Store

Alan Knight

Engineering Manager – Cincom Smalltalk

Cincom Smalltalk™

Cincom®

Overview

- Concepts
- Cincom Usage
- Current work
  - Tools
  - Database back-end
  - Loader
- Future plans

Concepts

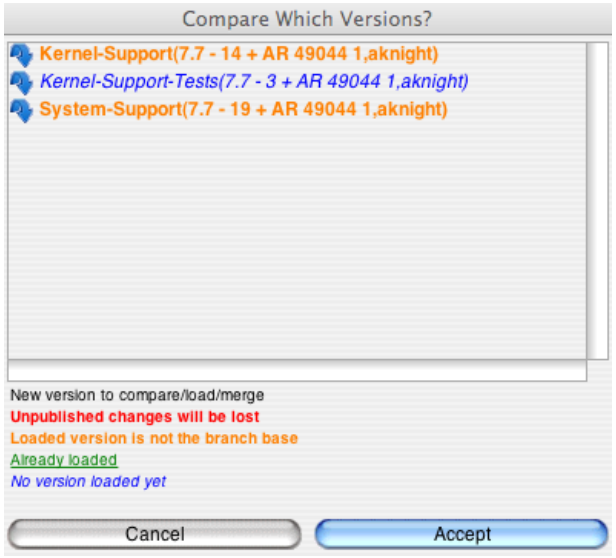
- Cincom Smalltalk source code control system
  - VisualWorks, ObjectStudio, WebVelocity
- Started as alternative to ENVY/Manager
- Relational database back-end
- Central database
  - But “replication” allows more distributed usage
- State rather than changes

Internal Cincom Processes

- Action Request (AR)
  - MARS system
- Trunks and branches by naming conventions
  - 7.7.1 – 132
  - 7.7.1 – 132 + AR 55555 3
- All code changes in branches, merged onto trunk when AR code passes review
- Oracle, SQL Server, Postgresql – with replication

…Cincom Processes

- Store AR Support
  - Compare/Load/Merge by AR
  - Package-level operation
  - Tool support in IDE
    - Knows which AR is being worked on
    - Automated support for naming convention
  - Very minimal and simple…
  - …. But very useful
- Shared integration image for base
- “head” scripts



Current Work

- Tools
- Atomic Loading
- Database Back-End
- Garbage Collection
- Constraints and Indexes
- Miscellaneous

Tools

- Major revisions of Store tools (still very much in progress)
  - Inevitable disruptions, but hopefully progress
- Problems
  - Old browser framework
  - Inconsistent
  - Poor internals, Poor UI
  - Performance
- Browsing with RB
- Code Comparison
- Merge, GC

Loading

- Load via file-in
  - Compile failures lead to broken image
  - Order dependencies
  - DLL/CC
  - Do-it-yourself brain surgery
- Atomic Loading
  - Temporary namespaces for compiling code
  - All code installed at once
  - Brain surgery made easy

Atomic Loading Caveats

- Not all brain surgery is easy
- Adding/Changing compilers
  - Later code needs to be compiled using earlier code
  - Install Early
    - Some cases can be recognized (Parser subclasses)
    - Package property #installBeforeContinuing
  - Package granularity
- Other dependencies possible

Analysis Loader

- Upcoming Work
- Current loader goes strictly in package order
- Instead, look at the entire bundle being loaded
  - Sort definitions by required load order
- Makes it much easier to organize code as desired
  - Packages as categories versus packages as independently loadable entities

Database

- Various schema issues
- Naïve object-relational mapping
  - Hard to optimize for schema
  - Hard to change schema
  - No real model
- Moved to GLORP framework, StoreForGlorp
- Numerous advantages, but tradeoffs
  - Space vs Time
  - Boundaries of caching

Misc

- Database constraints and indexes
  - Help ensure integrity
  - Avoid two packages with same version name
  - Some queries much faster
- Much better treatment of edge cases with Overrides
- History preserved across package renames
- Consistently use server timestamps
- More automation-friendly

Future Possibilities

- Further tools improvements
- Schema changes
- Optimizations
- Configuration management
  - Work with non-loaded components
  - Logic for variations
  - Level independence
- Many possibilities

Demo


20 sticks of spaghetti
+
one yard tape
+
one yard string
+
one marshmallow

Thank You!

Bonus slides if time permits

Database- Schema

- Significant schema issues
  - Performance (Blobs, lack of indexes)
  - Semantics
    - Methods, class definitions, directly connected to package
    - Concept of class extension must be constructed
    - Painfully slow
  - Excess stuff (e.g. separate metaclass definitions)

Database - Model

- Weak model
  - Directly coded to database entities
  - Naïve O/R mapping
  - No resemblance to Smalltalk meta-objects
  - Difficult for tools to work with
- StoreForGlorp
  - Much stronger model
  - Still somewhat constrained to the database entities
  - Much closer to meta-objects