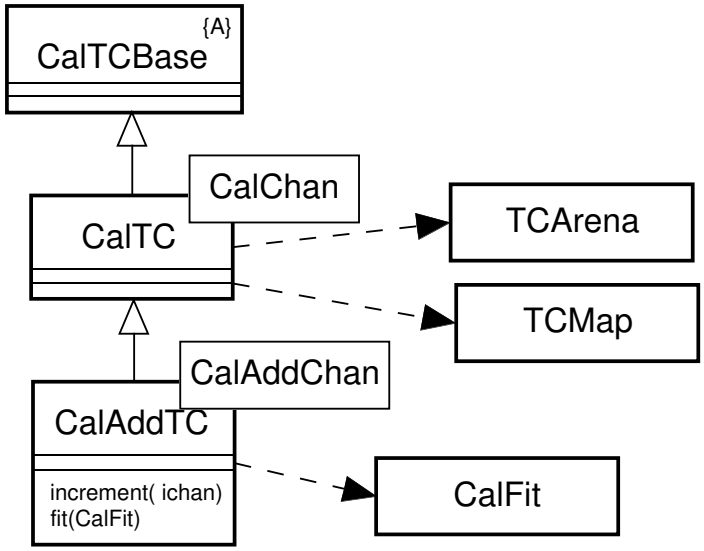
Calibration TC Status

- CalTC design was discussed during the working group meeting last week
  - ‘Proof of principle’ shown
  - Some design changes were proposed
- 2nd generation code to be released this week
  - BbrCalib and CalOnline packages

Calibration TC Requirements

- Calibration accumulation functionality
  - Construct a fully populated CalTC with ‘empty’ CalChans (using a Map)
    - In program memory for transient CalTCs
    - In Datagram for transport
  - Efficient access to single channel increment method from sparse input (indexed access)
  - Post-processing after accumulation (fitting)
- OdfTC functionality
  - Iterator isn’t generally useful
- Efficient dataflow transport
  - CalChan storage contiguous in memory with direct data members (Arena functions)
- Reconstitution after transport to a different machine architecture
  - Floating-point representation (OK for our platforms)
  - byte ordering (general dataflow problem)
  - CalChan vtable (pointer) $\rightarrow$ copy reconstitution
- Navigation of Tagged-container heirarchy
  - Distinguish identical classes with different contents (secondary ‘contains’ key)

CalTC Classes



Notes

- CalTC owns array of CalChans placed in TCArena with overloaded global new()
- CalTCBase stores position of CalChan array in char* (machine independent) units
- CalTCBase stores memory layout of CalChan
- Reconstitution uses only CalTCBase functions

Summary

- Preliminary implementation of CalTC in next release of CalOnline package
  - Statisfies requirements
  - Build requires Odf package, ODF_BASE variable
  - Placeholder versions of Map and Arena classes
  - Simple test program
- To Do:
  - Accumulation from input containers
  - Test chain from accumulation to storage
  - Byte-order problem