Anchor Modeling
A Technique for Information under Evolution

Lars Rönnbäck #AMweek 28/6 to 1/7, 2011
Graph showing relative occurrence of words in literature over the last century

Information is rapidly becoming the most important asset
“Panta rhei”
Everything flows

Heraclitus
500 BC
Evolving Information

- Changing content
- Changing structure
- Changing constraints
- Changing interpretation
- Changing origins
- Changing reliability

There's a big difference between saying: "This information has a 95% reliability" and "This information is 100% reliable".
What is a database?

- The purpose of a database is to store a body of information and allow searches over it.

- The purpose of a temporal database is to store a body of information under evolution and allow historical searches over it.

But, we are not there yet!
What is a Data Warehouse?

- Integrates information from many sources
- Keeps a history of changes
- Provides “one version of the truth”
- Enables reporting, ad-hoc analysis, mining
- Calculates and stores new information
Many sources and many users naturally result in many changes.
Patch or Redo?

- Patching works initially to cope with new requirements.
- Maintenance costs usually rise proportionally to the lifetime of the data warehouse.
- Redoing is unavoidable at some point (and for dimensional modeling sometimes accounted for).
- The average lifetime is five years.
- The return of investment should and could be much better with a longer lifetime!
What is Anchor Modeling?

Anchor Modeling combines normalization and emulation to provide an agile database modeling technique for evolving information that is implementable in current relational databases.

Most, if not all, of what Anchor Modeling is doing in its physical (relational) representation could be "hidden" from the end-user in a true temporal database.
Technologies

- Entity-Relationship Modeling
- Sixth Normal Form Tables
- Temporal Database Emulation
History

Best Paper Award @ ER'09

Paul Johannesson
Lars Rönnbäck
Olle Regardt
Maria Bergholtz
Petia Wohed

consulting

DW MDM EDW DW TDWI WWW ER09 TOOL AMW

03 04 05 06 07 08 09 10 11

research
Philosophy

- Make modeling free from assumptions
- Make modeling agile and iterative
- Make evolution non-destructive
- Do not duplicate information
- Do not alter existing information
- Decouple metadata from the model
- Provide a simple interface for queries
- **Changing content**
  - full support [6NF + time of change]

- **Changing structure**
  - full support (through extensions) [non-destructive schema evolution]

- **Changing constraints**
  - minimal support [only primary and foreign keys]

- **Changing interpretation**
  - achievable [explicitly modeled]

- **Changing origins**
  - restricted support [using metadata]

- **Changing reliability**
  - restricted support [using metadata]
Domain driven modeling

Data driven modeling

Use-case driven modeling

Data Vault/ODS/3NF (Inmon)

Anchor Modeling

Dimensional Modeling (Kimball)
Basic Notions

**Attributes** – properties
Example: The surname of a Person
<#42, ‘Rönnbäck’, 2004-06-19>

**Anchors** – entities
Example: A Person
<#42> (holds only identities of entities)

**Knots** – shared properties
Example: The gender of a Person
<#1, ‘Male’> + <#42, #1>

**Ties** – relationships
Example: The children of a Person
<#42, #4711>

**filled = primary**
Naming convention

- **Anchors**
  Two letter mnemonic + descriptor
  \[\text{PE} \_ \text{Person}\]

- **Knots**
  Three letter mnemonic + descriptor
  \[\text{GEN} \_ \text{Gender}\]

- **Attributes**
  Inherited anchor mnemonic + three letter mnemonic + inherited anchor descriptor + descriptor
  \[\text{PE} \_ \text{SUR} \_ \text{Person} \_ \text{Surname}\]

- **Ties**
  Inherited anchor and knot mnemonics separated by the roles they play in the relationship
  \[\text{PE} \_ \text{child} \_ \text{PE} \_ \text{parent}\]
Historization

Historization is done using the time of change as the start of an interval implicitly closed by another instance of the same identity with a later time of change.

closed interval
historical information

open interval
current information

<#42, ‘Samuelsson’, 1972-08-20>
<#42, ‘Rönnbäck’, 2004-06-19>
<#42, ‘Sommare’, 1982-03-01>

asynchronous insert

Note that UPDATE is never allowed or needed in an anchor database.
Temporal perspectives

- **Latest perspective**
  Shows the latest available information

- **Point-in-time perspective**
  Shows information as it was on the given timepoint

- **Interval perspective**
  Shows information changes that happened within the given interval

- **Natural perspective**
  Converts natural to surrogate identities
  (for composite identities these may span over several anchors)
Table elimination

A table $T$ can be removed from the execution plan if:

a) no column from $T$ is explicitly selected

b) the number of rows in the returned data set is not affected by the join with $T$

The temporal perspectives regain all benefits from 6NF!
Indexing

- All primary indexes are clustered indexes (index organized tables) which use no extra space.

- Secondary indexes are very rarely needed.

Clustered index on identity + historization.
1. The data is time dependent and its changes need to be kept.
2. The data is sparse, i.e. many entities lack some of their attribute values.
3. The number of distinct data values is small compared to the number of all data values.
4. Identifiers constitute a small portion of the total amount of data.
5. There are many identifiers.
6. There are many properties.
7. Searches address relatively few of the properties in the entities over which the search is done.
8. Searches include conditions that impose bounds on values.
Example model
Changes become extensions of the existing schema.

all previous versions of the schema are available as subsets of the current schema.
Tracking changes

- **Schema**
  Table containing the schema in XML format and when it was activated

- **Anchor, Knot, Attribute, and Tie**
  Views shredding the XML to columns over which detailed changes can be tracked

- **Evolution**
  Table-valued function comparing the schema valid at the given timepoint with the current database tables
Modeling anchors

Guideline 1: Use anchors for modeling core entities and transactions.

relational implementation:

![Anchor diagram]

anchor
Guideline 2a: Use a historized attribute if versioning of attribute values are of importance, otherwise use a static attribute.

relational implementation:

historized attribute

static attribute
Modeling attributes

Guideline 2b: Use a knotted static attribute if attribute values represent categories or can take on only a fixed small set of values, otherwise use a static attribute.

relational implementation:

knotted static attribute
Modeling attributes

Guideline 2c: Use a knotted historized attribute if attribute values represent categories or a fixed small set of values and the versioning of these are of importance.

relational implementation:

knotted historized attribute
Modeling ties

Guideline 3a: Use a historized tie if a relationship may change over time, otherwise use a static tie.

relational implementation:

historized tie

static tie
Modeling ties

Guideline 3b: Use a knotted static tie if the instances of a relationship belong to certain categories, otherwise use a static tie.

relational implementation:

![Database schema diagram]

knotted static tie
Guideline 3c: Use a knotted historized tie if the instances of a relationship belong to certain categories and the relationship may change over time.

Relational implementation:

knotted historized tie
The Modeling Tool

- Open Source
- Online (HTML5)
- Free to use
- In the Cloud
- XML Interchange Format
- Automatic generation of SQL scripts
- Interactive (force-directed) Layout Engine

www.anchormodeling.com/modeler
Inserting data

- **OLTP-like requirements**
  Insert trigger on the latest view where an identity is created if not provided

- **DW/OLAP-like requirements**
  Stored procedure generating the given number of identities

These are also automatically generated by the tool
Important Benefits

- Handles evolving information *(keeping the integrity intact)*
- Increases longevity *(databases with long life expectancy)*
- Simplifies modeling concepts *(less prone to error)*
- Enables modular and iterative development
- Needs no translation logic to the physical layer
- Automates generation of scripts
- No downtime when upgrading databases
- Scans only relevant data during searches
- Sparse data cause no gaps *(no null values)*
More Information

Homepage: http://www.anchormodeling.com
Blog . Video Tutorials . Modeling Tool

Twitter: anchormodeling

E-mail: lars.ronnback@anchormodeling.com

LinkedIn Groups:
Anchor Modeling
Temporal Data Modeling
Temporal Data

re!sight
insight • change • value