F Prime

A Flight-Proven, Multi-Platform, Open Source Software Product Line for Embedded Systems

https://nasa.github.io/fprime

F Prime is tailored to small-scale system deployments including CubeSats, SmallSats and instruments

F Prime includes:

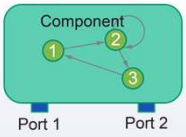
- **An architecture that decomposes flight software into discrete components with well-defined interfaces**
- **A C++ framework that provides core capabilities such as message queues and threads**
- **Modeling tools for specifying components and connections and automatically generating code**
- **A growing collection of ready-to-use components**
- **Verification tools for testing flight software at the unit and integration levels**

Component-based architecture enables a high degree of modularity and software reuse

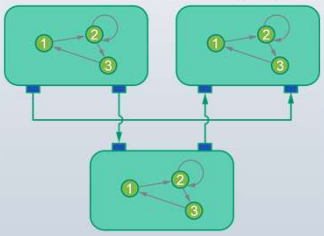
Software is composed of **components** (behaviors) and **ports** (interfaces between components)

Components are interconnect together to form **topologies**, which comprise the binary built as a **deployment**.

Components have no compile-time or link-time dependencies, so they can be easily recombined to form alternate topologies.



Connected components in a topology

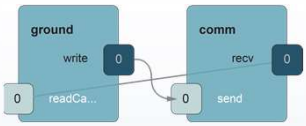


Architectural Features

Rapid Deployment	Component Reusability
Portability	Testability
High Performance	Software System Analyzability

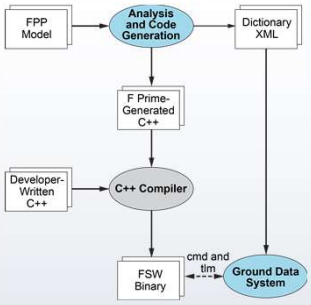
Supporting Key Phases of Software Development

Design



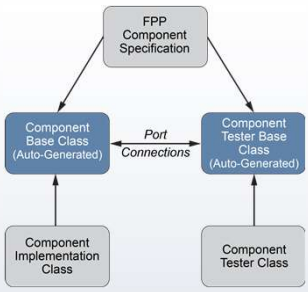
- **A domain-specific modeling language, F Prime Prime (FPP) provides modeling front end for specifying components, ports and topologies**
- **FPP can also specify component commands, telemetry, events and parameters for a ground dictionary**
- **Companion tools provided for visualizing topology graphs**
- **Analysis tools check correctness of models prior to code generation**

Implementation



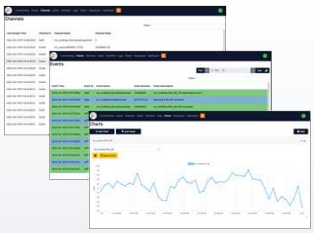
- **Autocode tools translate FPP models in to C++, representing the bulk of the structural code for the software system**
- **Users develop the behavioral code for the components**

Unit Testing



- **Auto-generated test classes provide mirrored interfaces to modeled components for developing unit tests**
- **Rules-based testing option further streamlines test construction and improves reusability**

Integrated Testing



- **Integrated Python-based Ground Data System included supporting integration and system level testing**
  - **Incorporates auto-generated command and telemetry dictionary**
  - **Integration and test API provided for writing automated tests**
  - **Web-based GUI for sending commands, viewing telemetry and events and simple plots**
- **F Prime can be adapted to support other GDSs employed on a project**

National Aeronautics and Space Administration
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

www.nasa.gov

For More Information: fprime@jpl.nasa.gov