TRACK 3: CLIENT/SERVER
Session 420

CLIENT/SERVER DESIGN AND IMPLEMENTATION ISSUES

Wilson Hitchings
TIER Corporation

TIER

BAA Design Guidelines

Case Sensitivity

- Use mixed case but be aware of collating sequence problems
- Will permitted values be mixed case?
- What about address? Some address verifyiers return upper case only, and US Postal regs are single case text for bulk rates
- What about conversion of legacy system data?

TIER

BAA Design Guidelines

Relationships

- Use descriptive verbs for relationship names, avoid “relates_to”, “associates”
- Never use same/similar relationship names between same two entity types
- Always add description on source and destination!
- Name suggestion list:

supplies/is_supplied_by	requests/is_requested_by
services/is_serviced_by	sells/is_sold_by
returns/is_returned by	contains/is_contained_by
employs/is_employed_by	

BAA Design Guidelines

Permitted Values

- Very restricted usage
- Good candidates are Days of Week, Months of Year, flags, gender status (may change!), compass directions (N,S,NW, etc.), certain coded fields
- GUI presentation makes them more attractive - radio buttons, drop downs
- Remember to include <spaces> as valid entries for optional fields and resetting GUI drop-downs!

BAA Design Guidelines

Standardized Fields

- Person's Name - hyphenated and international names can be long, plan to accommodate
- Soundex - a discrete field value calculated from a text field; update when source field is updated
- Date Formats - use 12/31/4000 as max date field for compatibility with DB2, Oracle and Sybase. Don't show max date value to user!

BAA Design Guidelines

Standardized Fields, con't.

Address components

- U.S. Postal standards qualify for bulk rates, but limit international mail
- States of US and Provinces of Canada are unique and can be in same table with no collisions
- Zip Code validation not for the faint of heart
- Many sites purchase address verification/correction software, so make sure your address fields are compatible with verification package

BAA Design Guidelines

Standardized Fields, con't.

SSN

Common format choices:

- 9 digit edit pattern on a numeric field is a popular choice
- Composite of 3 fields with no imbedded hyphen
- 11 byte character field, with edit routines to remove/display hyphens

BAA Design Guidelines

Standardized Fields, con't.

Telephone components

- Extension - 5 bytes
- Don't forget Country Code if you must support international customers
- Toll-free (800) numbers aren't overseas!
- Will you support vanity (textual) numbers like 1-800-ADW-SUKS?

BAA Design Guidelines

Identifiers

- Name them IDENTIFIER or ID in data model
- 9 digit, system generated
- 3 digit sequence numbers for children entities
- 20-char timestamp (but can't be randomized by IEF code) may not be available in all DBMS's
- Displayable Hex, or base 34 conversion can be used for large integers

Processes and Activity Hierarchy

Classic Activity Hierachy Guidelines

- Single set of Create, Update, Delete Elementary Processes for each entity
- One create point for an entity for the entire model
- Entire entity included in EP view
- Single Owner of all entity actions

Processes and Activity Hierarchy

Practical Activity Hierarchy Guidelines

- Process Model should stand separate from data model (except for Expected Effects)
- Multiple creation and update points for any given entity are valid
- AHD is a secondary toolset to an Entity Life Cycle Diagram, because it's entity states that drive processes, not their database actions!

Processes and Activity Hierarchy

Classic Elementary Process definition

- Must update database (no read-only EP's)
- Must leave data model consistent
- Must be atomic, cannot be sub-divided (EP's cannot call EP's!)

Processes and Activity Hierarchy

Practical Elementary Process definition

- Most logical unit of work is probably an Action Block that calls multiple EP's
- EP generally should be a self contained, re-usable piece of an Action Block
- Single Entity-to-EP rule does not reflect business rules being executed
- EP complexity based on process, not on data model

Processes and Activity Hierarchy

Practical Elementary Process definition

- Concept of single EP for update of an entity not realistic given starved views and overlapping processes
- Entity actions must be consistent (effects same entities every time) but optional attribution OK
- Entity actions should be the same after each execution of an EP; no flags that change processing

Processes and Activity Hierarchy

Practical Elementary Process definition

I/O Elementary Processes have several problems

- 100% sensitive to all Data Model changes, even the addition of new optional fields
- Opposite of starved view approach
- Requires calling processes to read all data in entity, even if it is insiginficant to the calling procedure
- Security issues surround access of sensitive data fields - everyone can see you!

Processes and Activity Hierarchy

When to create a sub-EP Process

- When logical action must be included but is not necessary according to the data model
  - » Logging of audit records
  - » Logical enforcement of mutually exclusive relationships or optional relationships
  - » Printing request table, asynchronous tasks

Processes and Activity Hierarchy

When to create a sub-EP Process

- When multiple processes contain identical code that could be broken out into an EPAB
  - » Audit records, logging, etc.
  - » Setting date fields based on a status
  - » Determining an error message based on parameter input

BSD Design Guidelines

Action Diagram Naming Standards

- Some sites prefix type designator with Business System Prefix
- Procedure & Procedure Step (Client/Server/Batch) - CSP_? or BAT_?
- Private, single-use Action Blocks - PRIV_?
- Action Blocks - AB_?
- Elementary Processes - EP_?
- Elementary Process Action Blocks - EPAB_?
- Externals - EXT_?
- Global and Common Access (GABs and CABs)

BSD Design Guidelines

View Name Standards

- In, Import, Input
- Out, Export, Output
- Local or work
- <unnamed> Entity Action unless required for clarity (old, new...)
- Some sites suffix “Persistent” to view name for clarity

BSD Design Guidelines

Exit States

- Most sites have unique prefix on both name of Exit State and text
- Some sites suffix exit states with termination property
  - » _RB - Rollback
  - » _AB - Abort

BSD Design Guidelines

DSD Naming Standards

- Make DSD names unique in DM
  - » unnecessary for code generation
  - » use a 4-char prefix to make unique
  - » must be managed like member names
  - » eliminates confusion and mistakes!
  - » greatly assists in field-level help implementation

Common Object Management

Data Model Independence Issues

- CABs should use work views as much as possible for Input/Output (allows some degree of isolation)
- Version control for common routines - detraction of sharability but must be done
- DB2 packaging - Composer 4 will support(?)
- TI found 90% of all text fields would fit into 30 characters; standardized lengths on next projects could significantly reduce length-based changes to modules

Common Object Management

Common, Shared Routines and Applications

- Identifier Generation
  - » Random, sequential, timestamp
- Tables and Codes - don't forget archived table entries for retro changes
- Postal Code verification
- Phone Number/Area Code verification

Common Object Management

Ownership & Responsibility

- Direct migration between development models if developer owned (peers)
- Migration to Common Object model first if committee owned (parent)
- Central "frozen" model concept works well for phased projects

Common Object Management

Clone areas and Shared Applications

- Address - ownership and isolation more important than saving space
- Text handling & presentation
  - » GUI Text usually one long string utilizing word wrap (causes problems with printing; word wrap routine not accessable by code)
  - » Block mode text handling best done by deletion and re-add of entire comment block
  - » If string search function necessary, use FIND function against text lines (make case *in*-sensitive)
  - » Think about standard text lenghts - 65 char, 1K and 3K sizes

Identifier Recommendations

- BAA identifiers are pretty unreliable
- Much higher percentage of generated identifiers in an implemented model than one might think
- Generated identifiers should never be part of business data - enablers only
- Common Objects team MUST provide generation routines!

Identifier Recommendations

Generated identifiers have four general sources:

- Sequentially ascending/descending numeric value
- Sequentially assigned but randomized numeric value
- Random Number generator (based on timestamp or seed value)
- Dithered alpha or combination of pieces of other fields

Identifier Recommendations

Sequentially ascending/descending value

- **Pro:** guaranteed unique, can be range assigned, no looped inserts
- **Con:** contention/bottleneck on stored value at create time, may cause hotspot on user entity in high-insert situations

Identifier Recommendations

Sequentially assigned (randomized)

- **Pro:** guaranteed unique, eliminates "hotspot" problem on end-user entity, is suitable for partitioning index
- **Con:** contention/bottleneck on stored value at CREATE time

Identifier Recommendations

Random Number generator (seed value)

- **Pro:** No hotspot problem, no database record to contend with, extremely fast
- **Con:** not unique, requires looped inserts, may require user to re-attempt insert, hard for non-Composer conversion routine to duplicate value

Identifier Recommendations

Dithered alpha / pieces of other fields

- **Pro:** No database record to contend with, fastest to assign
- **Con:** May require sequence number to make unique, dependence on other fields not readily visible, may be longer than desireable

Deletion approach

Myth: Deletes on-line are BAD

Truth: Unqualified Deletes are BAD

Typical situations that effect deletion performance

- Cascade of related entities
- Non-unique index impact
- Legacy/external system reconcilliation issues

Deletion approach

Performance Solutions

- Soft Delete - defers deletion to better time
- Encapsulated Business Areas - breaks cascade
- Unique Indexes only - no index sweeps upon delete
- Batch-only physical delete - avoids hand-coded RI but is an overnight process

Soft Delete Details

- Useful to defer deletion of key entity or prevent cascade during day
- Delete can be restored in certain situations
- Similar in processing to Archiving or end-dating
- Requires more Action Diagram support than Hard Delete

Soft Delete Details

- Hand-written Referential Integrity required
  - » READs must be written to ignore a "deleted" record, and don't forget QMF!
  - » Relationships must be nullified where necessary, or related entities also marked as deleted
  - » Update locks required to enforce RI are the same or greater than physical RI impact!

Error Handling

Error approaches are:

- Find-one-get-out
- Warnings vs. Fatals
- Accumulated or percentage-of-error (usually batch)

Error Handling

- Commit capabilities of environment effect decision
  - » Most TP environments use message get/put as synchpoint
  - » Some environments may support sub-transaction units of work (Forte or CICS)

Error Handling

- Most logical and easiest to code is all-or-nothing
- Data Concurrency decision will impact error handling
- Error Message should apply to highlighted field(s) only
- Cursor placement in field insufficient visual cue for error failure in GUI

Error Handling

- Video properties do not persist in block mode, but they do in GUI!
- Restoration of imports to exports may be necessary, so be careful in your events
- Normally the user error is re-displayed to reduce correctional keystrokes
- Progressive updates to screen will have to be reset when error encountered

Data Concurrency

Definition: “a point of reference for a datum that can be used to ensure the context of the datum is the same as when initial conversation started.”

- Essentially a guarantee of accurate data via a context point
- Necessary or not? Depends on significance of update...

Data Concurrency

- Inversely effects transaction concurrency under certain conditions
- High update frequency to related records may cause failure
- Shouldn't the context point cascade to related records (parent-children)?
- What about denormalized fields?

Data Concurrency

How to implement

- Counters - easiest and most secure (roll-over of counter is automatic in Composer)
- View comparison - ideal implementation but hard to code (multiple IFs)
- Date/Time - less secure than counter but requires no new fields as overhead if a "Last Update and Time" is kept

Archiving of Data

Definition: non-destrutive removal of data

- Why archive?
  - » Legal requirement
  - » Space problem - off-load aging data
  - » Restrict processing to certain records
- Extract or Archive in place?

Archiving of Data

How live applications are effected

- READs and UPDATEs must ignore archived data
- Shape of database will effect performance
- Code table entries may require immortality!
- Update of archived record should be allowed, so pick lists must understand archived data to restrict/allow an entry

Retrieval of archived data

- Data can rarely be restored without some conversion once it leave the database
- Identifiers may not be guaranteed unique anymore
- Data Model changes may make restoration difficult or impossible

Retrieval of archived data

- Support data (code table entries, related records) must be restored also
- Alternative to archive may be report format or microfiche
- Storage of archived data may not be on-line, so restoration requests may have to be queued

Walkthrough Recommendations

Goals for Walkthrough

- Promote re-usability, reduce redundancy
- Promote standards and guidelines
- Provide rudimentary completeness checking
- Identify design issues before they get expensive to fix
- Catch someone doing something right!

Walkthrough Recommendations

Walkthrough Preparation

- Procedure to be reviewed must be supplied, with all applicable documentation, 24-48 hours prior to Walkthrough
- Too may details of code can be hidden by printed version, so .DAT files are used by many sites
- Worthwhile for some sites to separate GUI review from code/design review

Audience Input

- Best Practices
- Things to avoid
- Productivity Tips and Techniques
- Extensions and add-ins to Composer
- Exchange of e-mail addresses, web pages, technical documents, etc.