten aus dem Feld in räumliche Datentypen mit einer geschickten Datenstruktur automatisch ablaufen kann. Dabei können Kontrollmechanismen für konsistente und möglichst fehlerfreie Daten sorgen und zusätzlich wird der Arbeitsaufwand der Forscher massiv verkleinert.

Mit handelsüblichen relationalen DBMS und deren räumlichen Datentypen können viele raum-zeitliche Abfragen zu Bewegungsdaten direkt über SQLs zu beantwortet werden. Obwohl nicht explizit für Bewegungsdaten vorgesehen, können auch komplexere Queries abgesetzt und beantwortet werden. Sogar Fragen beispielsweise zu raum-zeitlichen Beziehungen nach der Klassifikation von Yuan und McIntosh (2002), können – wenn auch mit stark verschachtelten Queries und vielen Bedingungen in der WHERE-Klausel – untersucht werden.

Für das Inkawu Vervet Projekt wurde mit dem konzeptuellen und logischen Datenbankentwurf eine Übersicht der vorhandenen Daten sowie einige Verbesserungen in einem relationalen Datenmodell präsentiert. Es wurde eine stärkere Vernetzung der einzelnen Elemente sowie Einschränkungen mit Primärschlüsseln und Wertebereichen angestrebt, um die Redundanz der Daten zu verkleinern. Das Potential der Bewegungsdaten sowie der räumlichen Datentypen aus handelsüblichen DBMS konnte anhand einiger Beispiele zu raum-zeitlicher Anreicherung demonstriert werden. Damit wurde auch ein mögliches Umsteigen auf eine räumliche, relationale Datenbank vorbereitet und mit ISAMUD eine Plattform vorgestellt, welche im Moment die wohl beste Lösung für das Management und die Analyse von Bewegungsdaten aus der Wildtierforschung darstellt.

## 10.3 Ausblick

Positionierungstechnologien werden auch in naher Zukunft grosse technische Fortschritte machen, hin zu immer kleineren GPS-Empfängern, welche mit weniger Batterieverbrauch in kürzerer Zeit noch genauere Ortungen ermöglichen (Tomkiewicz *et al.*, 2010). Daraus entstehen räumlich sowie zeitlich immer feiner aufgelöste Bewegungsdaten und stets wachsende Datenmengen. Dadurch werden keine Modelle von Tierbewegungen mehr benötigt, wo zwischen räumlich und zeitlich weit auseinanderliegenden *fixes* irgendwelche Bewegungen angenommen werden müssen, sondern es entstehen hoch aufgelöste *trajectories*. Durch diese zunehmende 'Kontinuierisierung' der Daten wird der Fokus der Analyse weg von den einzelnen Punkten stärker in Richtung Pfadanalyse gehen (Kie *et al.*, 2010).

In der Forschung von Wildtieren zeigt sich bei den Analysewerkzeugen ein Trend in Richtung von frei zugänglichen Softwarelösungen, zahlreiche Algorithmen werden z.B. mit *R* geschrieben, durch Fachleute geprüft und verfügbar gemacht. Auch im Bereich des Datenmanagements ist dieser Trend ersichtlich, mit ISAMUD (Kapitel 3.7.2) wird beispielsweise eine modulare Plattform vorgestellt, welche die besten Open Source Programme verbindet



und deren Stärken ausnutzt. Mit einer geschickten Datenstruktur wird eine Standardisierung angestrebt, wie Bewegungsdaten aus der Wildtierforschung gespeichert werden sollten. Eine solche, auf internationalen Datenstandards beruhende Plattform könnte die von diversen Forschern (Kranstauber *et al.*, 2011; Cagnacci *et al.*, 2010) geforderte Vereinigung und Vermischung der Daten ermöglichen und die Interoperabilität in der nahen Zukunft gewährleisten (Beyer *et al.*, 2010). Vor allem auch wegen den nicht vorhandenen Einstiegskosten kann sich eine solche Datenmanagementlösung in der wissenschaftlichen Gemeinschaft verbreiten und mit der einhergehenden Standardisierung für einen einfacheren Datenaustausch sorgen. Mit einer einheitlichen Datenstruktur sowie standardisierten Analysemethoden können vergleichbare Resultate erzielt und damit neue Einsichten in globale, ökologische Prozesse gewonnen werden. In der immer stärker vernetzten wissenschaftlichen Gemeinschaft wird auch Webzugriff auf solche Daten in Zukunft von grosser Wichtigkeit sein. *Web processing services* (WPS) sind ein viel versprechender Ansatz, um Algorithmen ohne spezifische Software auf dem eigenen Computer über das Internet durchzuführen und die Resultate mit diversen Outputs zu analysieren (Beyer *et al.*, 2010). Eine Auslagerung der Daten und Software – wie sich das auch im aktuellen Trend von *cloud computing* zeigt – könnte auch in der Naturwissenschaft Einzug halten.

Der Bedarf nach einem für Wildtierdaten optimierten Managementsystem zur Speicherung und Analyse ist gross, um dies zu erreichen, sollten die Forscher aus den verschiedenen Disziplinen GIScience, Datenbank und Ökologie vermehrt zusammenarbeiten. Davon wird nicht nur die bewegungsökologische Forschung, sondern auch die GIScience profitieren: Dodge *et al.* (2008) nennen fehlende Versuchsdaten als einen der Gründe für das limitierte Vorhandensein von implementierten und in der Praxis anwendbaren Analysemethoden. Durch die Verwendung von realen Bewegungsdatensätzen aus der Wildtierforschung könnten solche Methoden getestet und wiederum in der Ökologie (und anderen Bereichen) angewendet werden. Mit dem Gebrauch von Wildtierdaten gibt es auch keine Probleme mit dem persönlichen Datenschutz, da dies im Fall von Tieren weniger von Belang ist. Auch von einer Zusammenarbeit mit der Datenbankforschung kann die GIScience profitieren, da sich die Geographische Informationsgemeinschaft auf Grund der allgemein immer grösser werdenden Datensätze gezwungenermassen vermehrt mit Datenbanken auseinandersetzen muss.



# Literaturverzeichnis

- Allen, J.F., 1983. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26, 832–843.
- Andrienko, N. und Andrienko, G., 2007. Designing visual analytics methods for massive collections of movement data. *Cartographica*, 42.
- Andrienko, N.V., *et al.*, 2008. Basic Concepts of Movement Data. *In: F. Giannotti und D. Pedreschi, Hrsg. Mobility, Data Mining and Privacy - Geographic Knowledge Discovery*. Springer, 15–38.
- Armenakis, C., 1992. Estimation and Organization of Spatio-Temporal Data. *In: Proceedings of the Canadian Conference on GIS92*, Ottawa.
- Baglioni, M., *et al.*, 2009. Towards Semantic Interpretation of Movement Behavior. *In: AGILE Conf. 2009*, 271–288.
- Beyer, H.L., Jenness, J., und Cushman, S.A., 2010. Components of Spatial Information Management in Wildlife Ecology: Software for Statistical and Modeling Analysis. *In: S. Cushman, F. Huettmann und S. Cushman, Hrsg. Spatial Complexity, Informatics, and Wildlife Conservation*. Springer.
- Bogorny, V., Heuser, C.A., und Alvares, L.O., 2010. A conceptual data model for trajectory data mining. *In: Proceedings of the 6th international conference on Geographic information science*, GIScience'10, Zurich, Switzerland Berlin, Heidelberg: Springer-Verlag, 1–15.
- Cagnacci, F., *et al.*, 2010. Animal ecology meets GPS-based radiotelemetry: a perfect storm of opportunities and challenges. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365 (1550), 2157–2162.
- Cagnacci, F. und Urbano, F., 2008. Managing wildlife: A spatial information system for GPS collars data. *Environmental Modelling and Software*, 23 (7), 957 – 959.
- Calenge, C., Dray, S., und Royer-Carenzi, M., 2009. The concept of animals' trajectories from a data analysis perspective. *Ecological Informatics*, 4 (1), 34 – 41.
- Dodge, S., Weibel, R., und Lautenschutz, A.K., 2008. Towards a taxonomy of movement patterns. *Information Visualization*, 7 (3-4), 240–252.
- Egenhofer, M.J. und Herring, J., 1994. Categorizing binary topological relations between regions, lines, and points in geographic databases. *Tech. Report, Department of Surveying Engineering*, 9, 1–28.

- Elmasri, R. und Navathe, S.B., 2009. *Grundlagen von Datenbanksystemen*. 3. aktualisierte Bachelorausgabe München; Boston [u.a.] : Pearson Studium.
- Fieberg, J., *et al.*, 2010. Correlation and studies of habitat selection: problem, red herring or opportunity?. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365 (1550), 2233–2244.
- Frank, A.U., 2003. Chapter 2: Ontology for Spatio-temporal Databases. In: M. Koubarakis, Hrsg. *Spatio-Temporal Databases: The CHOROCHRONOS Approach.*, Vol. 2520 Springer, 9–77.
- Grenon, P. und Smith, B., 2004. SNAP and SPAN: Towards Dynamic Spatial Ontology. *Spatial Cognition and Computation*, 4 (1), 69 – 104.
- Güting, R.H., Behr, T., und Düntgen, C., 2010. SECONDO: A Platform for Moving Objects Database Research and for Publishing and Integrating Research Implementations. *IEEE Data Eng. Bull.*, 33 (2), 56–63.
- Güting, R. und Schneider, M., 2005. *Moving objects databases*. Morgan Kaufmann series in data management systems Morgan Kaufmann.
- Hebblewhite, M. und Haydon, D.T., 2010. Distinguishing technology from biology: a critical review of the use of GPS telemetry data in ecology. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365 (1550), 2303–2312.
- Heller, M., 1990. *The ontology of physical objects: four-dimensional hunks of matter*. Cambridge studies in philosophy Cambridge University Press.
- Hornsby, K. und Egenhofer, M.J., 2000. Identity-based change: a foundation for spatio-temporal knowledge representation. *International Journal of Geographical Information Science*, 14 (3), 207–224.
- Jones, M.B., *et al.*, 2006. The New Bioinformatics: Integrating Ecological Data from the Gene to the Biosphere. *Annual Review of Ecology Evolution and Systematics*, 37 (1), 519–544.
- Khatri, V., Ram, S., und Snodgrass, R.T., 2004. Augmenting a Conceptual Model with Geospatiotemporal Annotations. *IEEE Transactions on Knowledge and Data Engineering*, 16, 1324–1338.
- Kie, J.G., *et al.*, 2010. The home-range concept: are traditional estimators still relevant with modern telemetry technology?. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365 (1550), 2221–2231.
- Kranstauber, B., *et al.*, 2011. Software, Data and Modelling News: The Movebank data model for animal tracking. *Environ. Model. Softw.*, 26, 834–835.
- Kuijpers, B. und Othman, W., 2010. Trajectory databases: Data models, uncertainty and complete query languages. *J. Comput. Syst. Sci.*, 76, 538–560.
- Langran, G., 1992. *Time in geographic information systems*. Technical issues in geographic information systems Taylor & Francis.

- Laube, P., *et al.*, 2007. Movement beyond the snapshot - Dynamic analysis of geospatial lifelines. *Computers, Environment and Urban Systems*, 31 (5), 481 – 501 Geospatial Analysis and Modeling.
- Laube, P., Imfeld, S., und Weibel, R., 2005. Discovering relative motion patterns in groups of moving point objects. *International Journal of Geographical Information Science*, 19, 639–668.
- Mark, D., 1998. Geospatial Lifelines. *Integrating Spatial and Temporal Databases*, Seminar 98471.
- Merrill, E., *et al.*, 2010. Building a mechanistic understanding of predation with GPS-based movement data. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365 (1550), 2279–2288.
- Morales, J.M., *et al.*, 2010. Building the bridge between animal movement and population dynamics. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365 (1550), 2289–2301.
- Obe, R. und Hsu, L., 2011. *PostGIS in Action*. Manning Pubs Co Series Manning Publications.
- Okubo, A., *et al.*, 2001. The dynamics of animal grouping. *Journal of Theoretical Biology*, 14, 197–237.
- Owen-Smith, N., Fryxell, J.M., und Merrill, E.H., 2010. Foraging theory upscaled: the behavioural ecology of herbivore movement. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365 (1550), 2267–2278.
- Parent, C., Spaccapietra, S., und Zimányi, E., 2006. *Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.
- Pelekis, N., Theodoulidis, B., und of Computation, U.D., 2002. *STAU: a Spatio-Temporal extension for the Oracle DBMS*. UMIST.
- Pelekis, N., *et al.*, 2006. Hermes - A Framework for Location-Based Data Management. In: *In Proceedings of EDBT 2006* Elsevier.
- Pelekis, N., *et al.*, 2004. Literature review of spatio-temporal database models. *Knowl. Eng. Rev.*, 19, 235–274.
- Peuquet, D.J., 1994. It's About Time: A Conceptual Framework for the Representation of Temporal Dynamics in Geographic Information Systems. *Annals of the Association of American Geographers*, 84 (3), 441–461.
- Peuquet, D.J. und Duan, N., 1995. An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *International journal of geographical information systems*, 9 (1), 7–24.
- Praing, R. und Schneider, M., 2007. Modeling historical and future movements of spatio-temporal objects in moving objects databases. In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, CIKM '07*, Lisbon, Portugal New York, NY, USA: ACM, 183–192.

- Shekhar, S., *et al.*, 1999. Spatial Databases-Accomplishments and Research Needs. *IEEE Trans. on Knowl. and Data Eng.*, 11, 45–55.
- Silva, J. und Santos, M., 2010. Spatiotemporal database models and languages for moving objects: A review. In: *Information Systems and Technologies (CISTI), 2010 5th Iberian Conference on*, Juni., 1 –4.
- Sistla, A.P., *et al.*, 1997. Modeling and Querying Moving Objects. In: *Proceedings of the Thirteenth International Conference on Data Engineering, ICDE '97* Washington, DC, USA: IEEE Computer Society, 422–432.
- Smouse, P.E., *et al.*, 2010. Stochastic modelling of animal movement. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365 (1550), 2201–2211.
- Spaccapietra, S., *et al.*, 2008. A conceptual view on trajectories. *Data and Knowledge Engineering*, 65 (1), 126 – 146 Including Special Section: Privacy Aspects of Data Mining Workshop (2006) - Five invited and extended papers.
- Teorey, T., Lightstone, S., und Nadeau, T., 2011. *Database Modeling and Design: Logical Design*. 5. Auflage The Morgan Kaufmann series in data management systems Elsevier.
- Tomkiewicz, S.M., *et al.*, 2010. Global positioning system and associated technologies in animal behaviour and ecological research. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365 (1550), 2163–2176.
- Tryfona, N. und Jensen, C.S., 2000. Using abstractions for spatio-temporal conceptual modeling. In: *Proceedings of the 2000 ACM symposium on Applied computing - Volume 1*, SAC '00, Como, Italy New York, NY, USA: ACM, 313–322.
- Urbano, F., *et al.*, 2010. Wildlife tracking data management: a new vision. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365 (1550), 2177–2185.
- Vesco, U., *et al.*, 2011. An integrated database on ticks and tick-borne zoonoses in the tropics and subtropics with special reference to developing and emerging countries. *Experimental and Applied Acarology*, 54, 65–83.
- Wolfson, O., *et al.*, 1999. DOMINO: databases fOr MovINg Objects tracking. *SIGMOD Rec.*, 28, 547–549.
- Worboys, M., 2005. Event-oriented approaches to geographic phenomena. *International Journal of Geographical Information Science*, 19, 1–28.
- Yan, Z., 2011. Semantic Trajectories. Thesis (PhD). EPFL, Lausanne.
- Yuan, M., 1997. Use of a Three-Domain Representation to Enhance GIS Support for Complex SpatioTemporal Queries. *Transaction in GIS*, 137–159.
- Yuan, M. und McIntosh, J., 2002. A Typology of Spatiotemporal Information Queries. In: R. Ladner, K. Shaw und M. Abdelguerfi, Hrsg. *Mining Spatio-Temporal Information Systems.*, Vol. 699 of *The Kluwer International Series in Engineering and Computer Science* Springer, 63–81.

# Anhang A

## Anforderungstext

Der in der Anforderungsanalyse erstellte und im Laufe des konzeptuellen Entwurfs verfeinerte Anforderungstext wird mit einem Farbcode versehen präsentiert:

**spatial** or **spatially related** and **temporal** information

**entities**, *relations*, attributes

---

### requirements text

The Inkavu Vervet Project consists of several **persons** which are part of different **universities**. Pairs of observers go after one assigned group of monkeys where they complete various forms on a PDA depending on the different occurrences.

---

Data on location is obtained through an **automatic GPS data acquisition** of an individual per group representing the location of the whole group. The temporal granularity is 30mins and besides the coordinates, elevation and the date & time of acquisition, the data contains further some information about some accuracies (GPS-noise, TTFF, horizontal and speed), temperature, speed and direction of the monkey (through an accelerometer) and the remaining battery.

Additionally, the observers in the field have handheld GPS which they use when special occurrences (like Alarm Calls, Between Group Encounter, ...) happen.

A monthly **satellite image** with the vegetation's **productivity information** and **visibility data** as well as a vegetation map with different seasons will work as a context information.

---

The **Log** is filled once a day, it includes the *two observers*, which **car** they took, the **group** which they are following, **time of sunrise and sunset** (according to their handheld GPS), **time of first and last visual contact** with group and the collected **type of data** (lookup list: Observational, Habituation, Novel object experiment, Social transmission experiment, BGE experiment, more might be added later) and remarks. If an experiment is executed, the location is described. This data is mostly to check when experiments are done.

Monkeys travel around the field site within groups. Even if they mostly stick to one group, switching between them occurs. Different **kinds of changes in the group composition** are noted with a timestamp: monkeys can be spotted when just being born or shortly after it (birth), so a tentative name, the mother and sex are provided. A new independent (not baby) monkey in the group (immigration) is given a name, if a known individual is seen leaving the group (emigration) the name is noted (same if not seen during the whole day - > disappearance). Finally monkeys can die (death). It has to be stated, that monkeys can be spotted within multiple groups more or less at the same time, sometimes they're trying to get into another group and if not successful just get back to the old one.

---

Individuals are observed (**Focal**) 10 times within 10 minutes on a regular basis distributed over the day to get informations about the **distance to mother** (if applies, m), if there's an **infant nearby** (does not apply, physical contact, within 5m, not within 5m), if it is **in the sun** (yes/no), **position in group** (center/edge), lingering **height of individual** (2 m, 5, 10, 10+), **distance to refuge** (m), *nearest neighbour* and the **distance to nearest neighbour** (does not apply, physical contact, within 5m, not within 5m). The **time of observation** (minute accuracy), the **time zone** (4 equal time zones a day for equally distributed sampling), the **weather** situation (lookup list, multiple selection: cloudy, windy, ...), their *activity* (listed below), if chosen activity was **interrupted** within 1min (yes/no) and **remarks** are saved.

---

Within **moving**, there's information about the **gait** (walk, trot, gallop, leap, climb, descend) and the **speed** (slow, normal, fast).

Within **resting**, there's additional info about the detailed **action** (autogroom, sunbathing, generally inactive, vigilant), the **eyes** (open/closed) and the **stance** (sitting, lying, quadrupedal, bipedal).

**Feeding** includes the detailed **action** (forage, bite/chew, lick/drink, retch/regurgitate), the specific **item** (unknown, cheek pouch, soil, water, fungi, plant, invertebrate, amphibian/reptile, bird, mammal), which **part** (unknown, seed, fruit, ...), **state of the item** (unknown, ripe, unripe, young, larvae, ...), **species** (lookup list: plant ..., animal ...).

The details within **social** are about the behaviour, if it's **affiliative** (does not apply meaning it's aggressive, groom or being groomed, several forms of play, several forms of sexual behaviour) or **aggressive** (dominant stare, chase, hit or being hit, submissive behaviour, ...) and who the *partner* is.

---

The observed group is scanned every half hour for 10mins (**Scan**) and as many individuals as possible are observed for a short time. It consists of the *user* collecting the data, the **time of observation** + **exact time** (automatic counter +1min), estimated **group spread** (10m, 20, 50, 100, 100+), **weather** situation (lookup list, multiple selection: cloudy, windy, ...), the chosen *individual* and his *activity* (detailed information above), if there's an **infant nearby** (does not apply, physical contact, within 5m, not within 5m), **in the sun** (yes/no), **position in group** (center/edge), the animal's **height** (2m, 5, 10, 15, 20, 20+), **distance to refuge** (NA, 2m, 5, 5+), *nearest neighbour* and **distance to nearest neighbour** (does not apply, physical contact, within 5m, not within 5m), number of **monkeys within 10m** (N) and **remarks**. The **location** of the scan is provided through the GPS fix every half hour.



When there's an **AlarmCall** occurring, the particular **alarm vocalisation** (several to choose) and if they're **loud calls**? (yes/no), the *group*, the *first caller* and *additional callers* are noted. The **group response** (lookup list, multiple selections possible: ignore, scan air, scan tree, scan ground, approach caller, contract, disperse, go up tree, go down (in) tree) must be chosen, if there is **another group present/nearby** (yes/no) or if a **predator is sighted** (yes/no) and **mobbed** by multiple individuals (yes/no?). Additional information on the context of the alarm call should be provided in the description using ethogram stenography. Time and location are provided through a handheld GPS by the *person* who notes the alarm call.

If a Between Group Encounter (**BGE**) occurs, the *observed* as well as the *encountered group* is saved, start/end time, predominant *activity* at start (either moving, resting, feeding or social; no details) and nature of BGE (affiliative, aggressive, neutral, ignore), estimated group cohesion (10m, 20, 50, 100, 100+) of both groups and cooperation formed within both groups (yes/no). The **general behaviour of individuals from different groups** (multi-selection: feed in same patch, allogroom, play, engage in sexual behaviour, scream, displace, chase, hit, bite) is chosen. The number of actively involved individuals (N), their AgeSexClass (adult, sub-adult, juvenile males/females) and if possible the *5 most active monkeys* of both groups are noted. Was there a clear winner group (yes/no), which one? A detailed description is noted using ethogram stenography. Start/end location is provided through a handheld GPS by the *person* who notes the BGE.

Due to the hectic and complex nature of a BGE, the occurrence is mostly filmed and reconstructed afterwards.

When there's **Sexual Behaviour**, the *group*, the *actor* and *reactor* as well as their behaviour is described using ethogram steno. If there was a successful **copulation** (yes/no) and remarks. Time and location are provided through a handheld GPS by the *observer*.

If Within Group Competition (**WGC**) is happening, the researchers include the *group*, *actor* and *reactor* and both their behaviour (descriptive) in their notes. The context (descriptive) and a detailed description are noted. Time and location are provided through a handheld GPS by the *person* who notes the WGC.

**PhysicalMeasurement** is filled with the *person* measuring, the *individual* and the current *group*, **sex** (m/f/?), their **ageClass** (infant, juvenile, subadult, adult, old adult). Scars, wounds and the number of ticks on the animal are noted. Diverse measurements are taken (weight, tail length, feet length, upper arm circumference, ... [mm]). An ear mark pattern is made for easier recognition of the individuals (unique pattern -> number saved) and tissue sample is taken for genetic analysis and stored. The label of the tissue sample (group code + number of individual), general remarks and notes are saved.



## Anhang B

# Entity Relation Model

Auf den nächsten Seiten wird das vollständige *entity relationship model* vorgelegt, zuerst die Basis-Version und nachfolgend noch eine Variante mit den 'räumlich angereicherten' Strukturen aus Kapitel 8.

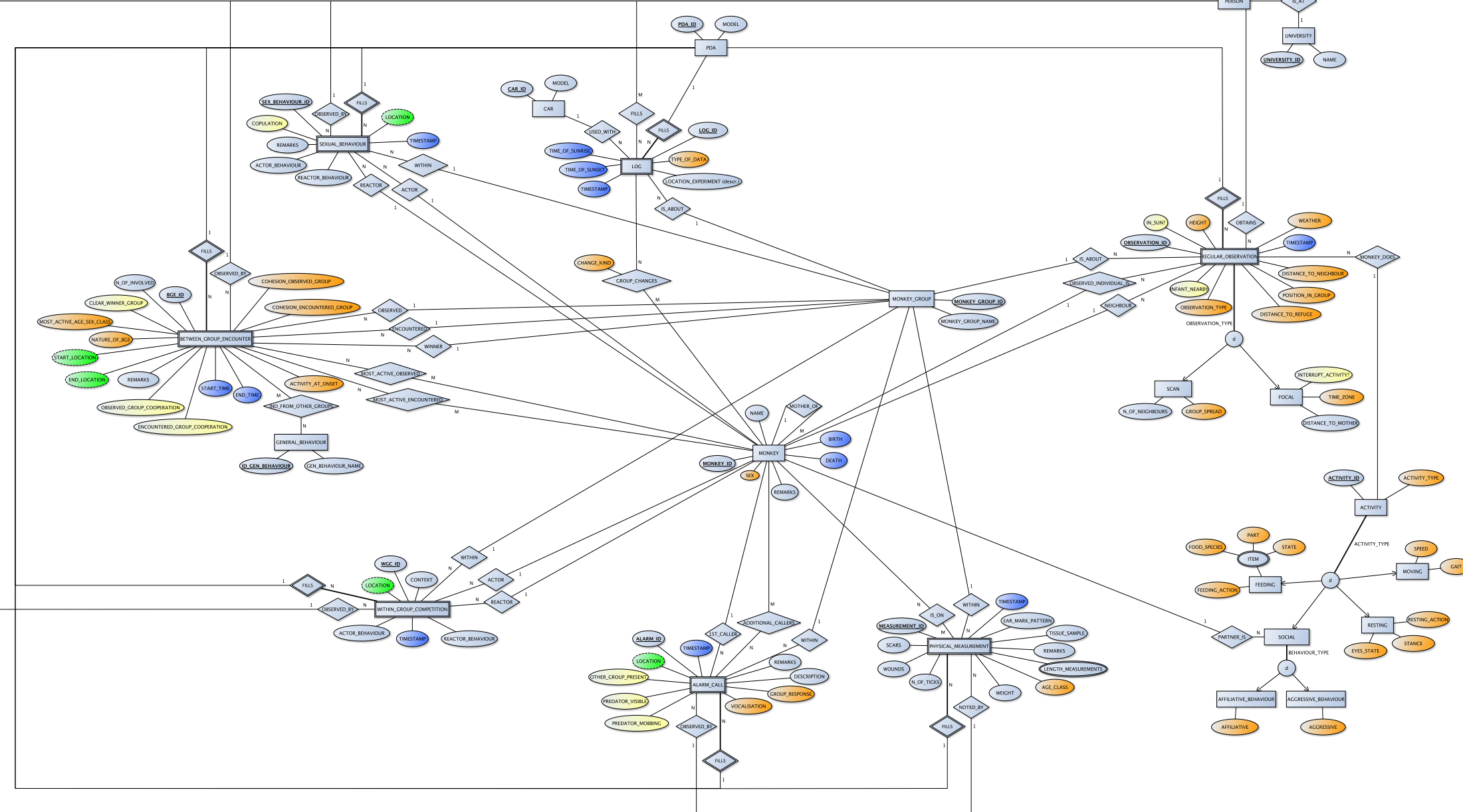
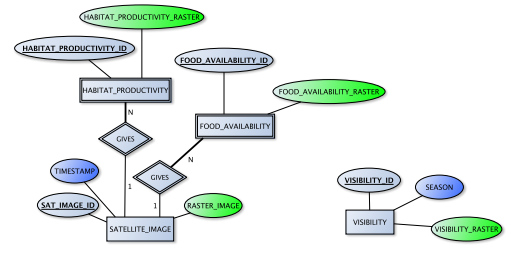
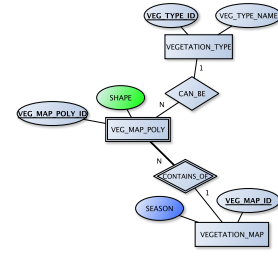
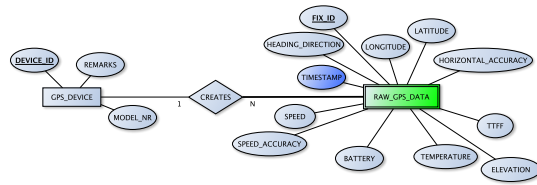
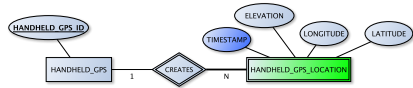


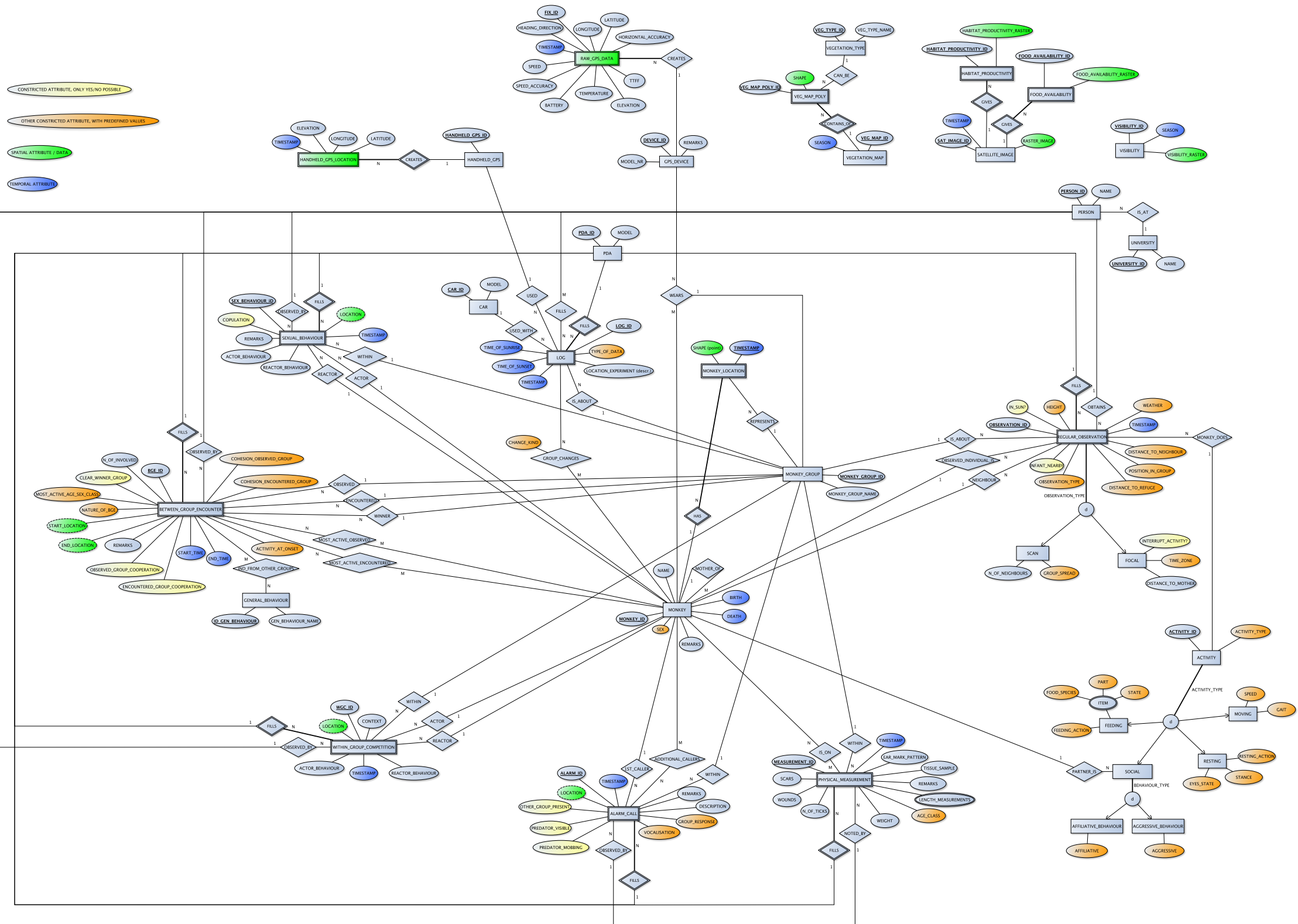
CONSTRICTED ATTRIBUTE, ONLY YES/NO POSSIBLE

OTHER CONSTRICTED ATTRIBUTE, WITH PREDEFINED VALUES

SPATIAL ATTRIBUTE / DATA

TEMPORAL ATTRIBUTE





# Persönliche Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und die den verwendeten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Näfels, 30. Januar 2012

Sebastian Schmidig