Non-Event Databases

(Objectivity databases for all non-event data)

- Objectivity databases for non-event data
  - Conditions
  - Ambient
  - Configuration
  - Spatial and Temporal
- Common features
- Design description
- Current status
- Outstanding issues

Common Features

- All are implemented using the same general BaBar tools
  - Persistent object base class
  - Transaction management
  - Object placement
  - Access control
  - Code is reused whereever possible
- Data access in code is buffered by proxies
  - User code interacts only with transient classes
  - Proxies transform persistent objects to transient in a hidden, automatic way
  - Non-event database Proxies are kept in the environment (static event)
- Data access is provided to dataflow by a download system
  - Reverse data transport
  - Conditions and Configuration data download use the same interface and implementation

Conditions Database

- Tracks time-dependent and quasi-static data
  - Calibration, alignment, geometry, BField map,...
- Assigns a time interval over which a data object is <u>valid</u>
  - Validity interval is a separate object from the data
- Assigns indepedent timelines to independent data
  - Validity intervals for a ‘logical’ object span from 1900 to 2037
- Allows multiple versions of data for the same validity interval

Conditions Database Status

- Conditions DB is in widespread use
  - Detector geometry + alignment is completely defined in the cond DB
  - Every subsystem has at least 1 calibration type defined, with data
  - Some flat files are still in use (online)
- Performance seems adequate
  - Write speed consistent with online calibration goals
  - Read in Opr doesn’t dominate startup time, or scale badly with nodes
- Diagnostic tools are (barely) adequate
  - Calibration data browser GUI
  - Low-level interval utilities
- Support is good (1 FTE software engineer)
- Datastream calibration is broken by split federations
  - Conditions written by Opr are lost in the sweep
  - Could be patched by a smarter sweep
    - import only online calibration containers (containers only need to move 1 way)
    - requires ‘level 3’ data distribution

Ambient Database

- A 'stripped down' conditions database
  - Same base code
  - Versioning is disabled
- Used to store online detector monitoring data

Ambient Database Status

- Fully deployed for online monitoring
- Archiver tuning is still in progress (to reduce data volume)
- Performance is adequate <u>when no other locks interfere</u>
  - Archivers run 7X24, and need 'continuous' write locks
  - Read locks (from Opr for instance) block the archivers
- Split federations forces access latency of 1-2 days
  - Users aren't allowed to read from the IR2 federation
  - Data sweep is done daily
  - More frequent data sweeps?
  - Incremental sweeps? (level 5 conditions data distribution)
- Reco access (proxy) was provided late
  - no running code accesses ambient data yet
- Ambient data are <u>indistinguishable</u> from conditions
  - Adds overhead to export of conditions data
  - A temporary problem, will be solved by more powerful data distribution tools

Configuration Database Status

- Used in day-to-day online detector operation since May
  - Key selection is made through RunControl (run type = alias name)
  - Used to configure dataflow FX, Oep, L3, calibration, ...
- Currently 400 configurations registered (most defunct)
  - 4 RunControl run types, plus calibrations
  - under 50 MByte total size
- Some configuration data is still in flat file form (calibration)
- Editing + manipulation of configurations is cumbersome
  - GUI interface is slow, connections must be made by hand
  - Tools to automate common tasks are imminent
- Work to extend the config DB for use in reconstruction is underway
  - Can replace some flat files there

Datastream Calibration Databases

- Support calibration in Opr using physics data as stimulus
- Spatial and Temporal databases store Mergeable objects
  - Each object holds accumulated statistics (histograms, scalar sums,..)
  - Two mergeable objects can be combined (combines statistics)
  - Mergeable objects can be fit to derive calibration constants
- Spatial database
  - Merges data accumulated on different nodes for a single Constants Block
  - Uses Objectivity as a way to implement distributed objects
    - More flexible than CORBA
    - Allows reuse of existing calibration code
  - Reuses event store technology (Trees and Tree nodes)
- Temporal database
  - Stores results of spatial merge for each Constants Block (~1/2 hour chunk)
  - Proxy interface merges multiple adjacent CBs if needed
  - Temporal DB is derrived from the conditions DB
- Same databases also (will be) used for Oep BhaBha calibration

Datastream Databases Status

- Code is complete and has been (briefly) excercised
- Can be fully excercised with split federations
  - Spatial and Temporal databases are not overwritten by the sweep
  - Extraction of constants from temporal data can be done offline (outside Opr)
  - Not suitable for factory-mode operation
- Some refinements have not been deployed
  - Ability to filter out ‘bad’ Contants Blocks in the temporal merge
  - Ability to merge fixed time block (only ‘running average’ is currently supported)
- Needs more support and operational experience

Conclusions

- BaBar is using several non-event Objectivity databases
- These work adequately for current operations
- Planned improvements will allow them work for factory mode
- Fixes to fundamental Objectivity problems will help here too!