

Molekularbiologische Datenbanken

Datenmodelle

Ulf Leser

Wissensmanagement in der
Bioinformatik

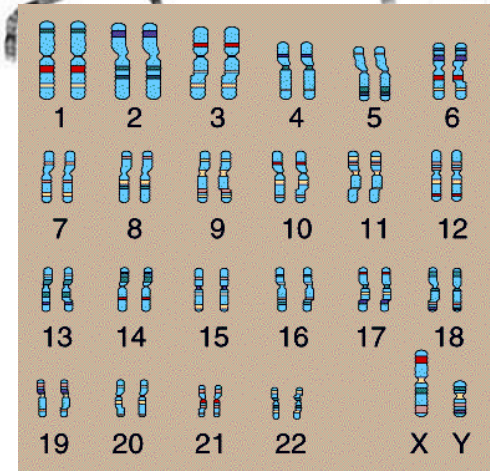
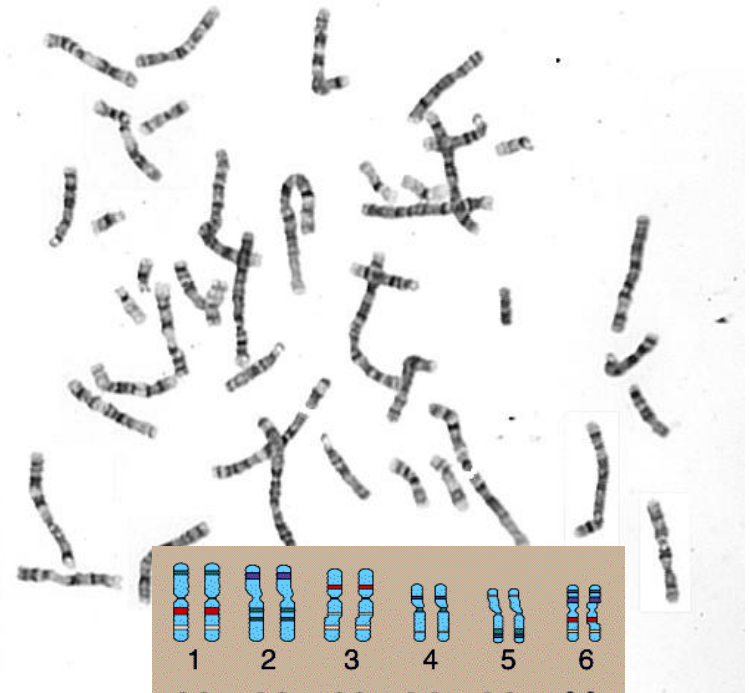


Motivation

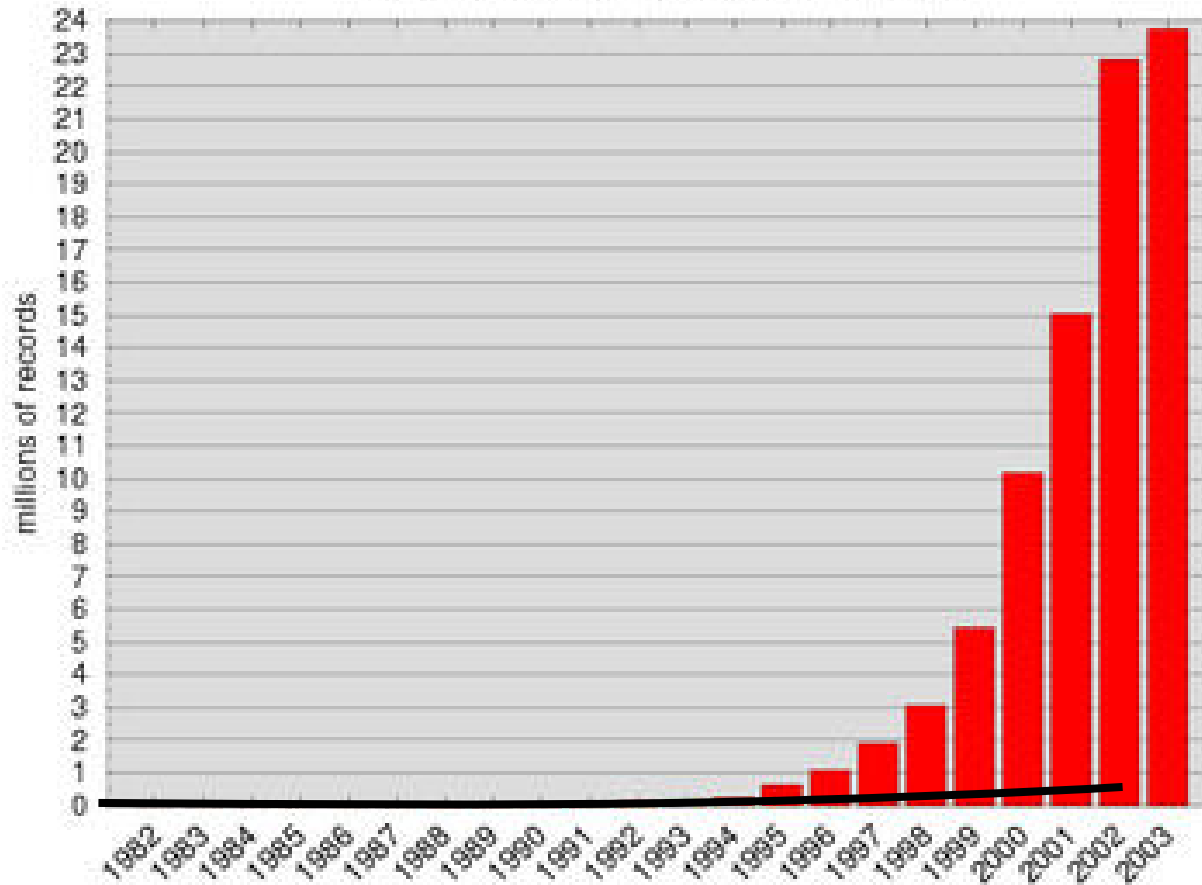
- Warum gibt es öffentlich zugängliche molekularbiologische Datenbanken (MDB) ?
- Data Submission = **Veröffentlichung**
- Veröffentlichung
 - Austausch von Informationen und Daten
 - Erlangung von Reputation
 - Öffentliche Gelder – öffentliche Daten
- **MDB: Veröffentlichungen, nicht Tupel**

Human Genome Projekt

- Start: ca. 1990
- Ziel
 - Sequenzierung des humanen Genoms
 - Identifikation aller Gene
 - ~3.000.000.000 Basenpaare
 - 25.000 – 33.000 Gene
- Ursprünglicher Plan: bis 2005
- Effekt wie Mondlandung:
Vision erzeugt Beschleunigung



Vergleich EMBL / PDB



Quelle: EMBL, Genome Monitoring Tables, Stand 12.2.2003

Text -> Daten

- Alle (großen) öffentlichen Datenbanken entstanden aus Büchern
 - Sammlungen bekannter Daten einer Art: DNA, Proteinsequenz, Proteinstruktur
 - Jährliche / Quartsweises Erscheinen
 - Buch – Band – CD – FTP – WWW
- Flaches, textorientiertes Datenmodell
 - Aufbau nach „Entries“
 - Viele Beschreibungen in freier Textform
 - Für Menschen, nicht Computer

Eigenschaften

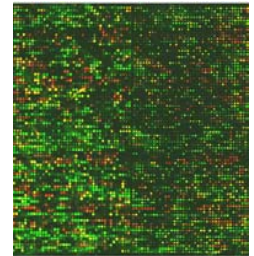
- Aktiv - passiv
- Curated- Archiv
- Redundanzgrad
- Integrationsgrad – Links, physische Integration
- Fokus – Organismus, Objektklassen, Krankheit, ...
- Primär – Sekundär – Tertiär
- Zweck – Projekt, weltweites Archiv, kommerzielles Produkt

Inhalt dieser Vorlesung

- Einige Anforderungen an MDB
 - Datenmengen, Flexibilität, ...
- Datenmodelle in Life Science Datenbanken
 - Entry-Based, ER/relational, ...

Anforderung: Datenmenge

- GenBank: 110 GB
 - Uncompressed Flatfiles, Rel. 134, 2/2003
- SWISS-PROT + TrEMBL: 1 GB
 - Oracle Export, Compressed, Stand 2/2003
- Rohdaten: **Deutlich mehr !**
 - TIFF eines Genexpressionschips: ca. 50 MB
 - Bilder von 2D Gelen
 - Tracefiles von Sequenziermaschinen
 - Massenspektrographie
 - ...



Anforderung: Flexibilität

- Entwicklung von MDB folgt nur selten
 - Anforderungen – Spezifikation – Implementierung – Test
- Forschungsfragen ändern sich
 - Andere Fragen
 - Andere Queries
 - Andere Daten
- Design muss **Wartbarkeit und Flexibilität** in Vordergrund stellen
 - Schemaänderungen
 - Einbringung neuer Datentypen
 - Optimierung auf neue Anforderungen
 - Integration externer Quellen
 - ...

Anforderung: Offenheit

- MDB meist Teil eines Forschungsprojekts
 - Datenbeschaffung (LIMS)
 - Datenarchivierung
 - Datenanalyse (MatLab, Data Mining, ...)
- Zugriff von verschiedensten Clients (Java, CGI, Perl, PHP, ...)
- Aufruf spezieller Algorithmen notwendig
 - Sequenzvergleich: Blast/Fasta
 - Sequenzanalyse: Motivesearch, MSA, Genidentifikation
 - Strukturvorhersage,
- Blast in Oracle / DB2 ?

Anforderung: Beachtung von Datenqualität

- Experimente erzeugen **unscharfe Daten**
 - Arbeit mit lebenden Organismen
 - Ablaufende Mechanismen oft inhärent statistisch
 - Ablaufende Mechanismen teilweise nur schlecht verstanden
 - Fehleranfällige, mit Rauschen behaftete Techniken: Bilderkennung, Signalverarbeitung, heuristische Algorithmen, ...
- Eine junge Wissenschaft
 - Falsche Daten und Veröffentlichungen nicht selten
- Zwei Experimente – zwei Ergebnisse
 - Widersprüchliche Daten
 - Fehlerberechnung

Beispiel: Funktionale Charakterisierung

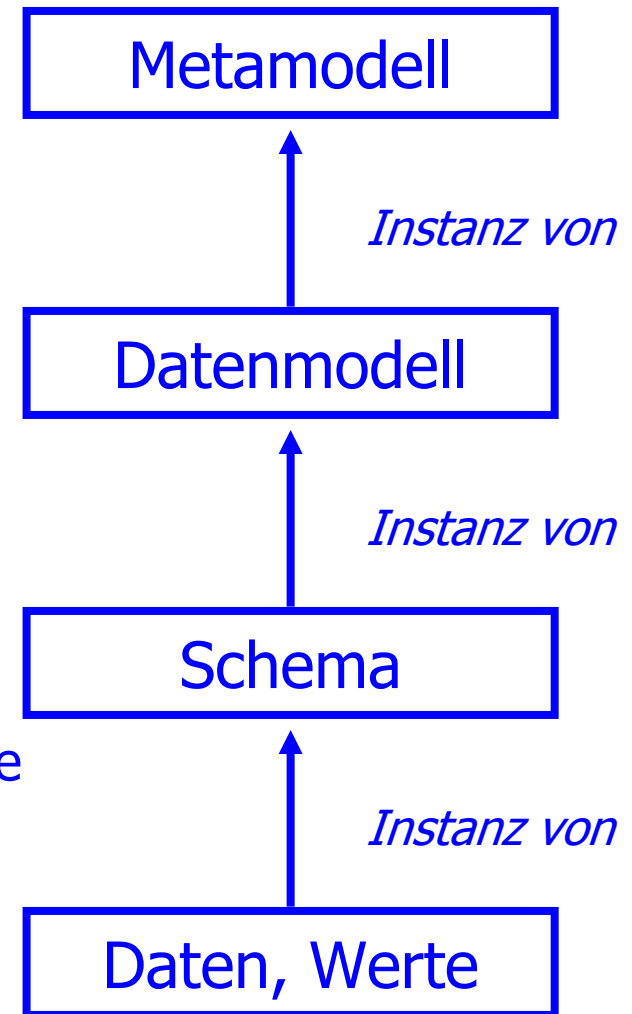
- Für über 50% aller bekannten Proteine in Yeast (Modellorganismus No. 1!) ist keinerlei Funktion bekannt
- Methoden
 - Ähnliche Sequenz – ähnliche Funktion
 - Ähnliche Struktur – ähnliche Funktion
- Verfahren ist ungenau
 - Was ist ähnlich ? Wie ähnlich ?
 - Biologie berücksichtigt ? (Gen ohne Promoter, Exon ohne Startcodon, ...)
- Problem
 - Annotation basierend auf Homologie ohne Kennzeichnung
 - Dient als Grundlage für weitere Annotationen
 - Ca. 30% aller Annotationen sind vermutlich falsch

Inhalt dieser Vorlesung

- Datenmodelle in Life Science Datenbanken

Daten, Schema, Modell, Metamodell

- Metamodell
 - Definition von Struktur
 - Meist auf Graphen abgebildet
 - Selten benutzt oder definiert
- Datenmodell
 - Definiert Strukturen und ihre Semantik
 - Bestimmt die Elemente der DDL
 - I.d.R. fest implementiert
- Schema
 - Container und Struktur der Werte
 - Definiert durch **Data Definition Language**
 - Anfragen adressieren Schemaelemente
- Daten
 - Tatsächliche Werte, uninterpretiert
 - Anfrageergebnisse



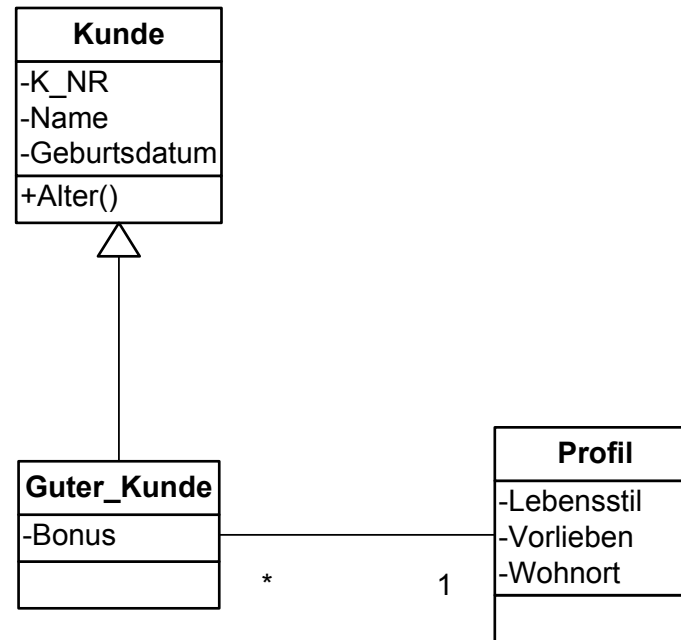
Beispiel: RDBMS

Tabellen, Attribute, Schlüssel

Kunde(K_nr, Name, Umsatz, Geb_datum, Beruf, ...)

1,	,Meier`,	130.44,	12/12/1945,	NULL, ...
2,	,Müller`,	null,	01/01/1962,	,Maurer`, ...
3,	,Brie`,	123.40,	01/10/1958,	NULL, ...

Graphische Notation



Kunde(K_nr, Name, Geb_datum, Alter())
Guter_Kunde ISA Kunde (bonus, ->Profil)
Profil(Lebensstil, Vorlieben, Wohnort)

Umgangssprache

- Formal
 - Datenmodell bestimmt mögliche Modellelemente
 - Konkretes System hat Schema
 - Bestehend aus Modellelementen
- Umgangssprache
 - Modell universeller Begriff
 - Modell bezeichnet Metamodell, Datenmodell, Schema
 - „Modell der Datenbank“
 - „UML-Modell des Programms“

Teil I: Datenmodelle und Systeme

- Entry-basierte Formate
- Entity-Relationship Modell
- Relationales Modell
- UML, objektorientierte Modelle
- XML
- ACeDB

1. „Entry-basiertes“ Datenmodell

- Kein Datenmodell im eigentlichen Sinne
- Flat-file
- Weite Verbreitung in Life Sciences
 - EMBL, SWISS-PROT, Interpro, Omim, Genbank ,...
 - Nahezu alle Datenbanken haben ein Entry-Based Austauschformat
- Beispiel SWISS-PROT
 - Menge von Proteinsequenzen
 - Core-Elemente: Sequenz, Taxonomy, Citation
 - Annotationen: Domänen, Sequenzvarianten, assoziierte Krankheiten, Sekundärstruktur, ...

ID GRAA HUMAN STANDARD; PRT; 262 AA.

AC D12574

DT 01-OCT-1989 (Rel. 12, Created)

DT 01-OCT-1989 (Rel. 12, Last sequence update)

DT 16-OCT-2001 (Rel. 40, Last annotation update)

DE Granzyme A precursor (EC 3.4.21.78) (Cytotoxic T-lymphocyte p. 1) (Hanukkah factor) (H factor) (HF) (Granzyme 1) (CTL tryptase) (Fragmentin 1).

GN GZMA OR CTLA3 OR HFSP.

OS Homo sapiens (Human).

OC Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;

OC Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.

OX NCBI_T

RN [1]

RP SEQUENCE

RC TISSUE

RX MEDLIN

RA Gershe

Werte mit verschiedenen, semantisch bedeutsamen Bestandteilen

komplexe, mengenwertige Attribute

Feldabhängige Formate (Microsyntax)

Eingebettete Objekte (keine Verweise)

RX MEDLINE=88330824; PubMed=3047119; RA Poe M., Bennett C.D., Biddison W.E., RA Rodkey J.A., Sigal N.H., Turner R.V., RT "Human cytotoxic lymphocyte tryptase and the characterization of inhibitor RL J. Biol. Chem. 263:13215-13222(1988)

RN [3] RP SEQUENCE OF 29-40, AND CHARACTERIZATION. RX MEDLINE=89009866; PubMed=3262682; RA Hameed A., Lowrey D.M., Lichtenheld M., Podack E.R.; RT "Characterization of three serine esterases isolated from human IL-2 RT activated killer cells."; RL J. Immunol. 141:3142-3147(1988).

RN [4]

...

Elemente

- Datenbank = Menge ähnlich strukturierter **Entries**
- Entry: Menge von **Feldern** (Attribute, Lines),
 - Identifikation durch Line Code
 - Können 0-n mal vorkommen (semistrukturiert)
 - Können **komplexe eigene Struktur** haben
 - Können eingebettete Objekte repräsentieren
 - Microsyntax in Werten (Sprechende Schlüssel)
- Keine deklarativen Konsistenzbedingungen
- Kein Klassen- oder Objektbegriff

Konsistenz versus Vereinbarung

Line code	Content	Occurrence in an entry
ID	Identification	Once; starts the entry
AC	Accession number(s)	Once or more
DT	Date	Three times
DE	Description	Once or more
GN	Gene name(s)	Optional
OS	Organism species	Once or more
OG	Organelle	Optional
OC	Organism classification	Once or more
OX	Taxonomy cross-reference(s)	Once or more
PN	Reference number	Once or more

The description always starts with the proposed official name of the protein. Synonyms are indicated between brackets. Example:

...

When a protein is known to be cleaved into multiple functional components, the description will start with the name of the precursor protein, followed by a section delimited by '[Contains: ...]'. All the individual components are listed in that section and are separated by semi-colons (;'). Synonyms are allowed at the level of the precursor and for each individual component.

Example:

Einordnung

- Datenmodell
 - Entries (1 Typ pro Datenbank)
 - Felder mit hierarchischer Schachtelung
 - Keine expliziten Assoziationen
- Schema
 - Keine explizite Repräsentation
 - Definition: Dokumentation / Parserimplementation
- Werte
 - Zusammengesetzte Werte möglich

Ist „entry-basiert“ ein Datenmodell?

- Gibt es ein Datenmodell „entry-basiert“?
 - Nein
 - Kein formaler Standard
 - Aber weit verbreitete Konvention
- Entry-basiert kann leicht in andere Datenmodelle übertragen werden
 - Relationales Modell (Später genauer)
 - Objektorientiertes Datenmodell: Eine Klasse mit komplexen und mengenwertigen Attributen
 - XML, semistrukturierte Datenmodelle (später mehr)

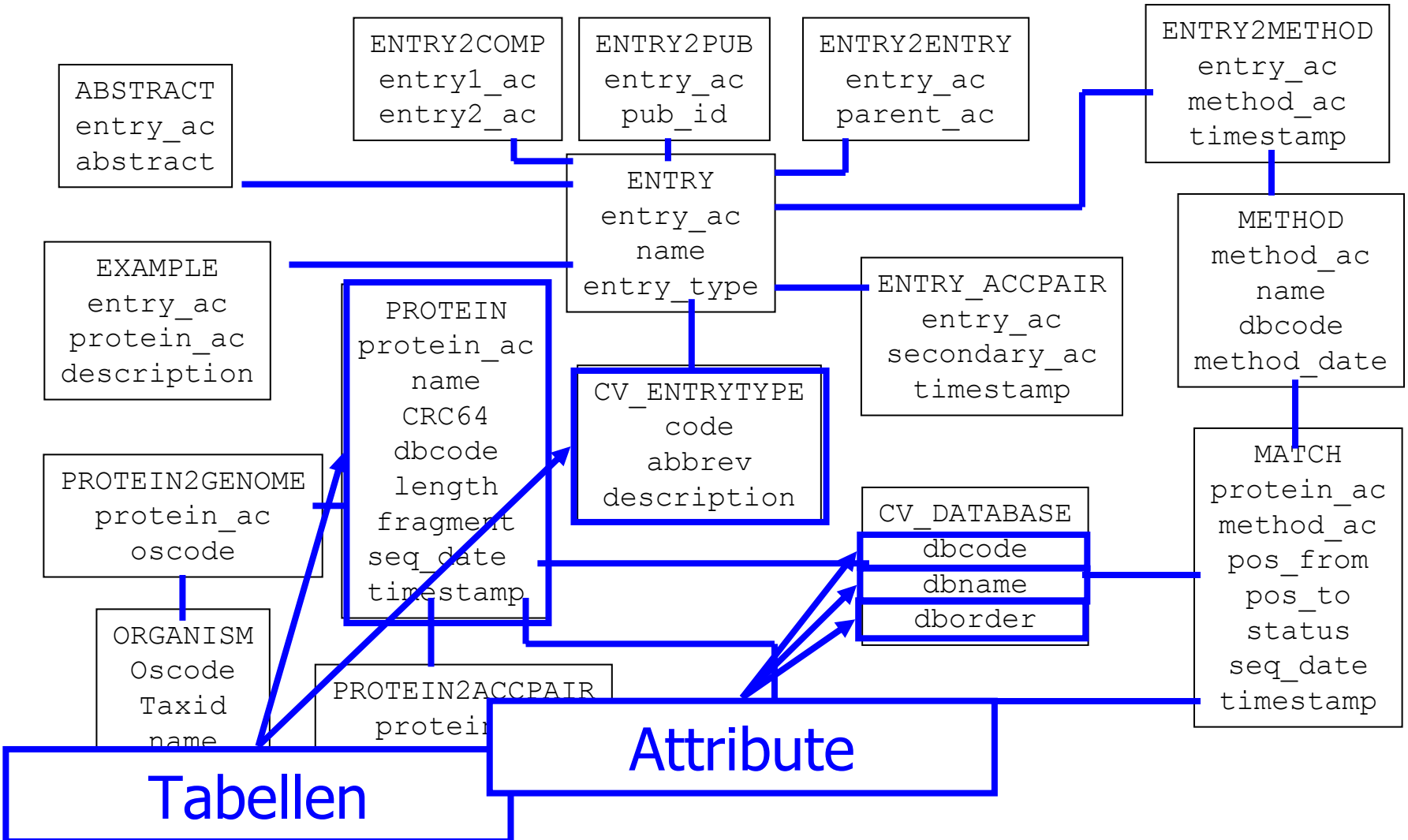
Bewertung

- Vorteile
 - Sofort lesbar für Menschen
 - Editierbar mit Emacs, Word, Wordpad, vi, ...
 - Plattformunabhängig (ASCII)
 - Hohe Flexibilität durch textorientiertes Editieren
 - Leicht zu durchsuchen (Grep, „search“ Button)
- Nachteile
 - Formatänderung bedingt Programmänderung
 - Keine Konsistenzbedingungen: FK, Wertebereiche, ...
 - Hohe Redundanz geschachtelter Objekte: Literatur, Tax., ...
 - Schwierig zu parsen (Microsyntax, keine Grammatik)
 - Keine strukturierten Anfragen möglich

2. Relationales Modell

- Industriestandard
- Konzentration auf Speicherung/Retrieval
 - Anwendungsneutrales Datenmodell
 - Semantisch arm, wenig Elemente
 - Nicht als Designmodell gedacht (ER)
 - Userinterface müssen programmiert werden
- Entwickelt für Transaction-Processing, Mehrbenutzerbetrieb, Client-Server
 - Overhead für typische „Read-Only“ MDB's
 - Komplizierte Installation, Administration, Backup, ...

Beispiel: InterPro Schema



Bewertung

- Datenspeicherung und -retrieval
- Vorteile
 - Strukturierte Anfragen
 - Sehr weit verbreitet, robust, Industriestandard
 - Skalierbarkeit und Optimierbarkeit
 - Viele Produkte verfügbar
 - Ständige Weiterentwicklung
- Nachteile
 - SQL schwierig zu lernen
 - Volltextsuche nicht direkt möglich
 - Erfordert speziell zu entwickelnde Editoren, Suchmasken, ...
 - Administration erforderlich
 - Datenaustausch / Backup schwieriger als mit Flatfiles
 - Teuer

Entity-Relationship Modell (ER)

- Modellierungsverfahren für RDBMS
- Automatische Übersetzung in relationale Schema einfach
- Elemente
 - Entity
 - Relationship
 - Attribute
 - Kardinalitäten
- Zunehmende Ablösung durch UML
 - UML sem. reichhaltiger -> eingeschränktes UML

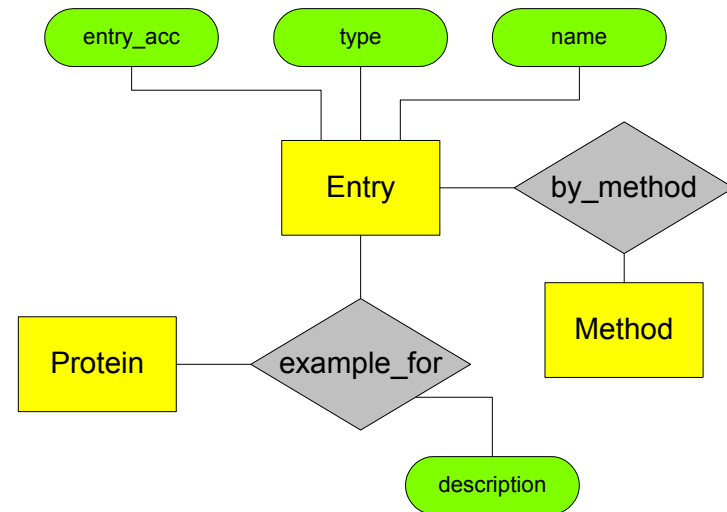
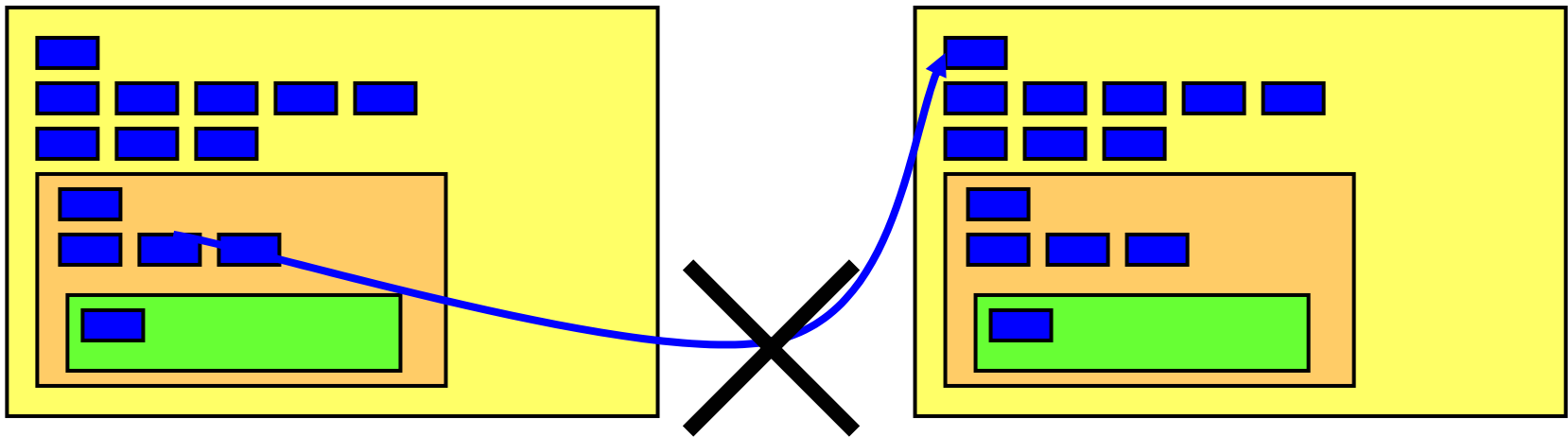


Abbildung Entry-based – Relational

- Flatfiles
 - Entries
 - Mengenwertige Attribute
 - Geschachtelte Subentries
 - Hierarchisch; keine Beziehungen



Natürliche Abbildung

```
FUNCTION entryToRelation( E)
  erzeuge relation R
  erzeuge Surrogateschlüssel K zu R
  FORALL A aus E           // Attribute
    CASE
      A einwertig -> A wird neues Attribut von R
      A mehrwertig -> erzeuge Relation A' mit FK K
      A komplex -> erzeuge Relation A' mit FK K
                    A' = entryToRelation( A)
    END CASE
  END FORALL
END
```

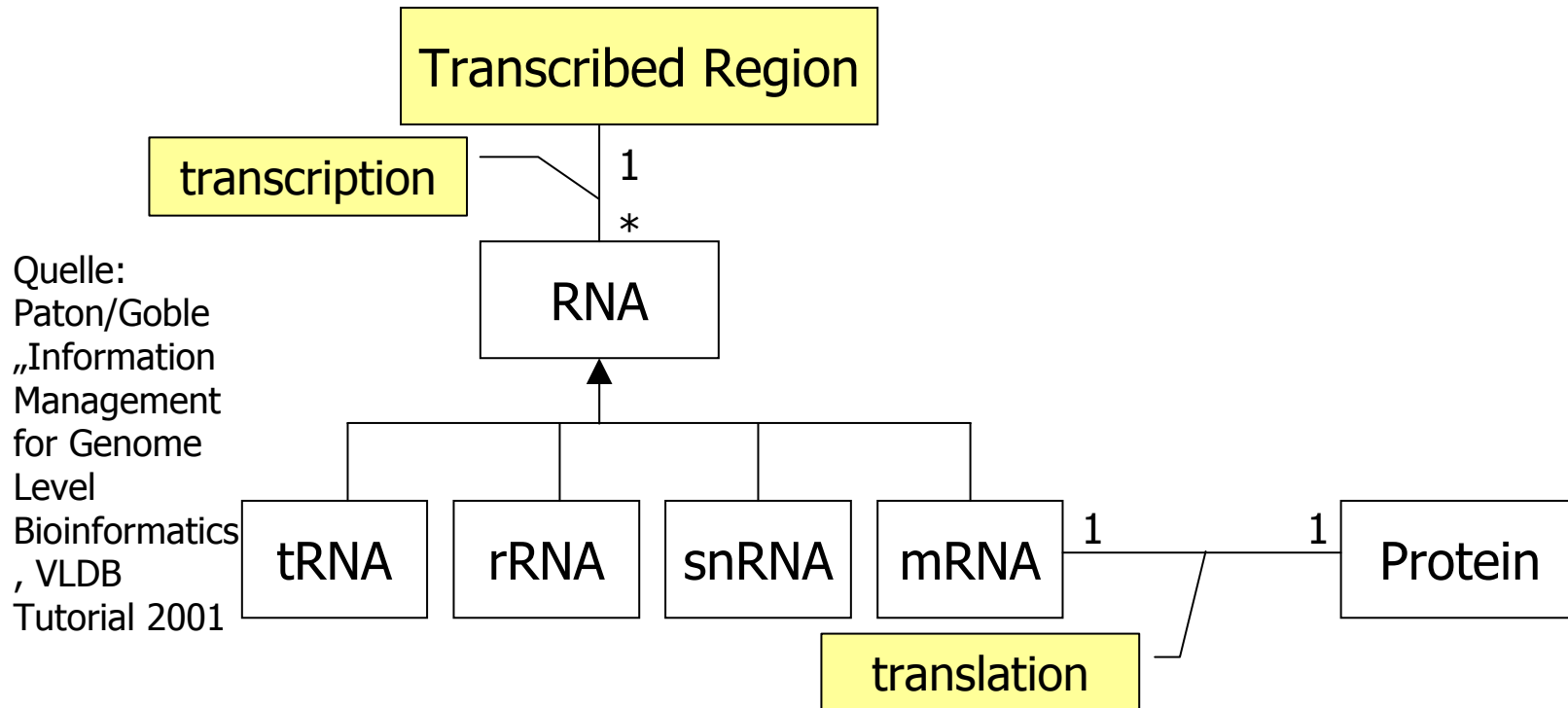
Entscheidungsspielräume

- Ist ein Attribut einwertig/ mehrwertig/ komplex?
 - Muss festgelegt werden
 - Attributparser i.A. nicht ableitbar
- Microsyntax: Auflösen oder belassen?
 - Erste Normalform
- Ordnung kann, muss aber nicht wichtig sein
 - Beispiel: Taxonomy
 - Gegenbeispiel: Publikationen
 - Abbildung in „Rank“ Attribute der Relation A'
- Redundanz: Mergen doppelter Subentries?
 - Beispiel: Publikationen
 - Erfordert (schwieriges) Testen auf Gleichheit

3. UML und objektorientierte Modelle

- Industriestandard zur Modellierung von
 - Software: Klassen-D., Sequenzcharts, Zustands-D.
 - Architekturen: Verteilungs-D., Komponenten-D.
 - Requirements: Use Cases
 - Prozesse: Aktivitäts-D., Collaboration-D.
- Viele Tools: Rational, TogetherJ, Argo-UML, ...
- **DB-Design: Klassendiagramme**
 - Modellierung in UML mit späterer Übersetzung in relationale Schema
 - Beispiele: Swiss-Prot, EMBL, GUS, GIMS, ArrayExpress, ...
 - Direkte Umsetzung in OODBMS möglich

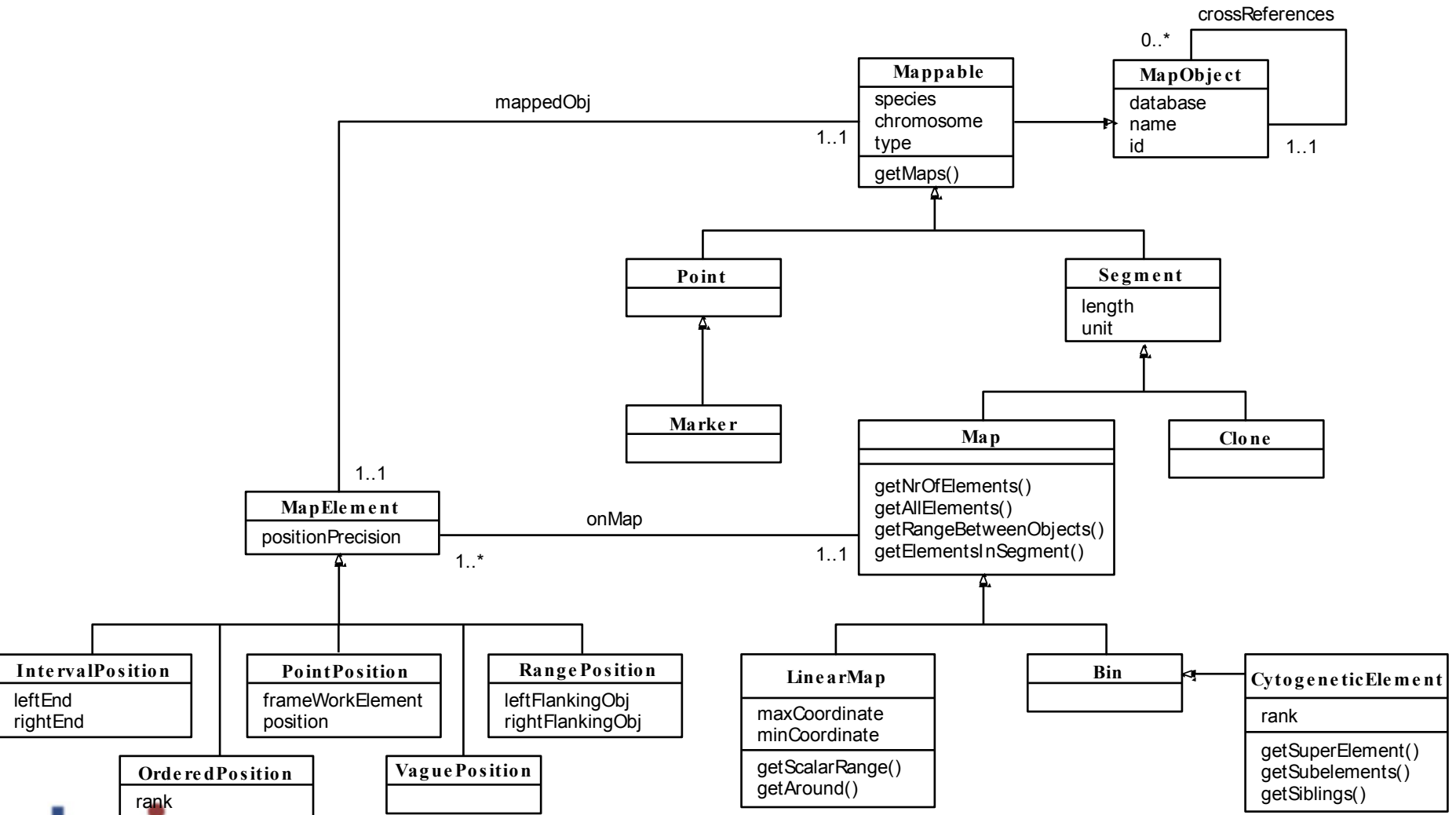
Beispiel : GIMS



Quelle:
Paton/Goble
„Information
Management
for Genome
Level
Bioinformatics
, VLDB
Tutorial 2001

- Oft nur informelle Verwendung, Illustration

Beispiel: LSR/CORBA



Elemente

- Datenmodell
 - Klassen
 - Attribute, Methoden
 - Assoziationen, Vererbung, (Aggregation?)
 - Kardinalitäten, Object Constraint Language (OCL)
- Schema: Klassendiagramm
 - Graphische Notation – aber klare Semantik
- Für Daten i.d.R. nicht verwendet
 - Methoden, Aggregation, OCL
 - Umsetzung unklar

Bewertung

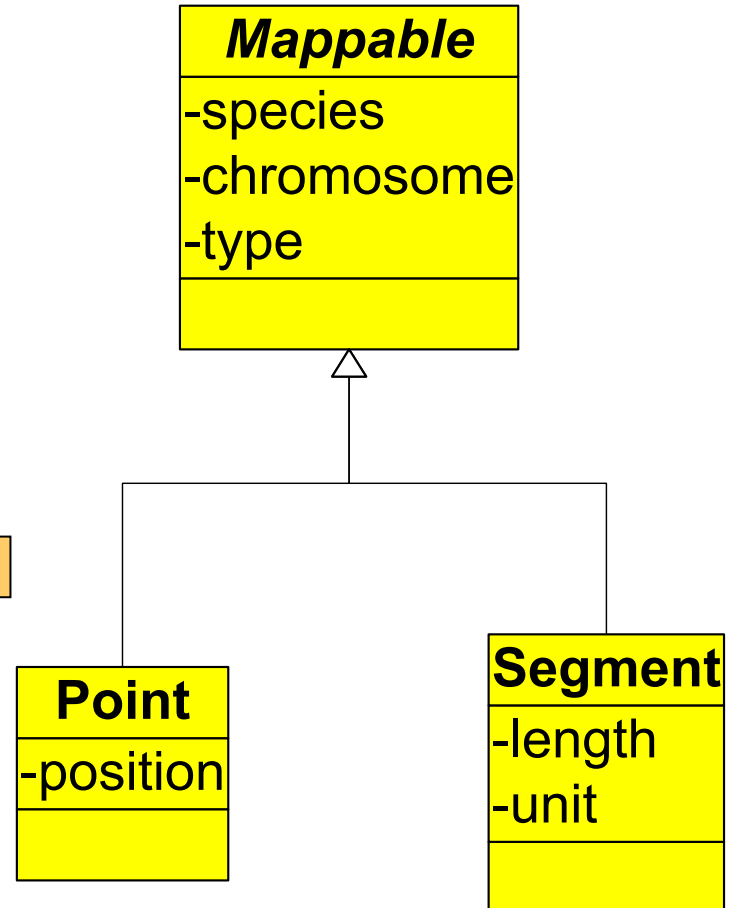
- Vorteile
 - Semantisch reichhaltiges Datenmodell (Vererbung)
 - Klar definierter graphischer Notation (durch Metamodell)
 - Industriestandard für Modellierung / Entwicklung
 - Enge Verkopplung mit Software möglich (Automatische Erzeugung von Persistenzschichten)
 - Nachteile
 - OO Datenmodelle werden in RDBMS gespeichert
 - Dualität OO – RDBMS nicht trivial (Impedance Mismatch)
 - OO-Übersetzung erzeugt wenig intuitive Schema
 - Schemaoptimierung nicht möglich (Roundtrip – Problem)
 - Queries adressieren rel. Schema, nicht UML Modell
- UML ist für Software, nicht für Datenbanken gemacht

Übersetzung UML - RDBMS

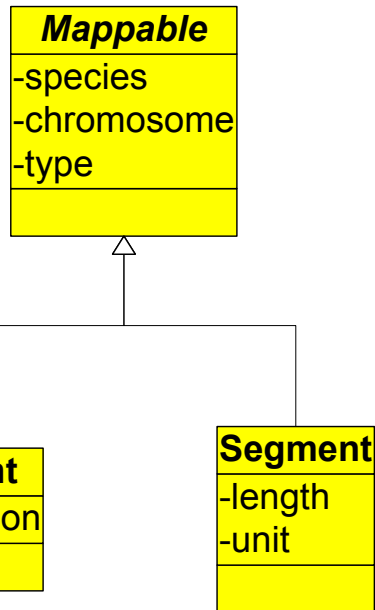
- Klasse – Tabelle
- Assoziation
 - 1:1 Einbeziehung oder Fremdschlüssel
 - 1:n Fremdschlüssel
 - m:n Brückentabelle
- Vererbung ?
 - Extensionale Inklusion
 - Intensional Erweiterung
 - **Verschiedene Möglichkeiten**

Objektrelationales Mapping

- Vier Varianten
 - **Union**: 1 Tabelle
 - **Vertikale Zerlegung**: 1 schmale Tabelle pro Klasse
 - **Horizontale Zerlegung**: 1 breite Tabelle pro Klasse
 - Volle Materialisierung
- Fragen
 - Speicherverbrauch: **Redundant**
 - **Extension**: Zugriff auf alle Objekte einer Klasse
 - **Intension**: Zugriff auf alle Attribute einer Klasse



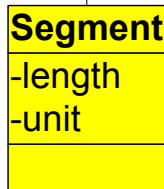
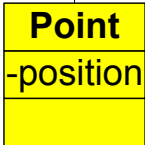
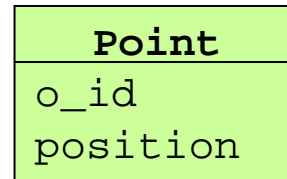
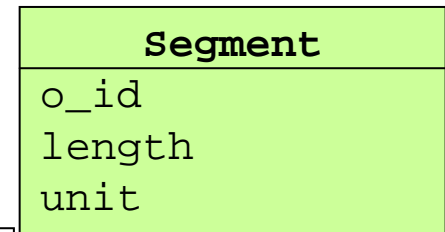
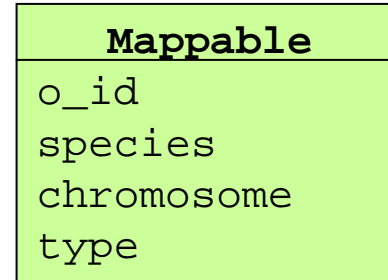
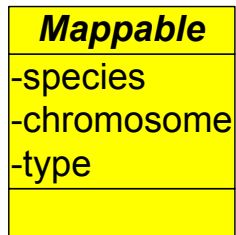
Union-Tabelle



MapHierarchy	
class_id	
species	
chromosome	
type	
length	
unit	
position	

MapHierarchy						
class_id	species	chromo	type	length	unit	pos
1	h.s.	X	STS			
1	m.m.	21	EST			
2	h.s.	12	STS			12q2
2	h.s.	13	STS			12q3
3	m.m.	14	STS	200	KB	
3	m.m.	X	STS	1.5	cM	

Vertikale Zerlegung (schmale Tabellen)



Point

o_id	pos
2	12q2
2	12q3

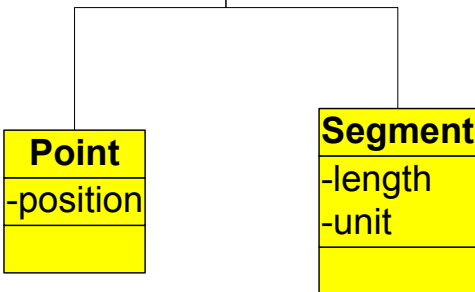
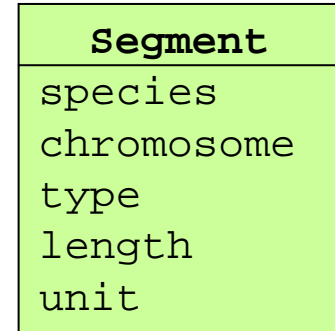
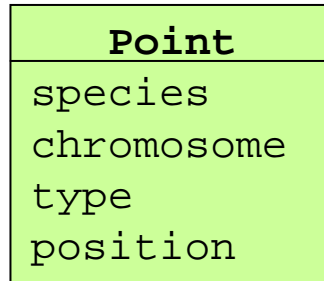
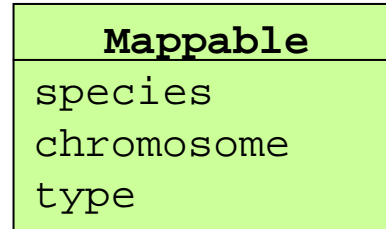
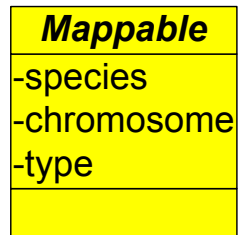
Mappable

o_id	species	chromo	type
1	h.s.	X	STS
1	m.m.	21	EST
2	h.s.	12	STS
2	h.s.	13	STS
3	m.m.	14	STS
3	m.m.	X	STS

Segment

o_id	length	unit
3	200	KB
3	1.5	CM

Horizontale Zerlegung (breite Tabellen)



Mappable

species	chromo	type
h.s.	X	STS
m.m.	21	EST

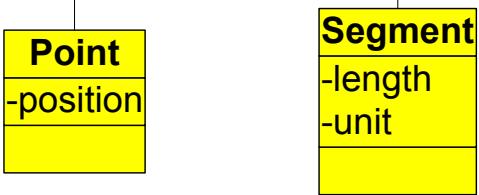
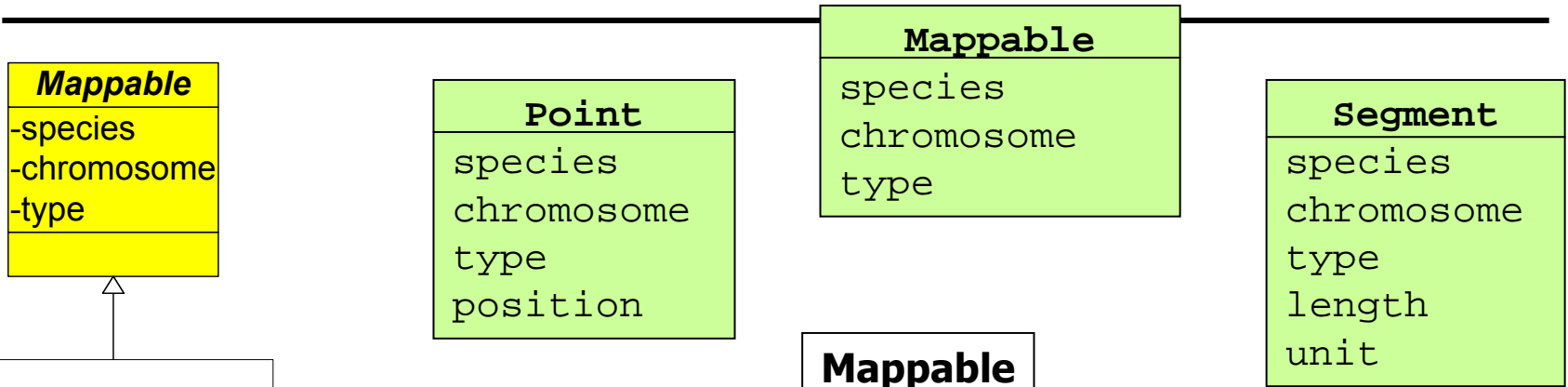
Point

species	chromo	type	pos
h.s.	12	STS	12q2
h.s.	13	STS	12q3

Segment

species	chromo	type	length	unit
m.m.	14	STS	200	KB
m.m.	X	STS	1.5	cM

OR- Volle Materialisierung



species	chromo	type
h.s.	X	STS
m.m.	21	EST
h.s.	12	STS
h.s.	13	STS
m.m.	14	STS
m.m.	X	STS

species	chromo	type	pos
h.s.	12	STS	12q2
h.s.	13	STS	12q3

species	chromo	type	length	unit
m.m.	14	STS	200	KB
m.m.	X	STS	1.5	cM

Vergleich

	Speicher- verbrauch	Zugriff Extension	Zugriff Intension	Konsistenz- sicherung	Inserts / Updates
Union	Hoch	Eine Query; Bedingung auf class_id	Eine Query; Bedingung auf class_id	NULL-Werte garantieren	1 Insert
Vertikal (Schmal)	Gering (OID doppelt)	Eine Query (nur OID + wenige Attribute)	N Joins (Alle Vorfahren)	FK von Kindertabs dürfen nicht überlappen	N Insert (Alle Vorfahren)
Horizont. (Breit)	Minimal	M Unions (Alle Nachfahren)	Eine Query		1 Insert
Voll	Hoch	Eine Query	Eine Query	Redundante Daten – Anomalien	N Insert (Alle Vorfahren)

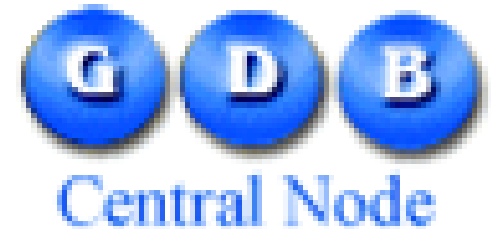
Bewertung

	Speicher- verbrauch	Zugriff Extension	Zugriff Intension	Konsistenz- sicherung	Inserts / Updates
Union	Hoch	Eine Query	Eine Query	NULL-Werte garantieren	1 Insert
Vertikal (Schmal)	Gering	Eine Query	N Joins	Überlappende FKs	N Insert
Horizont. (Breit)	Minimal	M Unions	Eine Query		1 Insert
Voll	Hoch	Eine Query	Eine Query	Anomalien	N Insert

➤ Optimale Methode ist anwendungsabhängig

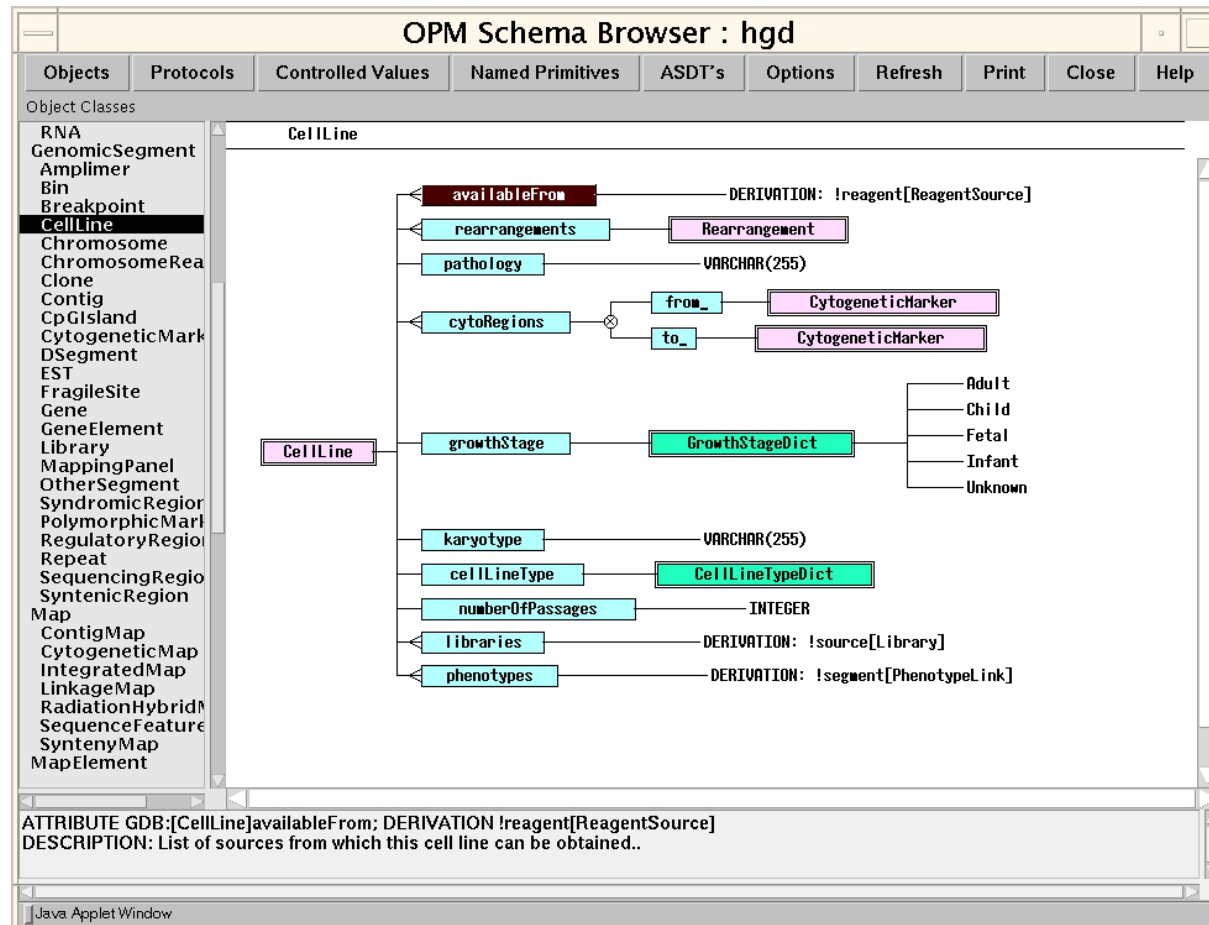
Beispiel: OPM

- Objektorientiertes Datenmodell
 - Vererbung
 - Mengenwertige und berechnete Attribute
- Einige Verbreitung in Life Science
 - GSDB, GDB, MGD, ...
- OPM-QL und OPM*QL
- Übersetzung in rel. Schema / Anfragen
 - Feste, vertikale Zerlegung
 - Viele Joins
 - Performanceprobleme



OPM Toolbox

- Graphical Schema Editor
- Retrofitting Tool (View Based)
- Schema Translator
- Query Translator
- Webinterface Creator
- Doc Creator
- ...



4. XML

- Extensible Markup Language
 - Version 1: 1998
 - Einschränkung von SGML
 - W3C Standard
 - Kern eines Sprachzoo: XSL, Xpath, XQuery, XLink, ...
- XML: Sprache zur Definition von Markupssprachen
- Standard zu **Definition von Austauschformaten**

```
<Feature-tables>
  <Feature-table>
    <Reference>
      <RefAuthors>
        Moore W.S., DeFilippis V.R.
      </RefAuthors>
      <RefTitle>
        The window of taxonomic resolution ...
      </RefTitle>
      <RefJournal>
```

...

XML, DTD und XSchema

- Document Type Definition (DTD)
 - Definition erlaubter Entities und deren Attribute
 - kontextfreie Grammatiken, Erweiterte BNF
 - Keine Datentypen
- XML-Schema: Erweiterung
 - Constraints und Kardinalitäten
 - Primitive Datentypen
- XML Dokument ist
 - **Wohlgeformt**: Entspricht XML Syntax
 - **Gültig/valide** bzgl. einer DTD: Entspricht der DTD

```
<!DOCTYPE Bsm1 PUBLIC "-//EBI//Labbook, Inc. BSML DTD//EN" ...
<Bsm1>
  <Definitions>
    <Sequences>
      <Sequence id="MIVN83300" ic-acckey="U83300" title="MIVN83300" comment="Veniliornis nigriceps
strain LSU1305 cytochrome b gene, mitochondrial gene encoding mitochondrial protein, partial cds.
" length="946" topology="linear" molecule="dna">
        <Attribute name="version" content="U83300.1" />
        <Attribute name="organism-classification" content="Eukaryota; Metazoa; Chordata; Craniata;
Vertebrata; Euteleostomi; Archosauria; Aves; Neognathae; Piciformes; Picidae; Veniliornis" />
        <Attribute name="source" content="Veniliornis nigriceps LSU1305" />
        <Attribute name="database-xref" content="GOA:003345" />
        <Attribute name="database-xref" content="SPTREMBL:003345" />
      <Feature-tables>
        <Feature-table>
          <Reference>
            <RefAuthors>Moore W.S., DeFilippis V.R.</RefAuthors>
            <RefTitle>The window of ... </RefTitle>
            <RefJournal>(in) Mindell D.R. (eds.). AVIAN MOLECULAR EVOLUTION AND ... </RefJournal>
          </Reference>
          <Feature id="FTR_U83300.1_0" class="SOURCE" title="source" display-auto="1">
            <Qualifier value-type="strain" value="LSU1305" />
            <Qualifier value-type="organelle" value="mitochondrion" />
            <Interval-loc startpos="1" endpos="946" />
          </Feature>
        </Feature-table>
      </Feature-tables>
      <Seq-data>aantttggatctctcctagggcatttgcctaataacacaaattgtcacagg
cctcctgcttgccaccactacactgccgacacaaccctagccttttctt
ccgtcgccatacatgccgcaacggttcaatacggctgactaatccgtaac
ctccatgccaacggggcctcattctttttcatctgcatctacttacacat
...
    </Seq-data>
  </Sequence>
</Sequences>
</Definitions>
</Bsm1>
```

BSML / XEMBL

Elemente

- Datenmodell
 - Hierarchische Schachtelung von Elementen
 - Entspricht 1:N Beziehungen
 - Attribute, Datentypen (XML-Schema)
 - Constraints (nur in XML-Schema)
 - Assoziationen durch XPath und IDREF
- Schema = Formatdefinition
- Speicherung
 - Flatfile
 - XML Datenbank (Tamino, ...)
 - **Relationales Mapping**

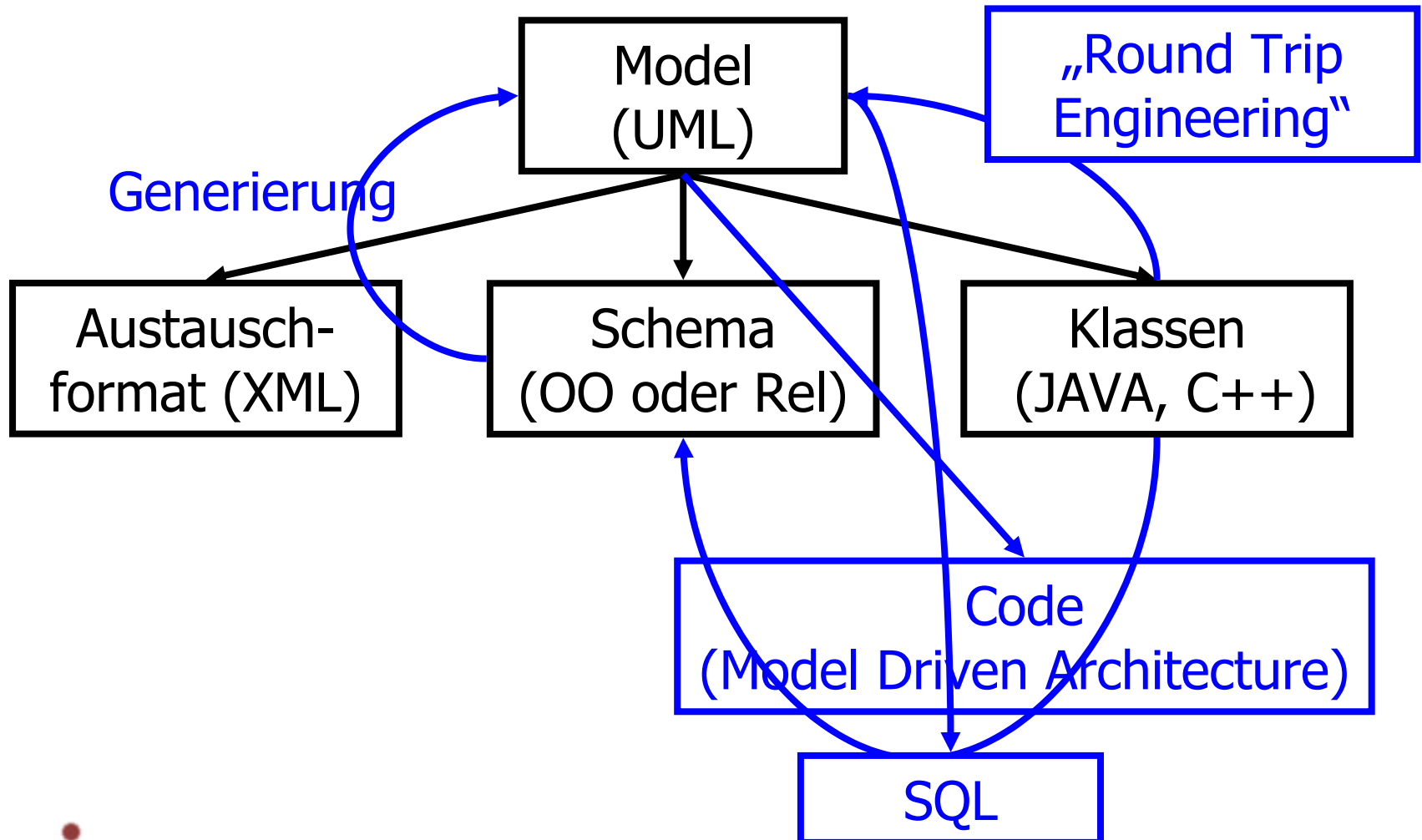
XML - Verwendung

- XML ist ein Austauschformat
- Keine Ablösung von Flatfiles als von Menschen lesbare Repräsentation
- Keine Ablösung von RDBMS als primäres Speichermedium
- Benutzung zur Verteilung computerlesbarer Daten

Bewertung

- Vorteile (gegenüber Entry-Based)
 - Industriestandard, viele Tools (Editoren)
 - DTD generierbar: UML, JAVA Serialisierung, ...
 - Effiziente Parser in allen Sprachen (DOM, SAX)
 - Unterstützung durch DB Hersteller (IBM, Oracle)
 - Strukturierte Anfragen (XQuery) und Textsuche möglich
- Nachteile
 - Dokumente sehr lang (Komprimierung)
 - Nicht wirklich lesbar (Stylesheets, XSLT, etc.)
 - Ohne DTD: Keine Validierung, keine Semantik für Datenaustausch
 - Mit DTD: Geringere Flexibilität, Dokumente ungütig bei Änderungen
- DTD / XML-Schema dient zur Definition von Standards

Ziel – und Probleme

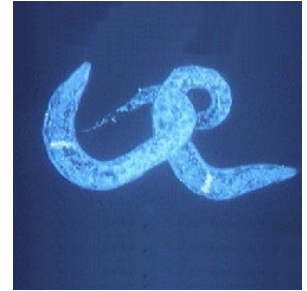


Zusammenfassung

- Austauschformate
 - Entry-based
 - XML
- Speichern und Anfragen
 - Relationales Modell
 - Objektorientiertes Modell (eher Software)
- Vorteile der Flatfiles nicht unterschätzen
 - Viele Labs ohne RDBMS / Informatiker
- I.d.R. verschiedenste Formate in einem Projekt

5. Nicht vergessen: ACeDB (Slides thanks to Ela Hunt)

- Kein HGP ohne ACeDB
- Entwicklung im Rahmen des C. elegans Mappings (von und für Biologen)
- **Schnellster Weg zur eigenen MDB + Anwendung**
 - Storage, Anfragen, GUI, Webserver, etc.
 - Frei, viele Plattformen (Mac, Linux, Windows)
- Datenmodell
 - „Schwach“ objektorientiert
 - Semistrukturiert (erstes?)
 - Konfigurationsdatei editieren = Schemaänderung



Datenmodellierung in ACeDB

Model : ?Map

Type **UNIQUE** Genetic
Cytogenetic
Physical

Colleague ?Colleague **XREF** Associated_map

Display Non_graphic
Title **UNIQUE** ?Text
Remark ?Text
Flipped
Unit **Text**

Centre **UNIQUE** **Float** **UNIQUE** **Float**
Extent **UNIQUE** **Float** **UNIQUE** **Float**

Contains Locus ?Locus **XREF** Map
Alleles ?Allele **XREF** Map
Chrom_Band ?Chrom Band **XREF** Map
Contig ?Contig **XREF** Map
Map ?Map **XREF** Map **#Map** position

Klassendef.

Varianten

Pointer, Assoziationen

Mischung von
Datenmodell
und Interface-Tags

Primitive Datentypen

Strukturierte Typen

Screenshots

ACEDB 4.8b, AGR

Save.. Edit.. Query.. Admin.. Help Exit

Status: Ready

Search in the selected class or in all objects named or related to

Class selection: Other class

- Chromosome Gene_name
- Map Gene_pro
- Map_population
- 2_point_data Locus
- RI_Mar
- Author Clone
- Colleague AGI_cl
- Paper Other_c
- Patent
- Journal Probe
- Allele
- Database Allele
- Model
- Method Germplasm
- View Image
- Species

Sequence: E

Selected DNA [0]

-17019 35671

Columns Zoom In.

EMBL:AL

DNA E

Struct

Databa

Origin

Subcl

Al

Ho

Vi

Co

At2g1034

At2g1035 EMBL:AC0

At2g1036

At2g1037

Query By Example: Paper

Quit Help Clear New KeySet Builder Command Edit

Search Undo

From: Class Class: Paper

ITEM NAME :

Reference	Title	
	Journal	
	Publisher	
	Editor	
	Page	
	Volume	
	Year	
	In_book	
	Contained_in	
	Medline_acc	
Author		
Affiliation		
Brief_citation		
Abstract		
Type		
Contains		
Refers_to	Locus	
	Allele	
	Rearrangement	
	Sequence	
	Strain	
	Clone	
	Protein	
	Expr_pattern	
	Cell	
	Cell_group	
	Life_stage	

Keyword

ACeDB Einordnung

- Informatiker hassen es, Biologen lieben es
- Ausgerichtet auf manuelles Editieren und Browsen
 - Queries sind schwierig (auch für Biologen) und langsam
 - Keine nebenläufige Datenbearbeitung (Dateisperre)
 - Keine API für Updates, Inserts, Deletes
 - „ACE Files“ mit obskurer Semantik
- Viele Gerüchte
 - Skalierbarkeit jenseits 1GB ?
 - Fehler in Datenmodelldefinitionsdatei (Tabs !) – Crash
 - Recovery?
- Nicht unterschätzen !