

# JTRS Visual Modeling Studio

Garrett Wampole - Ben Litchfield - Jason Gilman - Jason Offord - David Bryant  
Sponsor - Charles Linn, Harris RF Communications  
Faculty Advisors - Prof. Jim Vallino & Prof. Stephanie Ludi



R·I·T

Software Engineering

## Current Software-Based Radios

- Monolithic code packages
- Hardware and software allocation determined in advance
- Closed, non-extensible interfaces
- Static, non-distributed architecture

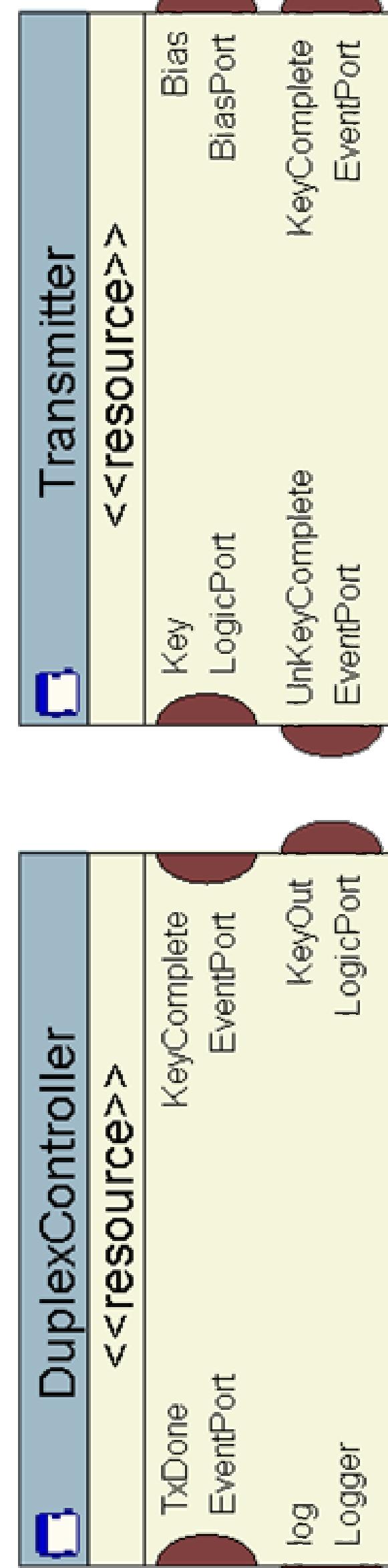
## JTRS Software-Defined Radios

- Separation of hardware and software concerns
  - Application packages can be loaded and installed at runtime
  - Applications have component-based, modular design
  - Standardized interfaces allow software components to examine the operating environment dynamically
- Domain Profiles and the SCA**
- Radio hardware capabilities, application components, and their interconnections and dependencies are described by the Domain Profile
  - Domain Profile is composed of a set of XML files whose format is specified by the Software Communications Architecture (SCA)
  - Domain Profiles can be large, a complex application can take thousands of lines of XML to describe

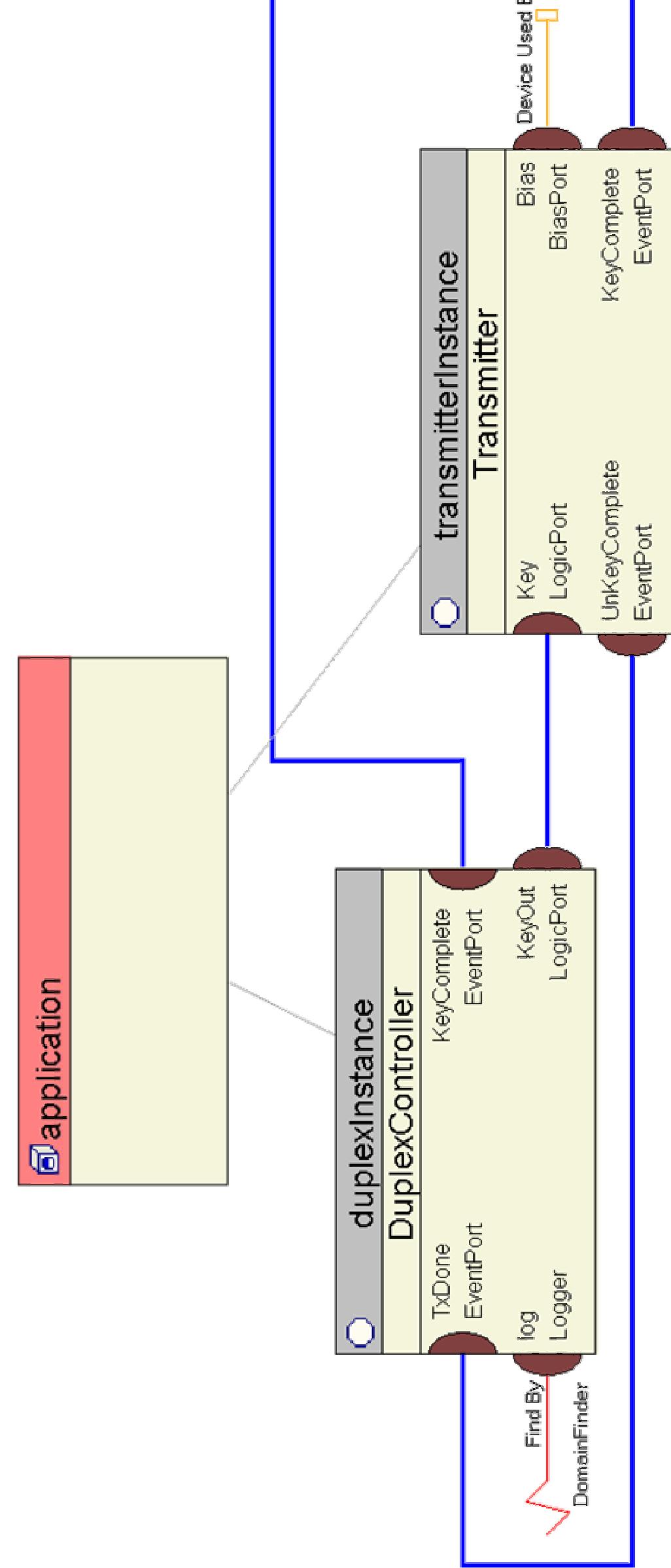
## JVMS

- JVMS allows designers to graphically model SCA Domain Profiles and generate the corresponding XML
- Projects can be saved to an interchangeable format at any point and returned to later
- Relationships between components are represented graphically and their attributes can be modified visually
- Models can be validated against certain criteria, designers are guided to where errors exist
- Flexible architecture for adding new validation rules, the set of rules which are run against projects can be configured at runtime
- Extensive online help provides designers with descriptions and hints about SCA attributes
- Models can be exported to various image formats

## Component View



## Assembly View



## Development Process and Technologies

- JVMS takes advantage of Microsoft's .NET platform
- .NET provides consistency across Windows® platforms and a rich set of libraries
- Interop assemblies maintain platform abstraction while providing access to the underlying operating system
- Assign each team member clearly defined roles and responsibilities
- Development cycle split into three phases, each with clearly defined goals and target features
- Conduct a product release after each phase, validate functionality with customer and factor input into next phase
- Create set of process support tools before start of project
- Web-based facilitates communication between team members, customer, and faculty advisors
- Available at: <http://jtrs.kelut.org/>