Sat

The CFS is written in ANSI C and the current flight applications are also written in C. However there are benefits to an object –oriented design approach so this project defines an object-based application design pattern with a template application source code and a unit test framework. The template source is complete so it can be compiled and run. The template is designed for automated variable substitution and the resulting files can also be compiled and run with modification which allows a developer to always start with a working application.

Object-based Design Approach

Applications are built upon a common set of cFE services therefore how applications manage their interface to these services can also be normalized or standardized. The common application services/interfaces include

1. Command dispatching
2. Event message generation and filter management
3. Table management
4. Telemetry Generation
5. Child task management

Not all of the services/interfaces are complex enough to warrant a helper class so most of the services are used directly.

TODO – Application design picture

TODO – Go though each service and describe approach

Application Design Steps

1. Don’t jump into coding, do an explicit design phase so you get the object breakdown correct. Design the objects and determine the application layering. Ask yourself questions to determine what level things may make the most sense like command functions. At what level of the object hierarchy should a command function exists? This usually depends on the functional impact of the command and what data (i.e. knowledge) is required.
2. Provide reset status function, command functions, aggregate data structure
3. Coding: Keep the end in mind from the start. Use a template processing to create the application shell. As you code each object think about the scope of macros definitions.
4. Be consistent. Keep structure names hierarchal so they read well and the functions/data are well matched. This reduces coupling and makes readability and maintenance easier. Keep functions alphabetical.
5. Balance global/local thinking. Too easy to get caught up in overthinking reusability. Don’t try to do everything at once from a systems perspective. If see common objects then may or not.
6. The template uses a Singleton Design Pattern by having an object store a pointer to its instantiation. The pointer is passed to the object during construction. An alternative is to have a reference to an object passed to each member function. Two factors drove the singleton design: (1) If an object has a command function then it is registered with the command manager’s dispatch function. When the command function is called it would need to know which object instance to operate upon. (2) Applications typically have one instantiation of an object, at least for the objects that interface with the application main and register command functions. A different design pattern and template could be added that supports the multiple object paradigm.

Deploy/Install an Application

1. Create app source folder aaa in opensat/apps/aaa
2. Create app folder aaa in build/ppp/aaa
3. Copy aaa/fsw source files into opensat/apps/aaa
4. Copy aaa/deploy makefile to build/ppp/aaa
5. Copy aaa/deploy/aaa\_mission\_cfg.h build/mission\_inc
6. Copy aaa/deploy/aaa\_platform\_cfg.h to build/aaa/inc
7. Add to make file in build platform
8. Add to startup script
9. If needed add HK message to schedule table

Update CARE ground system

1. Add cmd/tlm message IDs to /config/care-msg-ids.xml
2. Create DB XML file and place in /db-apps folder. Same msg id names defined in care-msg-ids.xml must be used

Genera Notes

1. Same app\_config.h header when multiple apps can be confusing
2. Deviations from cFE
   1. Text based tables
   2. Directory hierarchy is different. Explain why. One deploy folder used to contain all
3. UT philosophy
   1. Don’t use flight tables. Keep them in deploy folder
4. GNC application framework is left for historical reasons. Its goals were commendable but probably better if C++ and libraries and tools available.