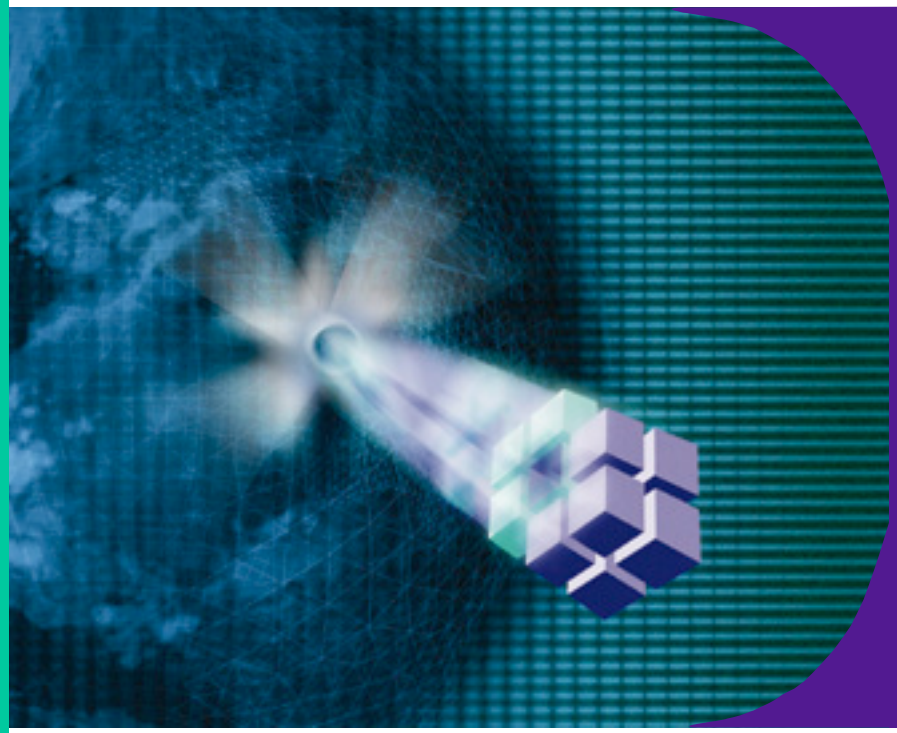
CINCOM Smalltalk™



Object-Relational Persistence in Smalltalk

Alan Knight (knight@acm.org)

Cincom Systems of Canada

CINCOM

What is GLORP

- Generic Lightweight Object-Relational Persistence
- Open-source library for O/R mapping
  - LGPL(S)
- Camp Smalltalk project
  - Sponsored by The Object People (2000)
  - Portable
- Planned as one part of major overhaul of Object Lens

Motivating Example

- Cincom internal application
- Widely used internally and by customers
- Critical
- Relatively simple model
  - 19 tables
  - somewhere between 12 and 46 domain classes
- Client-server (although server functionality desirable)

Issues

- Schema changes extremely difficult
- "Interesting" schema
- Performance critical
- Some operations over very high-latency links
- Significant amounts of data
- Versioning schema, heavily linked data
- Must support multiple databases

- Demo

Relationship Example


```
(aDescriptor newMapping: ToManyMapping)
    attributeName: #methods;
    referenceClass: StoreMethodInPackage;
    useLinkTable;
    join: (Join
        from: primaryKeyField
        to: methodsTable packageRefField).

methodMapping query alsoFetch: [:each |
    each definition].
methodMapping query expectedRows: 1000.
```

- Note that Join defines read, write, and join.
- Note optimizations

Packages and Classes

- In Store, "Package" means one version
- "Class" means ClassDefinition
- Entities in database are not like traditional in-memory entities
  - e.g. what package contains this method
- Introduce additional entities
  - StoreVersionlessPackage
  - StoreMethodInPackage (etc)
  - StoreClassExtension
- Clearer domain model, ability to express querie

Demo 2

CINCOM Smalltalk™

- Querying UI
- Replicating
- Browsing
- Optimizations
- Caching

Writing

- Unit of Work
- Purely transactional
- No explicit writes
- In-memory rollback
- Automatic insert/update
- Automatic write order/referential integrity
- Mostly automatic database transactions
- Optimized

Notable Technical Points

- Object-level rollback in Unit of Work
- RowMaps and write optimization
- Block to expression conversion
- Joins
  - declarative - defines the table relationship
  - simple
  - handles all uses

Acknowledgements

- The Object People
- Cincom
- All the contributors and users of GLORP

References

- GLORP
  - http://www.glorp.org
  - http://glorp.sourceforge.net
- General
  - Ambler: Object Primer, http://www.agiledata.com (good emphasis on importance of both worlds)
  - Fowler: Patterns of Enterprise Application Architecture (good patterns, once you ignore the non-domain model stuff)
  - Fabian Pascal: Practical Issues in Database Management (pure relational extremist)
  - ROE - Avi Bryant
  - Gemstone

Reading non-Object Data

- Reading pure data, ordering

  query := Query readManyOf: StorePackage.

- Aggregate functions

  query orderBy: [:each | each name].

  query retrieve: [:each | each name distinct].

  query retrieve: [:each | each primaryKey max].

- Retrieving pieces of objects

  query retrieve: [:each | each id].

  query retrieve: [:each | each name].

  query retrieve: [:each | each address] (changing contexts)

- Note: All internal queries generated by user-accessible mechanisms.

Change Hats: VisualWorks

- Next-generation database frameworks, inputs
  - VisualWorks Object Lens
    - Strong in many respects, but very dated
    - Client-server orientation
  - Object Studio POF
    - Very strong modelling
  - GLORP
    - Open-source
    - Extremely flexible mapping layer
  - SQLWorks
    - Good server orientation
    - *very* high-performance
- Goal: Synthesize the best of all these

Core Issues

- Object identity vs primary keys
- Pointers vs. foreign keys
- Networks of objects vs. rows
- Queries vs. traversing relationships
- Encapsulation vs. program independence
- nil is not null
- The role of the application

Incidental Issues

- Keys: natural vs. generated
- Integrity constraints
- Inheritance
- Mismatched schemas
- Agility

General Database/Multi-User Issues

- Scaling
- When to read data?
- What do queries look like?
- Performance, performance, performance
- Locking
- Caching, refreshing, keeping data in sync
- Transaction semantics (knowing which transaction to use)
- Transaction lengths
- Multi-user within an image?

Approaches

- Metadata or code generation
- Associating objects with transactions
- SQL, OO query language, objects as queries, special syntax
- Explicit or automatic writes
- Marking objects dirty
- When to take objects out of cache
- Different framework architectures (e.g. brokers, subclassing)

GLORP Licensing

- LGPL (Lesser GNU Public License)
  - Can be used as a library, does not affect linked code
  - Modifications must be released under the same license
- But…
  - LGPL's terminology very difficult to interpret with respect to Smalltalk.
- Approach
  - Include a clarification with GLORP, specifying a sensible Smalltalk interpretation of the license
  - Copyright held by the authors/not assigned
  - Counsel: Lawrence Rosen (also general counsel of OSI)