Archiving Dynamic Databases

Peter Buneman
University of Edinburgh
Sanjeev Khanna
University of Pennsylvania
Keishi Tajima
Japan Advanced Institute of Science and Technology
Wang-Chiew Tan
University of California, Santa Cruz

Digital Libraries grant IIS 98-17444
(NSF,DARPA,NLM, LoC,NEH, NASA)
http://db.cis.upenn.edu
http://db.cis.upenn.edu/Research/provenance.html

~~Archiving Dynamic Databases~~

Archiving Scientific Data

Database Group at Edinburgh

Malcolm Atkinson (Glasgow)

Persistence, performance, DB software, Gridology

Peter Buneman (Pennsylvania)

Data provenance, annotation, XML, scientific data

Wenfei Fan (Bell Laboratories)

Database constraints, data publishing, XML

Christoph Koch (Vienna)

Query languages, Logic, XML, Scientific data

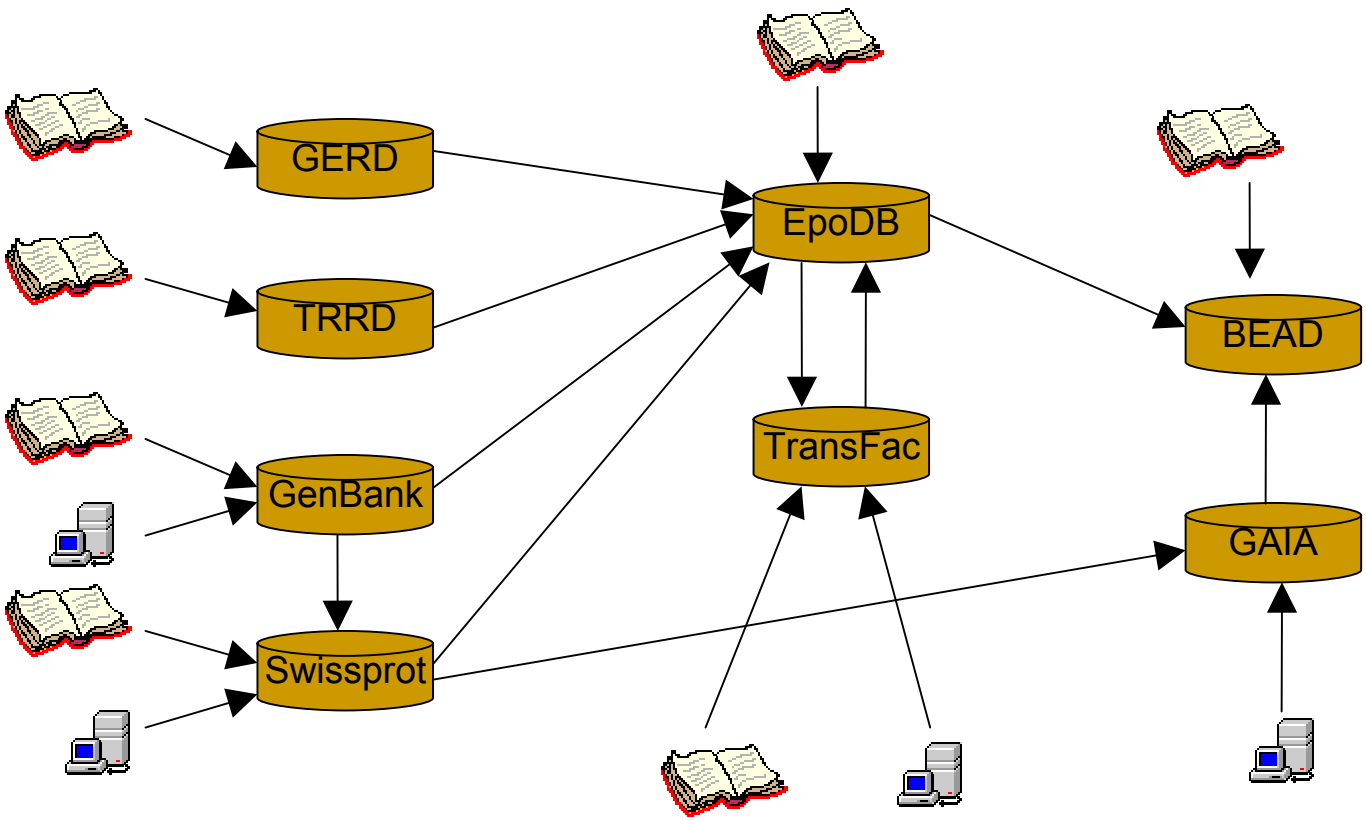
Stratis Viglas (Wisconsin)

Optimisation, data integration

Why does scientific data change?

- Mostly new data (accretion)
- It also gets *modified*
  - New/better experimental evidence
    - (scientific constants are surprisingly inconstant.)
  - New annotations (metadata?)
  - Propagation

Most molecular biology data is copied/transformed from other databases



The Importance of Archiving

If a new version of a database is created every day/hour/minute ...

... then we should archive it every day/hour minute

because someone might "cite" it.

Moore's law: £1000 now buys you 1GB/day – but horrible access problems

Keys

- A crucial part of any DB / file structure / file format design is the specification of *keys*. A.k.a:
  - unique identifiers,
  - locators,
  - canonical paths.
- File formats (notably XML) do a poor job of providing key specifications.
- Keys arise naturally in good DB design

The Structure of Keys

BL MS Cotton Nero A X

A manuscript in the British Library which used to be in Mr. Cotton's library (which burnt down) under a bust of Nero on the top shelf ten books along.

/timit/train/dr1/fcjf0/sa1.wav

SWISS-PROT: a curated database

```
ID   11SB_CUCMA     STANDARD;      PRT;   480 AA.
AC   P13744;
DT   01-JAN-1990 (REL. 13, CREATED)
DT   01-JAN-1990 (REL. 13, LAST SEQUENCE UPDATE)
DT   01-NOV-1990 (REL. 16, LAST ANNOTATION UPDATE)
DE   11S GLOBULIN BETA SUBUNIT PRECURSOR.
OS   CUCURBITA MAXIMA (PUMPKIN) (WINTER SQUASH).
OC   EUKARYOTA; PLANTA; EMBRYOPHYTA; ANGIOSPERMAE; DICOTYLEDONEAE;
OC   VIOLALES; CUCURBITACEAE.
RN   [1]
RP   SEQUENCE FROM N.A.
RC   STRAIN=CV. KUROKAWA AMAKURI NANKIN;
RX   MEDLINE; 88166744.
RA   HAYASHI M., MORI H., NISHIMURA M., AKAZAWA T., HARA-NISHIMURA I.;
RL   EUR. J. BIOCHEM. 172:627-632(1988).
RN   [2]
RP   SEQUENCE OF 22-30 AND 297-302.
RA   OHMIYA M., HARA I., MASTUBARA H.;
RL   PLANT CELL PHYSIOL. 21:157-167(1980).
CC   -!- FUNCTION: THIS IS A SEED STORAGE PROTEIN.
CC   -!- SUBUNIT: HEXAMER; EACH SUBUNIT IS COMPOSED OF AN ACIDIC AND A
CC       BASIC CHAIN DERIVED FROM A SINGLE PRECURSOR AND LINKED BY A
CC       DISULFIDE BOND.
CC   -!- SIMILARITY: TO OTHER 11S SEED STORAGE PROTEINS (GLOBULINS).
DR   EMBL; M36407; G167492; -.
DR   PIR; S00366; FWPU1B.
DR   PROSITE; PS00305; 11S_SEED_STORAGE; 1.
KW   SEED STORAGE PROTEIN; SIGNAL.
FT   SIGNAL        1     21
FT   CHAIN        22    480       11S GLOBULIN BETA SUBUNIT.
FT   CHAIN        22    296       GAMMA CHAIN (ACIDIC).
FT   CHAIN       297    480       DELTA CHAIN (BASIC).
FT   MOD_RES      22     22       PYRROLIDONE CARBOXYLIC ACID.
FT   DISULFID    124    303       INTERCHAIN (GAMMA-DELTA) (POTENTIAL).
FT   CONFLICT     27     27       S -> E (IN REF. 2).
FT   CONFLICT     30     30       E -> S (IN REF. 2).
SQ   SEQUENCE   480 AA;  54625 MW;  D515DD6E CRC32;
     MARSSLFTFL CLAVFINGCL SQIEQQSPWE FQGSEVWQQH RYQSPRACRL ENLRAQDPVR
     RAEAEAIFTE VWDQDNDEFQ CAGVNMIRHT IRPKGLLLPG FSNAPKLIFV AQGFGIRGIA
     IPGCAETYQT DLRRSQSAGS AFKDQHQKIR PFREGDLLVV PAGVSHWMYN RGQSDLVLIV
     FADTRNVANQ IDPYLRKFYL AGRPEQVERG VEEWERSSRK GSSGEKSGNI FSGFADEFLE
     EAFQIDGGLV RKLKGEDDER DRIVQVDEDF EVLLPEKDEE ERSRGRYIES ESESENGLEE
     TICTLRLKQN IGRSVRADVF NPRGGRISTA NYHTLPILRQ VRLSAERGVL YSNAMVAPHY
     TVNSHSVMYA TRGNARVQVV DNFGQSVFDG EVREGQVLMI PQNFVVIKRA SDRGFEWIAF
     KTNDNAITNL LAGRVSQMRM LPLGVLSNMY RISREEAQRL KYGQQEMRVL SPGRSQGRRE
//
```

Locators in SWISS-PROT?

```
ID   11SB_CUCMA     STANDARD;      PRT;   480 AA.
AC   P13744;
. . .
RN   [1]
RP   SEQUENCE FROM N.A.
RC   STRAIN=CV. KUROKAWA AMAKURI NANKIN;
RX   MEDLINE; 88166744.
RA   HAYASHI M., MORI H., NISHIMURA M., AKAZAWA T., HARA-NISHIMURA I.;
RL   EUR. J. BIOCHEM. 172:627-632(1988).
RN   [2]
RP   SEQUENCE OF 22-30 AND 297-302.
RA   OHMIYA M., HARA I., MASTUBARA H.;
RL   PLANT CELL PHYSIOL. 21:157-167(1980).
. . .
//
```

E.g. The second author (RA) of the first citation (RN=1) of the entry with accession number (AC) = P13744

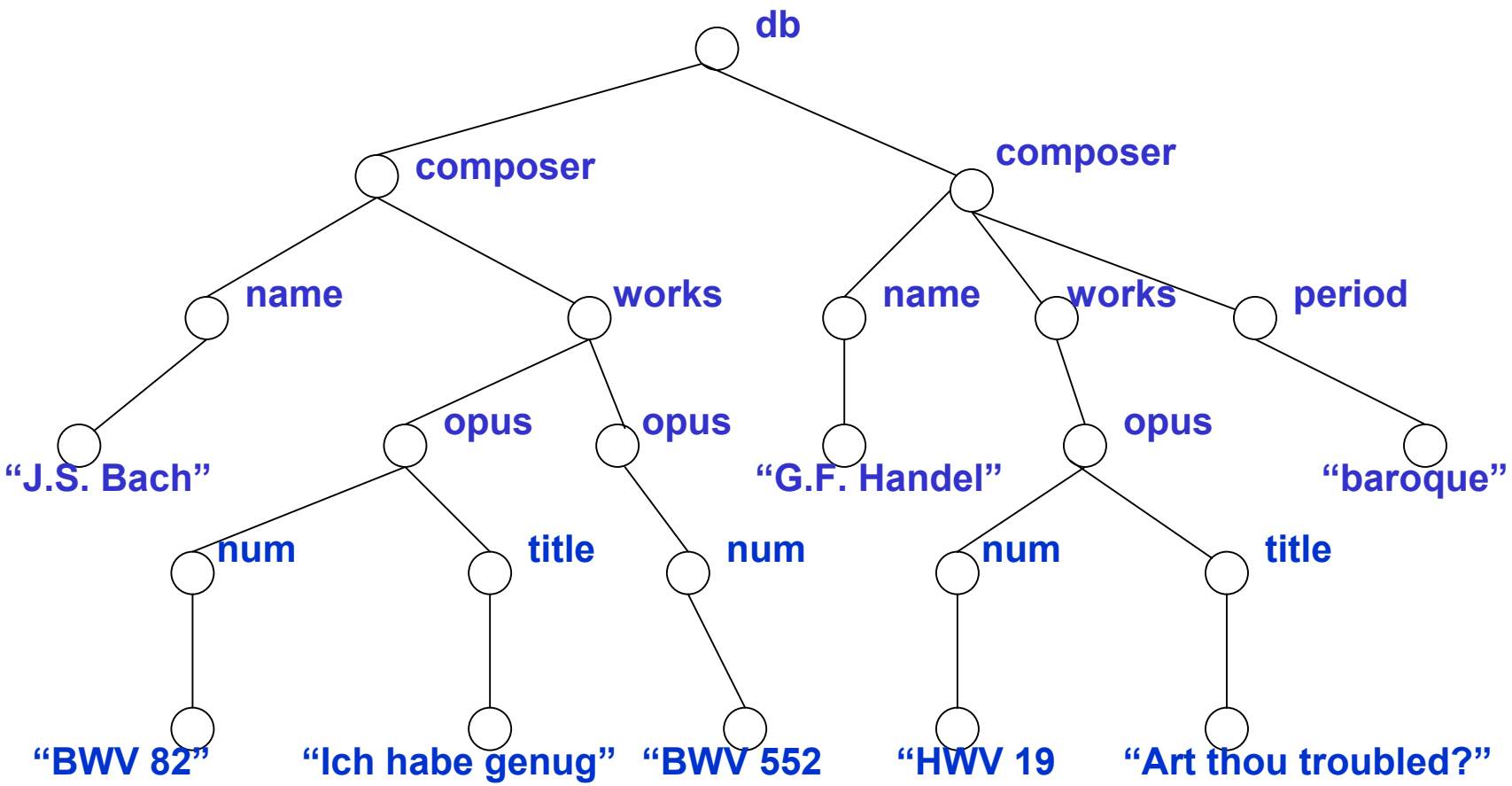
Note that this is a "fine-grain" locator

Specifying Locators/Keys

- They appear to be implicit in most scientific data sets
- They arise naturally in well-designed databases (e.g. from E-R diagrams.)
- Most data formats (notably XML) do not have any/adequate key specification languages

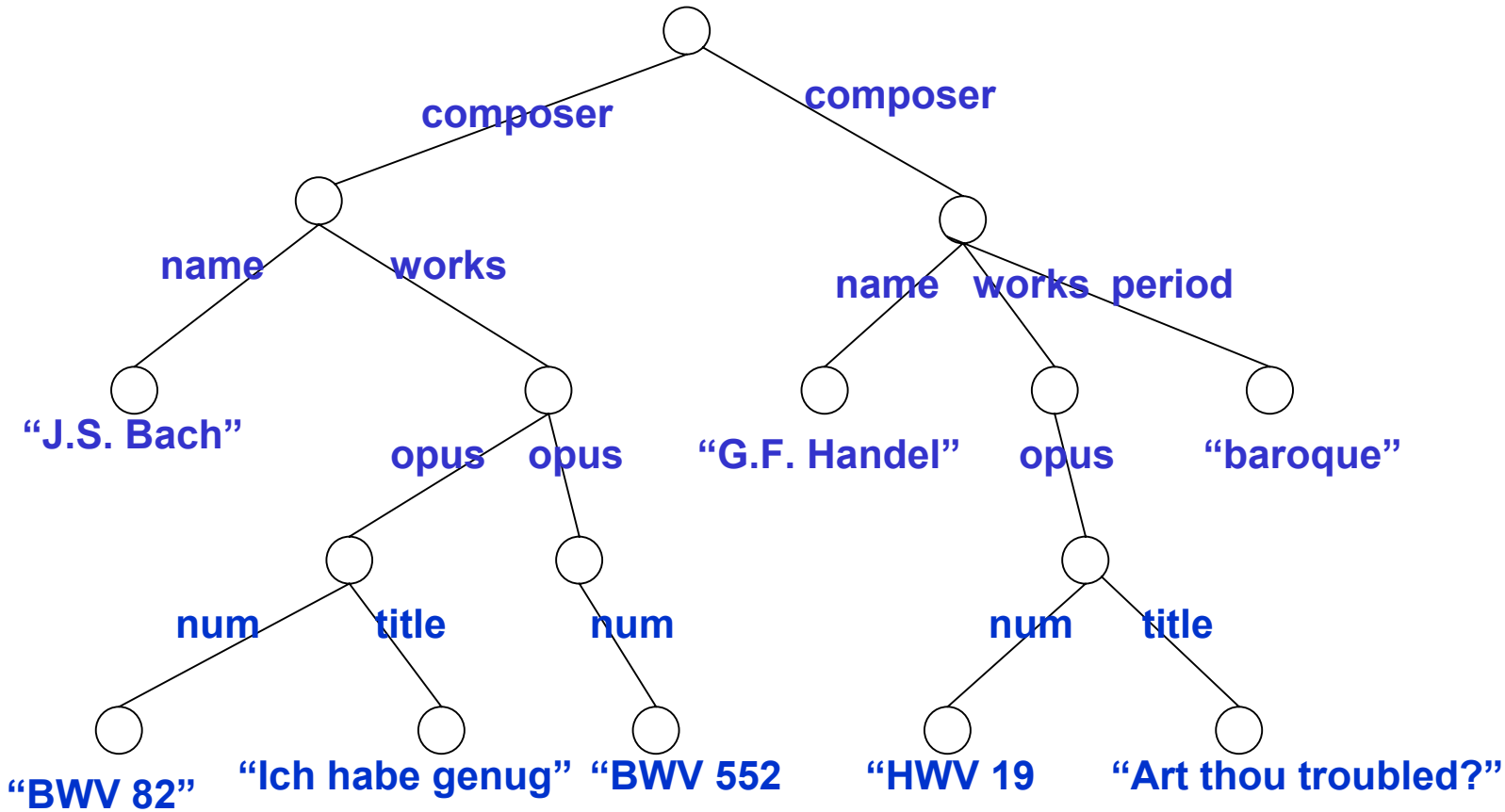
We need a method for specifying keys in hierarchical data...

Semistructured data: node-labeled as in XML



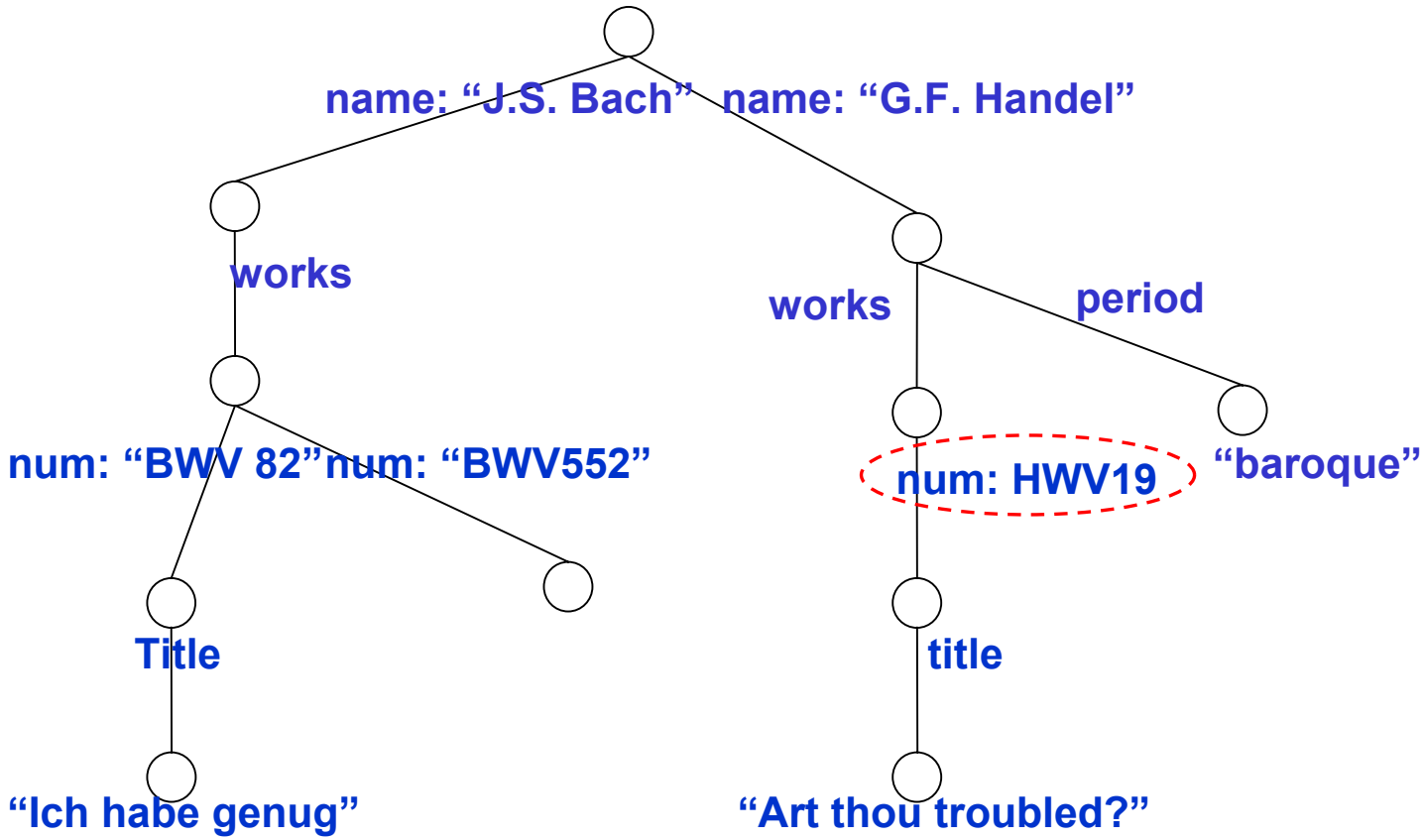
[order is important in XML]

Semistructured data: edge-labeled as in UnQL, XML-QL



[These systems mostly ignore horizontal order]

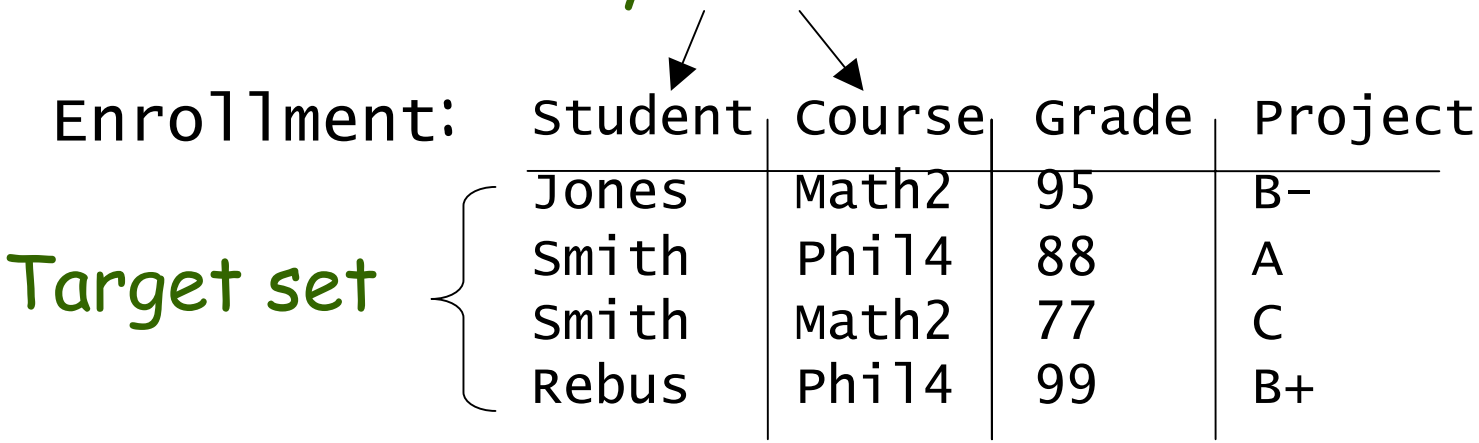
Semistructured data: deterministic model



Keys for XML (Davidson, Fan, Hara, Tan)

- Implicit keys are ubiquitous in scientific data formats (easily converted to XML)
- Some proposals for key specifications in XML work (DTD IDs, XML-Schema)
- "Deep citation" in digital libraries.
- Natural consequence of translating back from deterministic model to XML (node-labeled)

Keys for Relational DBs



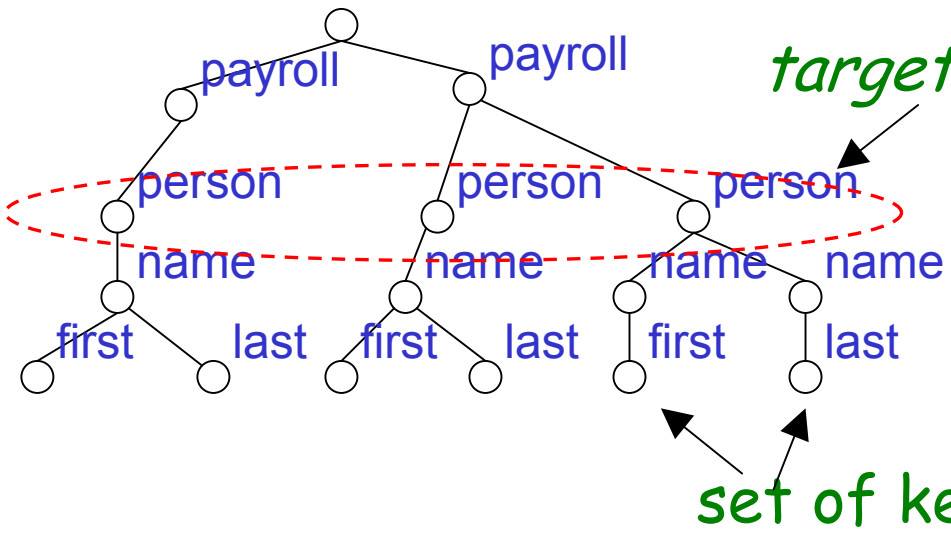
- Keys are critical in database design
- Keys are used to build indexes (optimization)
- Need to understand key inference

Key specification for node-labelled formats

General form: $(Q \{P_1, \ldots, P_n\})$

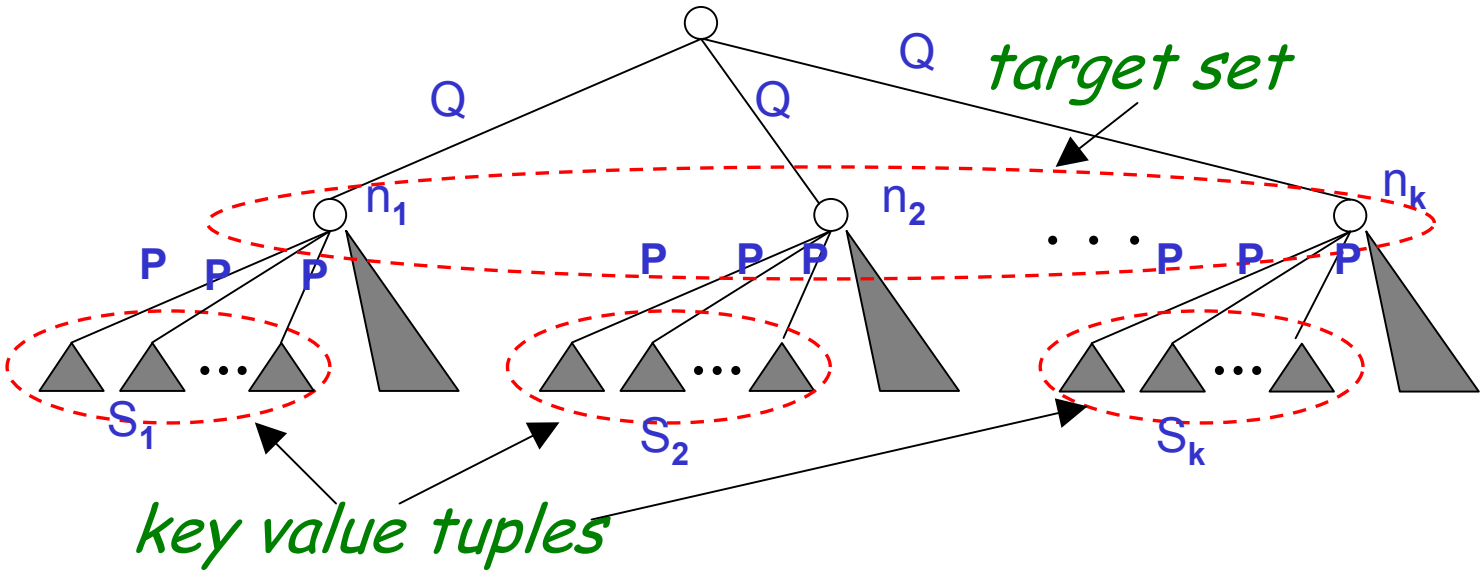
path expressions
(correspond to attributes)

Example: /payroll/person{name/first, name/last}



Does not impose uniqueness on payroll nodes

Meaning of a key spec. (single key path (Q{P}))



"Value" equality

nodes identical

- If $S_i \cap S_j$ nonempty then $n_i = n_j$
- ($|S_i| = 1$ ["strong" keys])

Relative keys

General form: $Q\{P_1, \ldots, P_n\}/Q'\{P'_1, \ldots, P'_{n'}\} \ldots$

Example:

/book{name}/chapter{number}/verse{number}

number specifies chapter *only* within book

number specifies verse *only* within chapter

Also:

/bible{}/book{name}/chapter{number}/verse{number}

empty key: at most one bible node

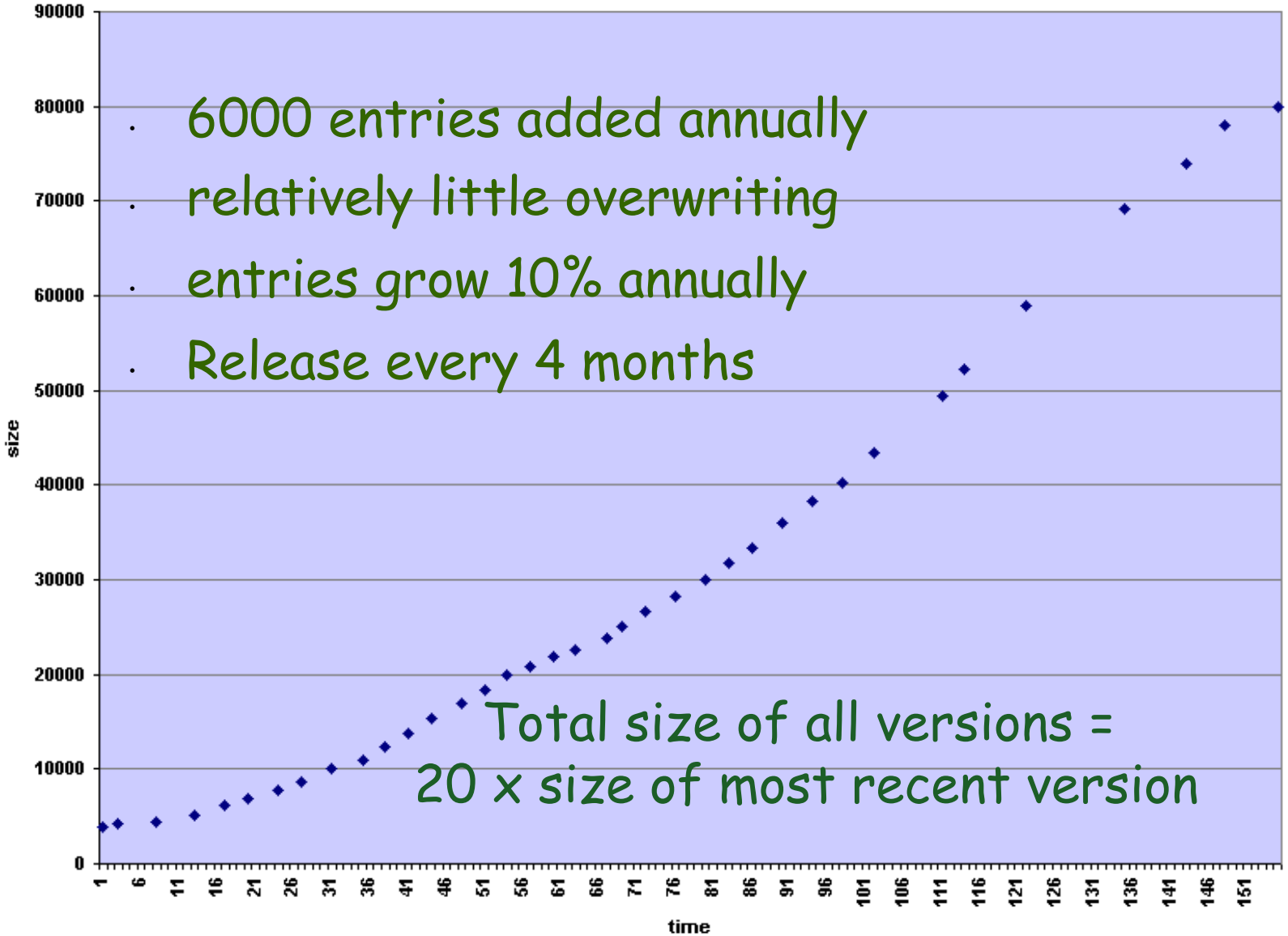
Notes on Keys

- Proposals here have been incorporated into XML schema (probably a bad idea!)
- Closely related to (interfere with) data models:
  - payroll{}/employee{id}/[name{}, sal{}, …] (something like a "complex object"/nested relational model)
  - Recent decidability results by Davidson, Hara, Fan and Libkin
- Lots more to study. Inference for relative keys (now partly done), foreign keys …

How do we Build Archival Databases?
[Khanna, Tajima, Tan]

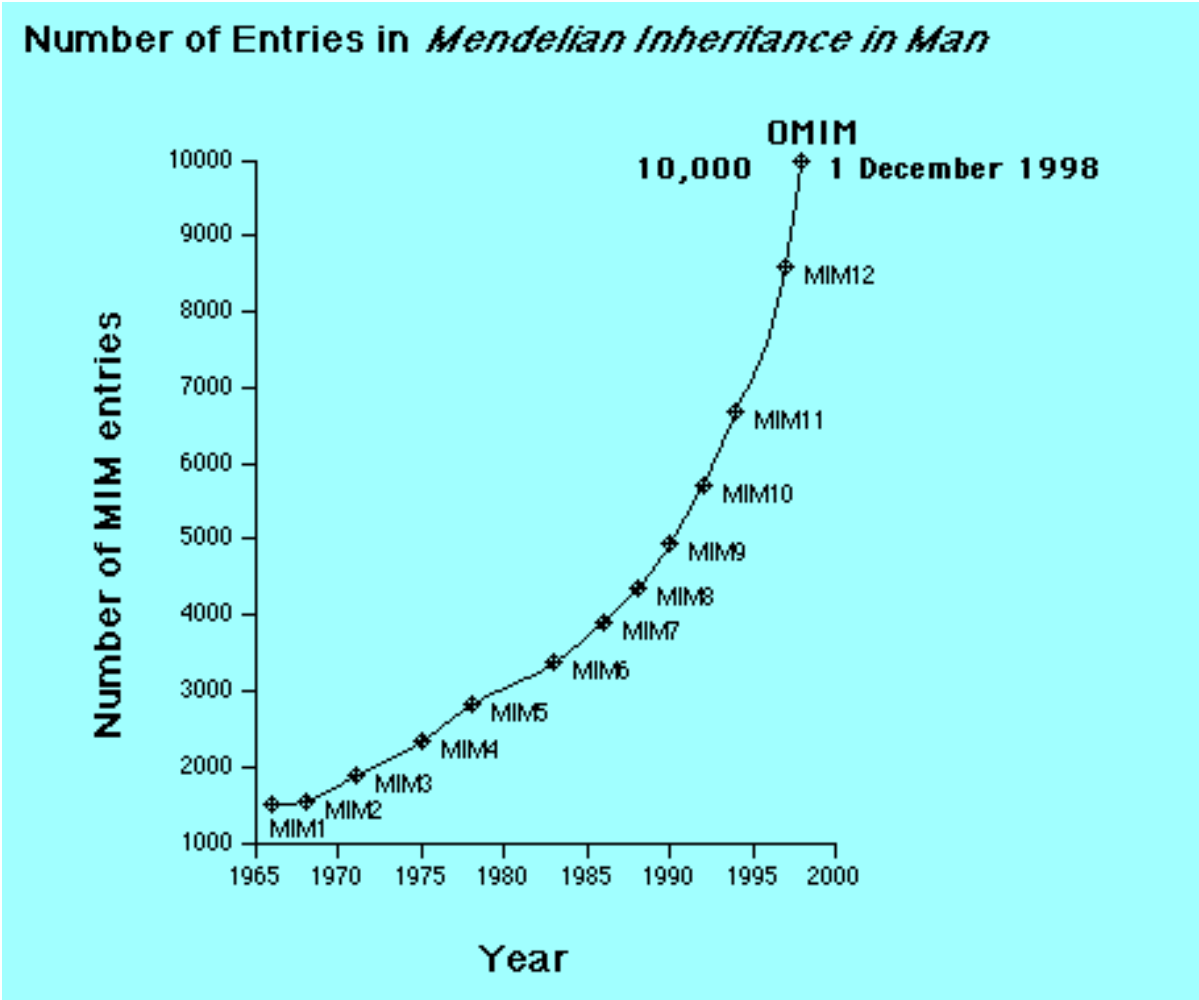
- Many scientific databases keep *archives*. It's important to preserve the state of knowledge as it was in the past
- Archive frequently: space consuming
- Archive infrequently: delay in getting recent information published.

Swissprot


6000 entries added annually
relatively little overwriting
entries grow 10% annually
Release every 4 months
Total size of all versions =
20 x size of most recent version
size
time
0
10000
20000
30000
40000
50000
60000
70000
80000
90000
1
6
11
16
21
26
31
36
41
46
51
56
61
66
71
76
81
86
91
96
101
106
111
116
121
126
131
136
141
146
151

Online Mendelian Inheritance in Man

- Printed editions stopped in 1998
- Updated daily!



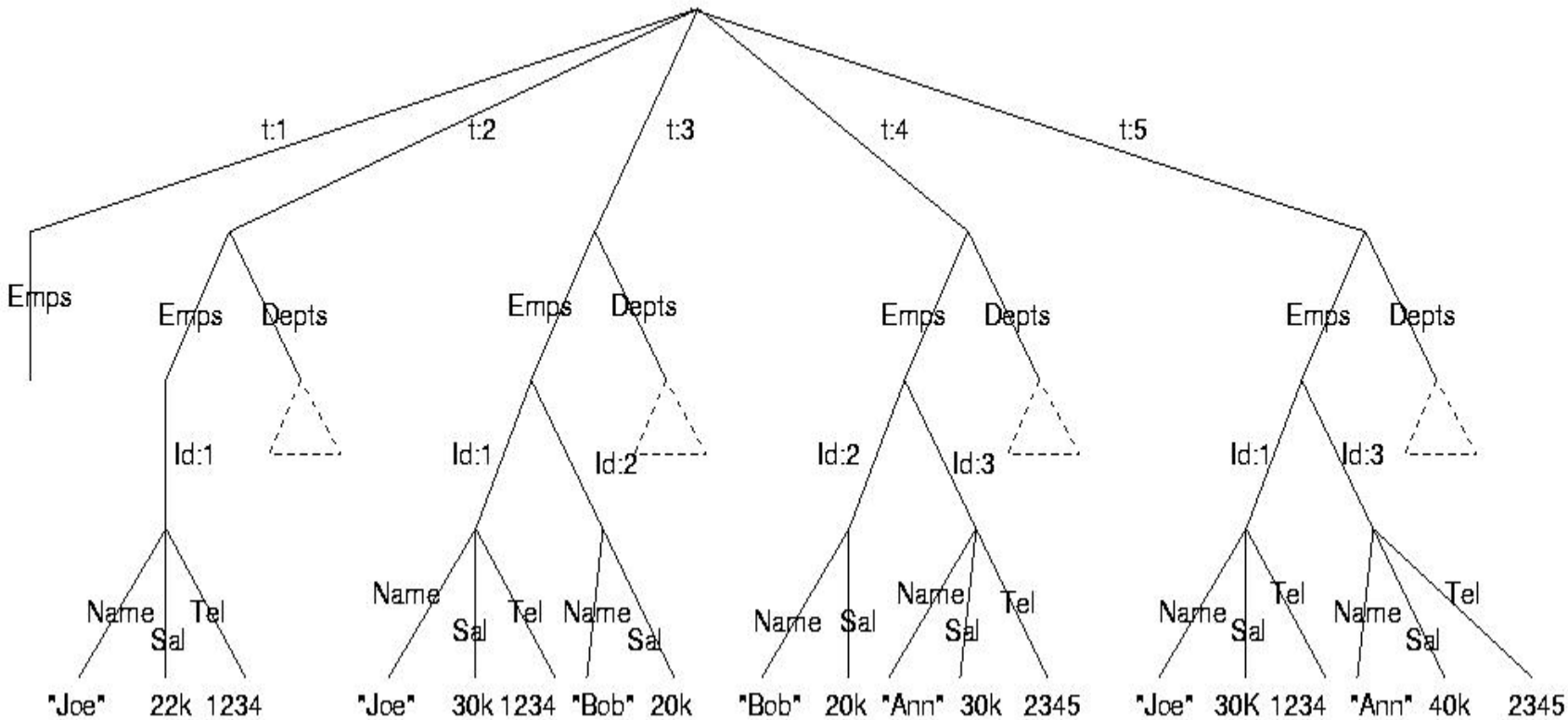
OMIM vs. Swissprot

- Both valuable *curated* databases
- Similar gross structure -- sequence of entries, each with internal structure
- Swissprot:
  - All past versions available
  - Slow release -- every 3-4 months
- OMIM
  - Past versions unavailable
  - Rapid release -- every day (or more often)

Why not use diff?

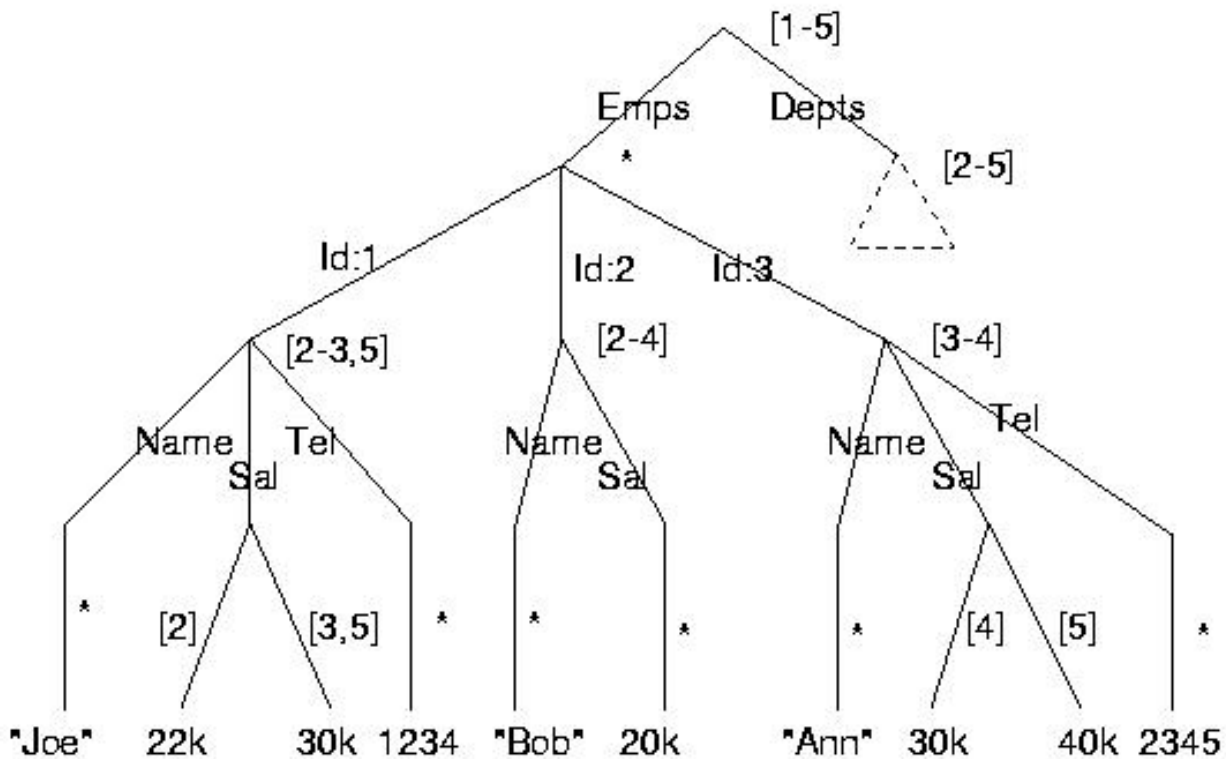
- Diff currently used for archival part of CVS
- Tree diffs have not yet come to market
  - Line diffs work well on formatted XML
- Diffs do not preserve "object-hood"
- Expensive to unwind 365 diffs

A Sequence of Versions



We use keys to obtain a deterministic model

"Pushing" time down



[Driscoll, Sarnak, Sleator, Tarjan: "Making Data Structures Persistent."]

An initial experiment

- Recorded all OMIM versions for about 14 weeks (100 of them)
- XML-ized all of them
- Combined into archive XML format file by pushing time down.
- Also recorded diffs between versions
- Did the same the same thing for the last 20 available versions of Swissprot

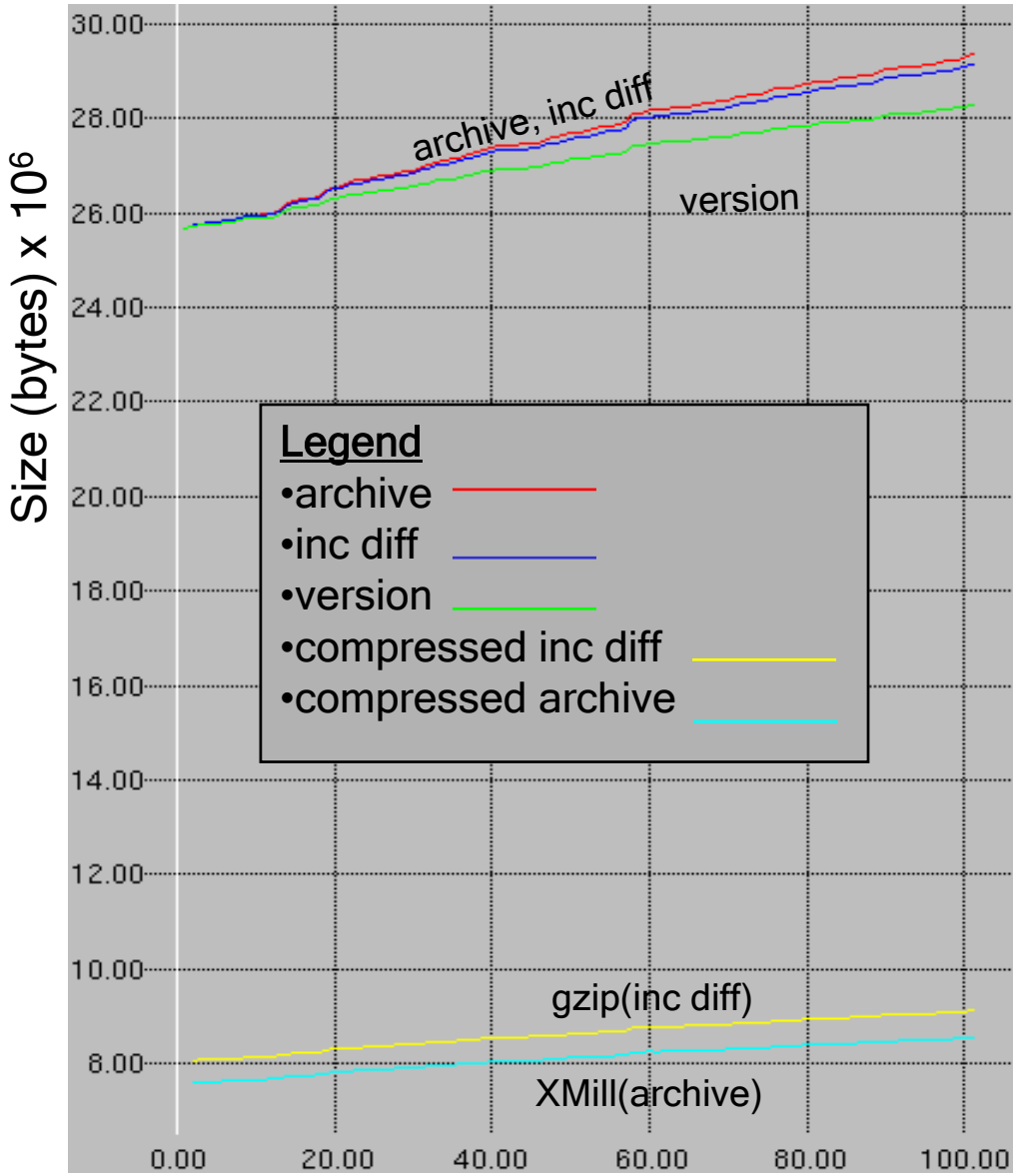
100 days of OMIM

Uncompressed

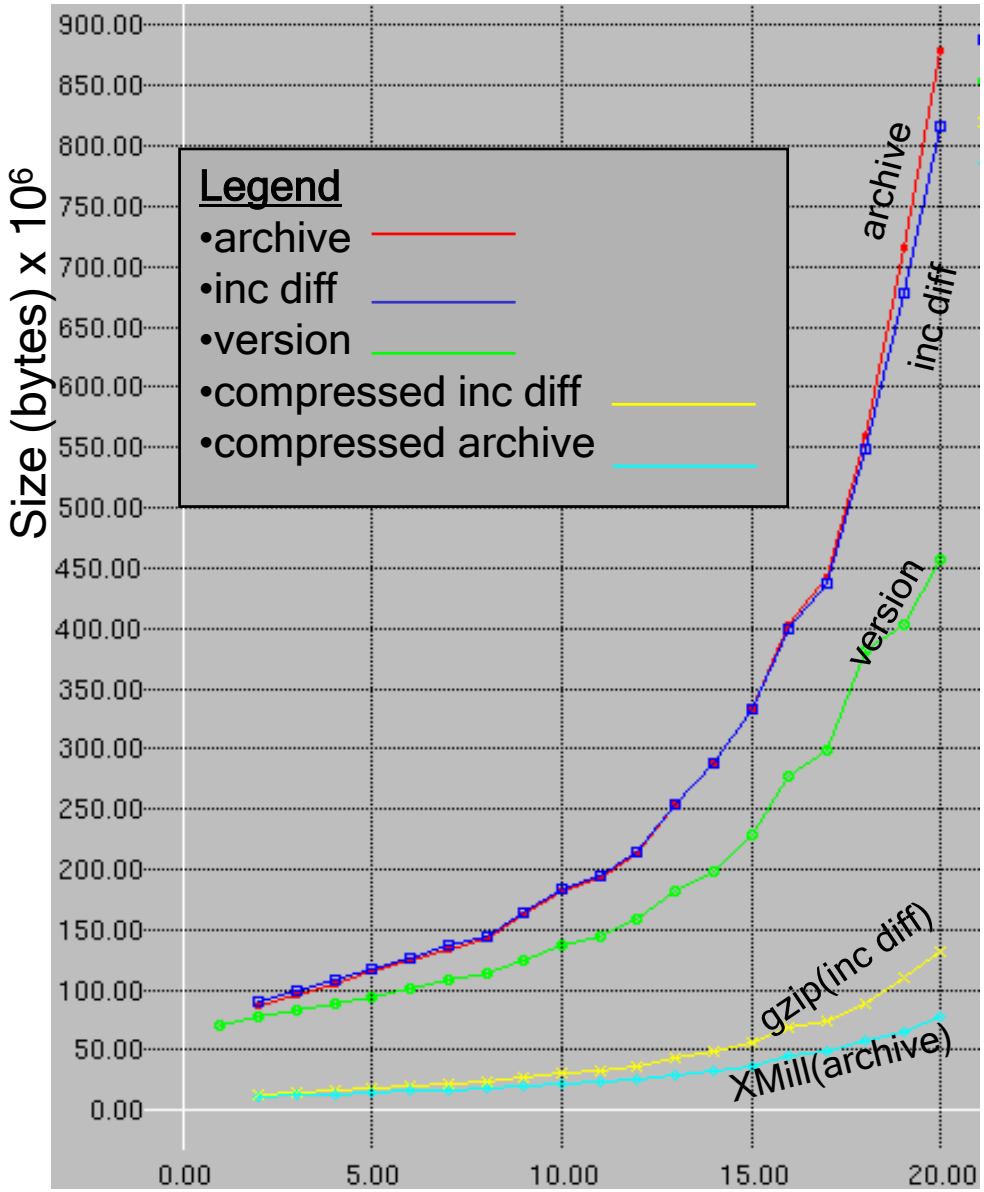
- Archive size is
  - ≤ 1.01 times diff repository size
  - ≤ 1.04 times size of largest version

Compressed

- archive size is between 0.94 and 1 times compressed diff repository size
- gzip - unix compression tool
- XMill - XML compression tool



~ 5 years of Swissprot



Uncompressed

- Archive size is
  - ≤ 1.08 times diff repository size
  - ≤ 1.92 times size of largest version
- Compressed
- archive size is between 0.59 and 1 times compressed diff repository size

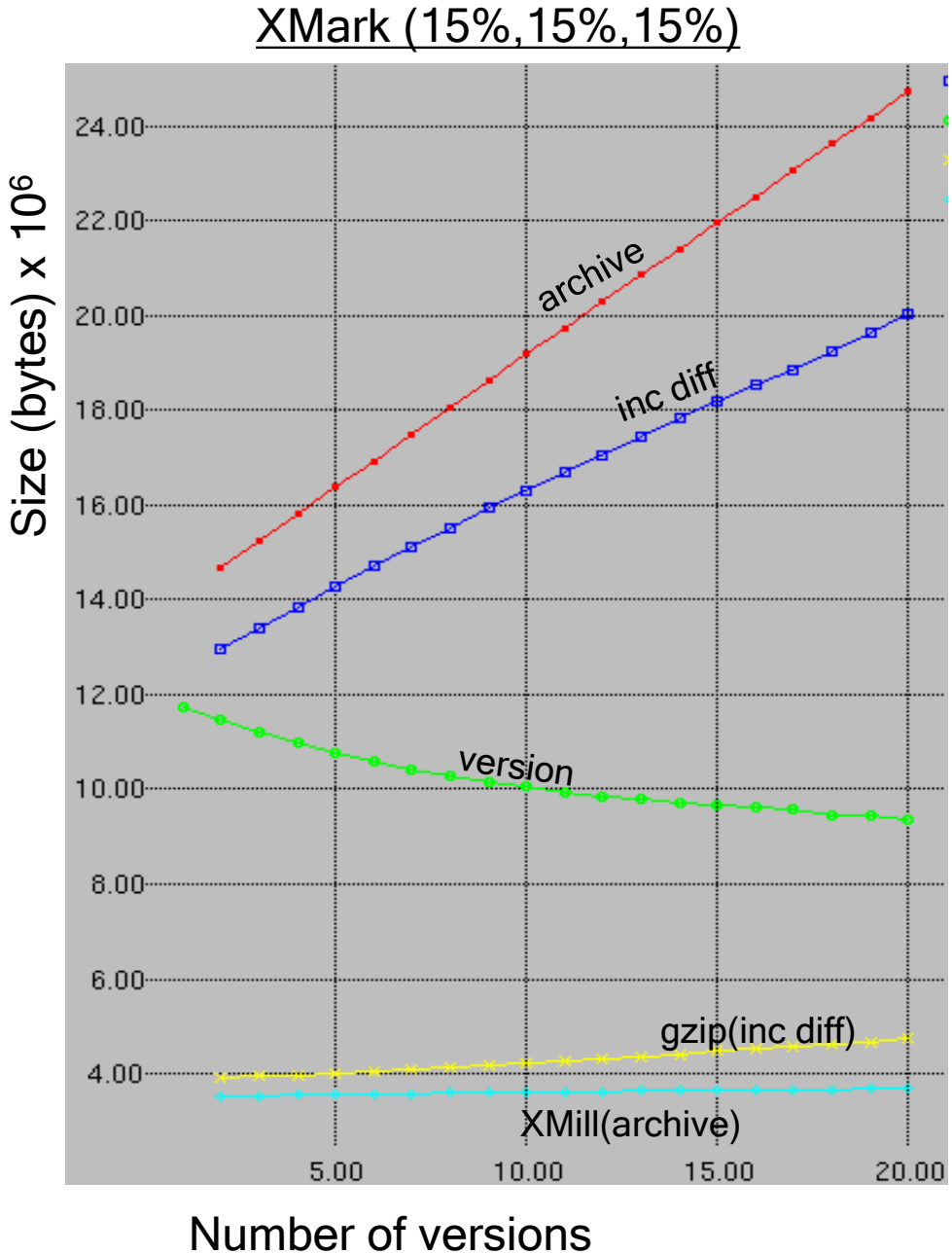
Synthetic XMark Data

Uncompressed

- Archive size is
  - ≤ 1.23 times diff repository size
  - ≤ 2.11 times size of largest version

Compressed

- archive size is between 0.78 and 1 times compressed diff repository size



The Bottom Line

- Can archive a whole year of Swissprot or OMIM with < 15% overhead (size of most recent version)
- Retrieval is a linear scan of archive
- Works well with compression.
  - Down to 30% of most recent version.
- Archive as often as you like! (Almost)
- Permits temporal queries on objects

Further work…

- What to do when regions of data are unkeyed?
  - present system reverts to diffs.
- "Discovering" keys for archiving
- Keyed (a.k.a. deterministic) models have also been used for file/view synchronization
- Useful for "deep" citation?
- Could they hold the "key" to other aspects of data curation? (Models for provenance and annotation.)