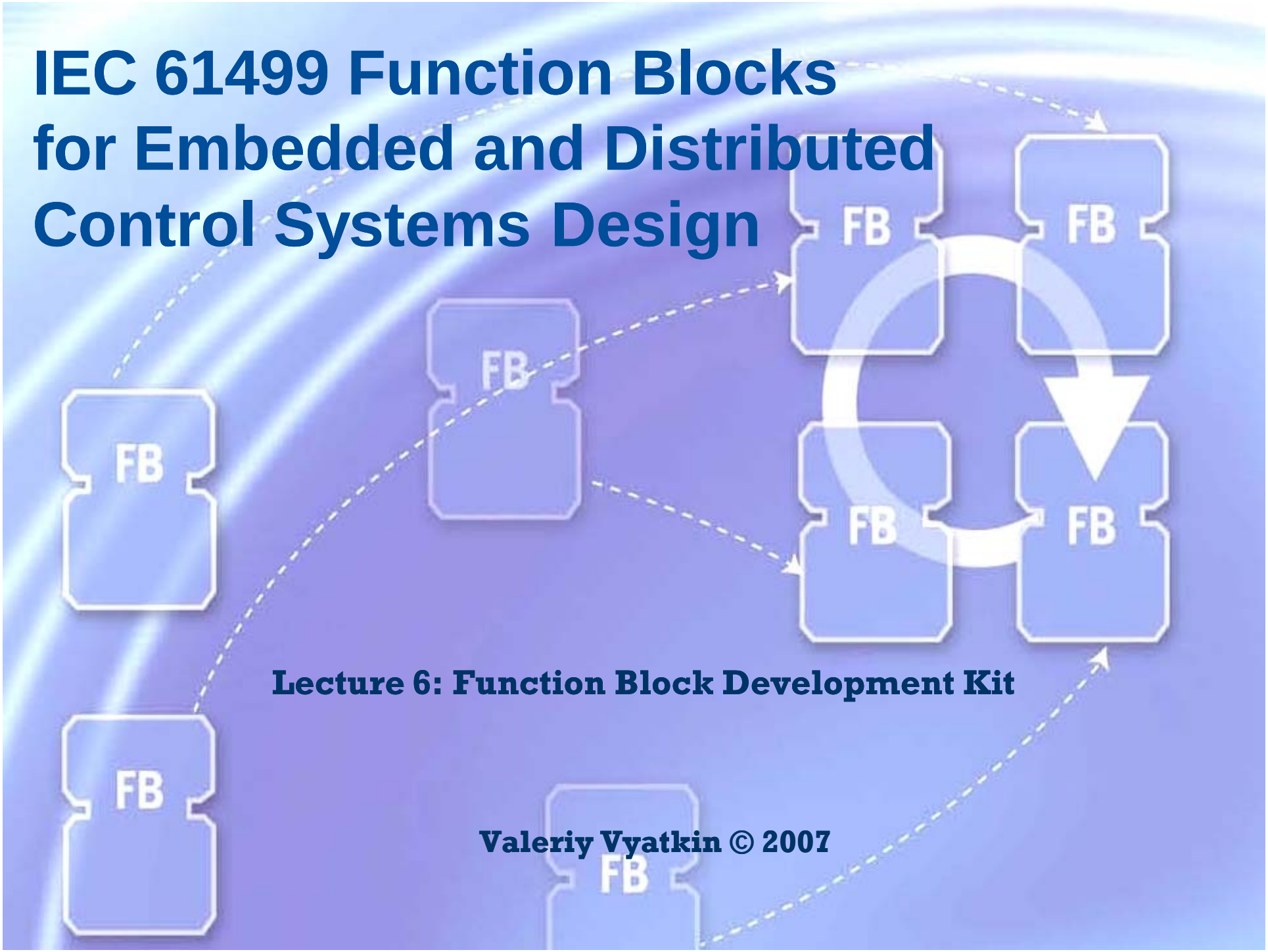
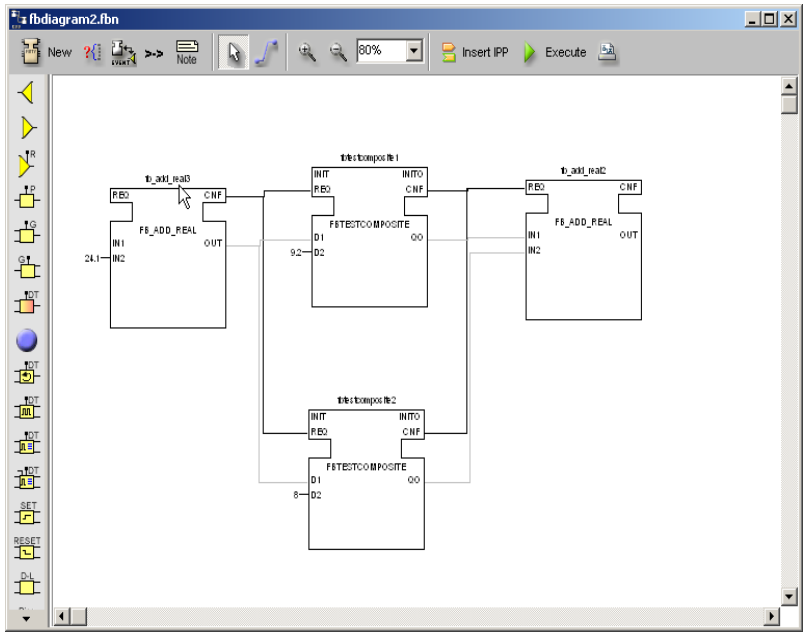
IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design



Lecture 6: Function Block Development Kit

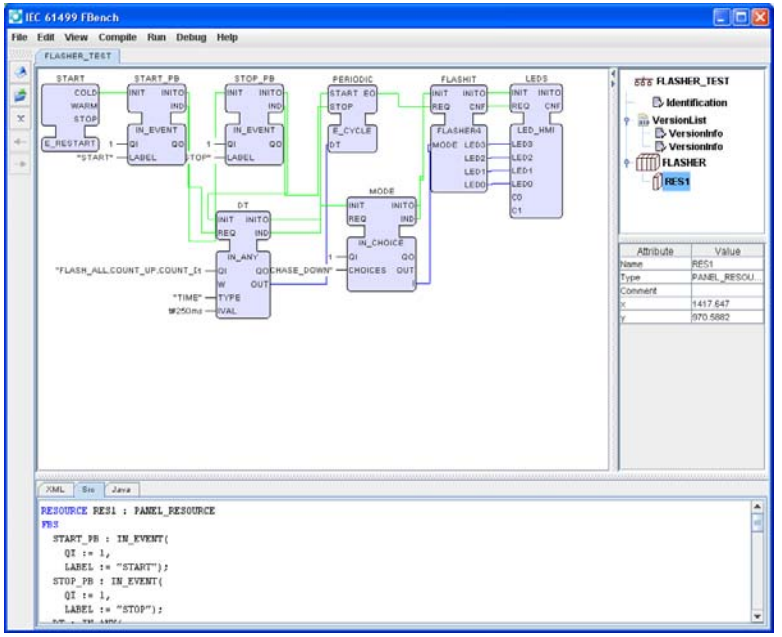
Valeriy Vyatkin © 2007

Available Function Block Tools



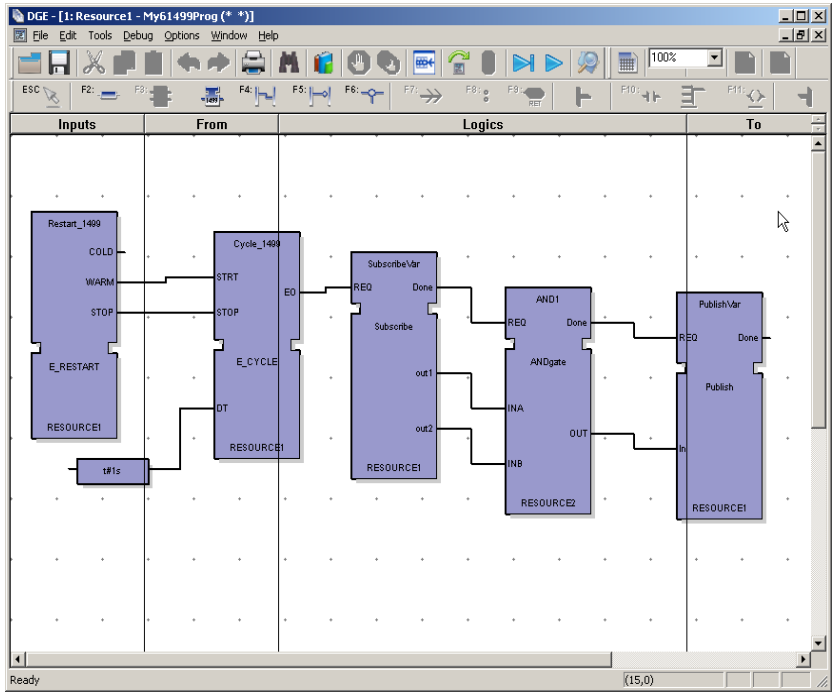
CORFU –University of Patras, Greece

Provides a link between UML and IEC 61499 Function blocks

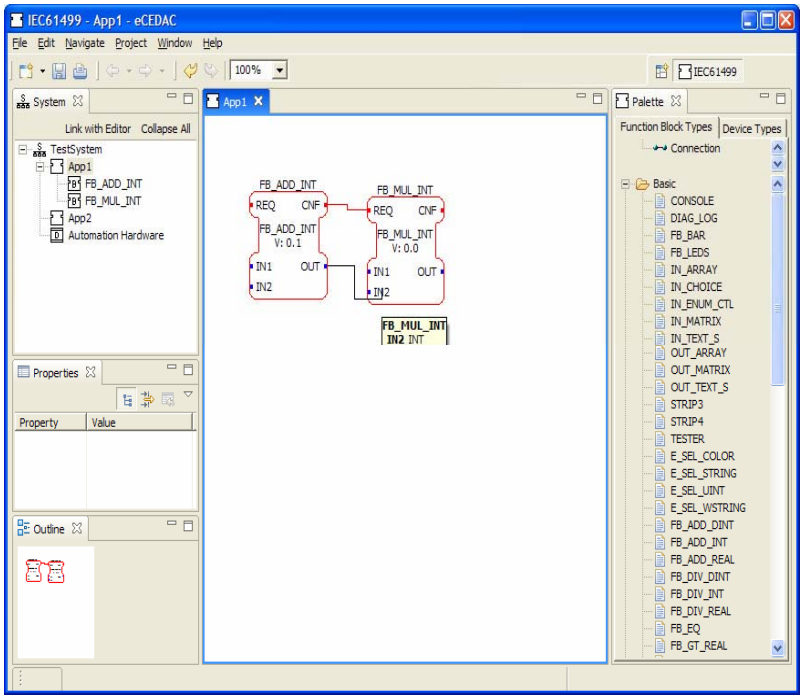


FBench – Open Source development of the University of Auckland, New Zealand, based on OOONEIDA Workbench

Available Function Block Tools

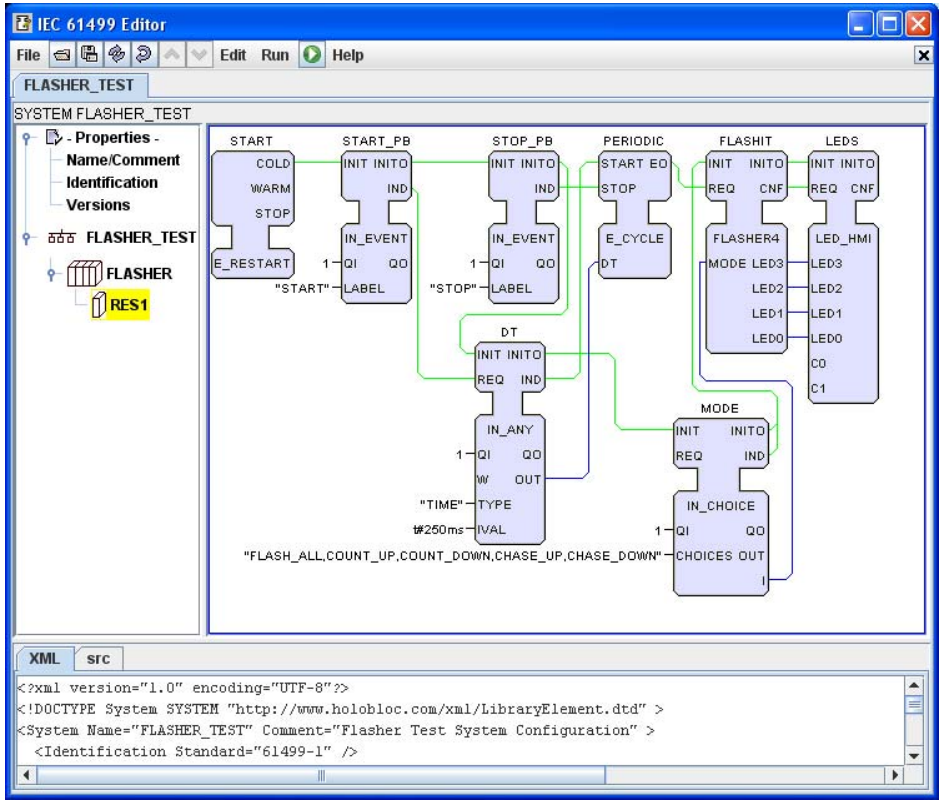


ISaGRAF – first commercial tool, ICS Triplex, Canada



ODECE –tool implemented as an Eclipse plug-in, developed at the Technical University of Vienna, Austria

Function Block Development Kit (FBDK)



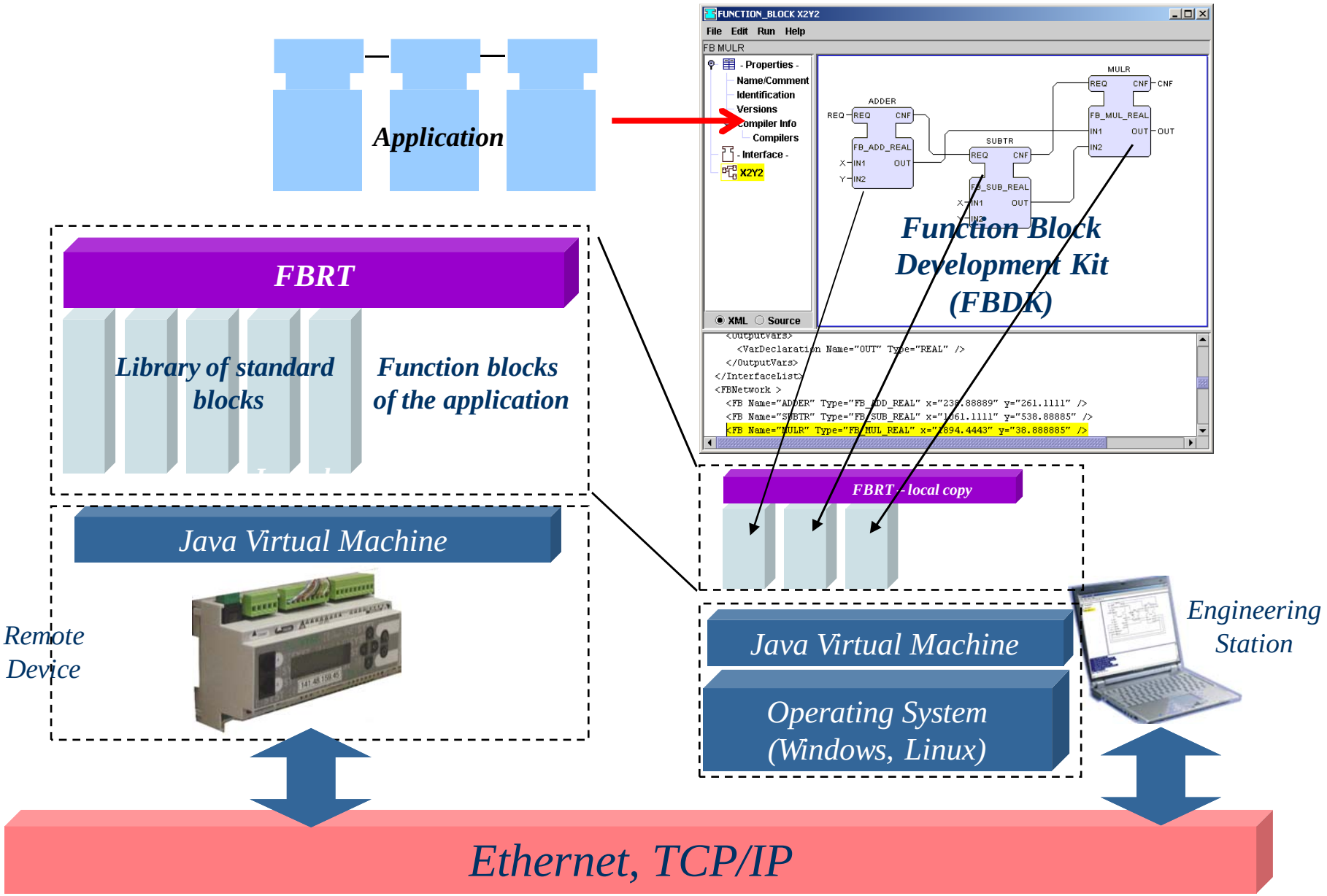
Most popular research tool

Developed by: Rockwell Automation/ Holobloc Inc., USA

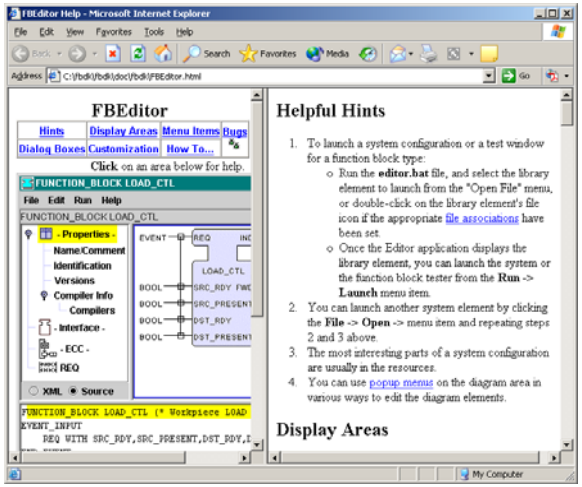
Free downloadable from www.holobloc.com

Dr. James Christensen – FBDK Creator and leader of IEC 61499 Standardisation Working Group

FBDK: Java-based implementation of IEC61499

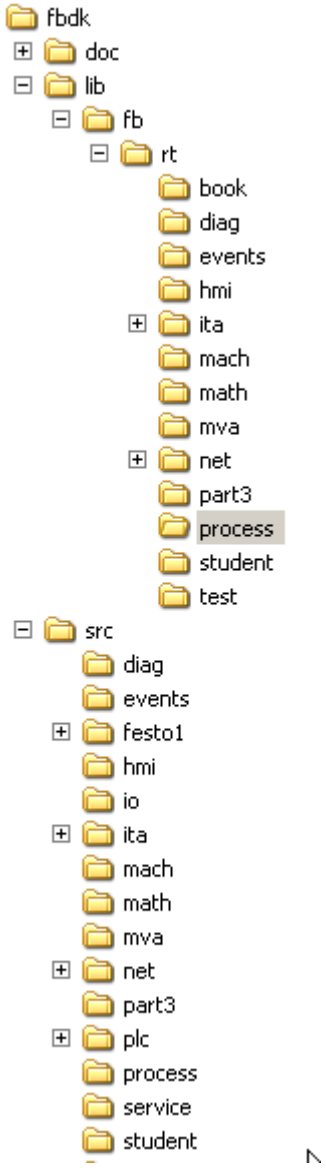


Data Structure



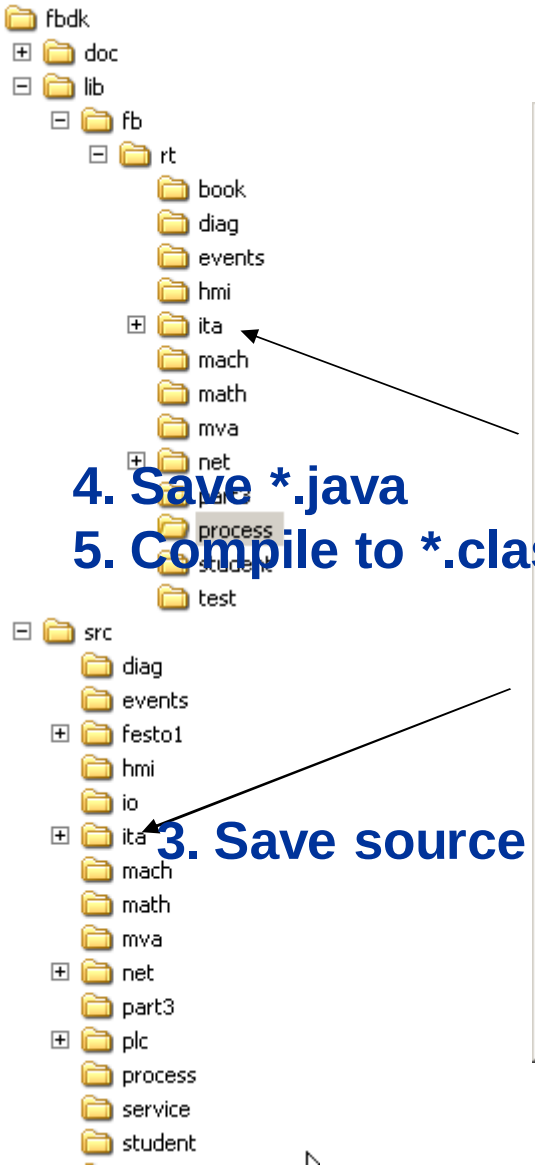
- Documentation in HTML
- Library of JAVA sources and compiled CLASS files
- Library of sources:
  - Function block types (*.fbt)
  - Adapter interface types (*.adp)
  - Resource types (*.res)
  - Device types (*.dev)
  - System configurations (*.sys)
  - ...

Folders



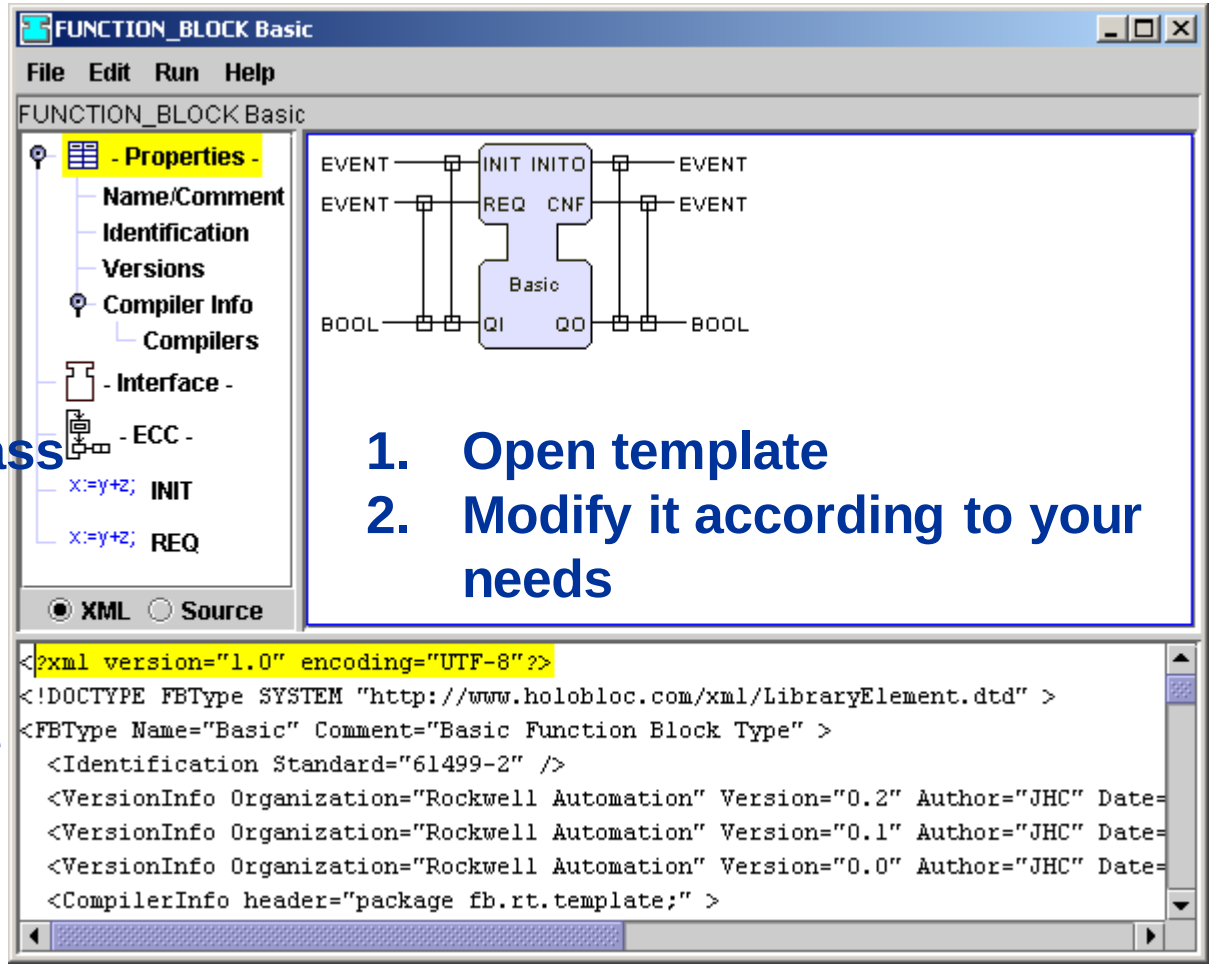
- Nested according to the Java package hierarchy
- lib: fb.rt. + package name
- src: package name

Create a basic FB

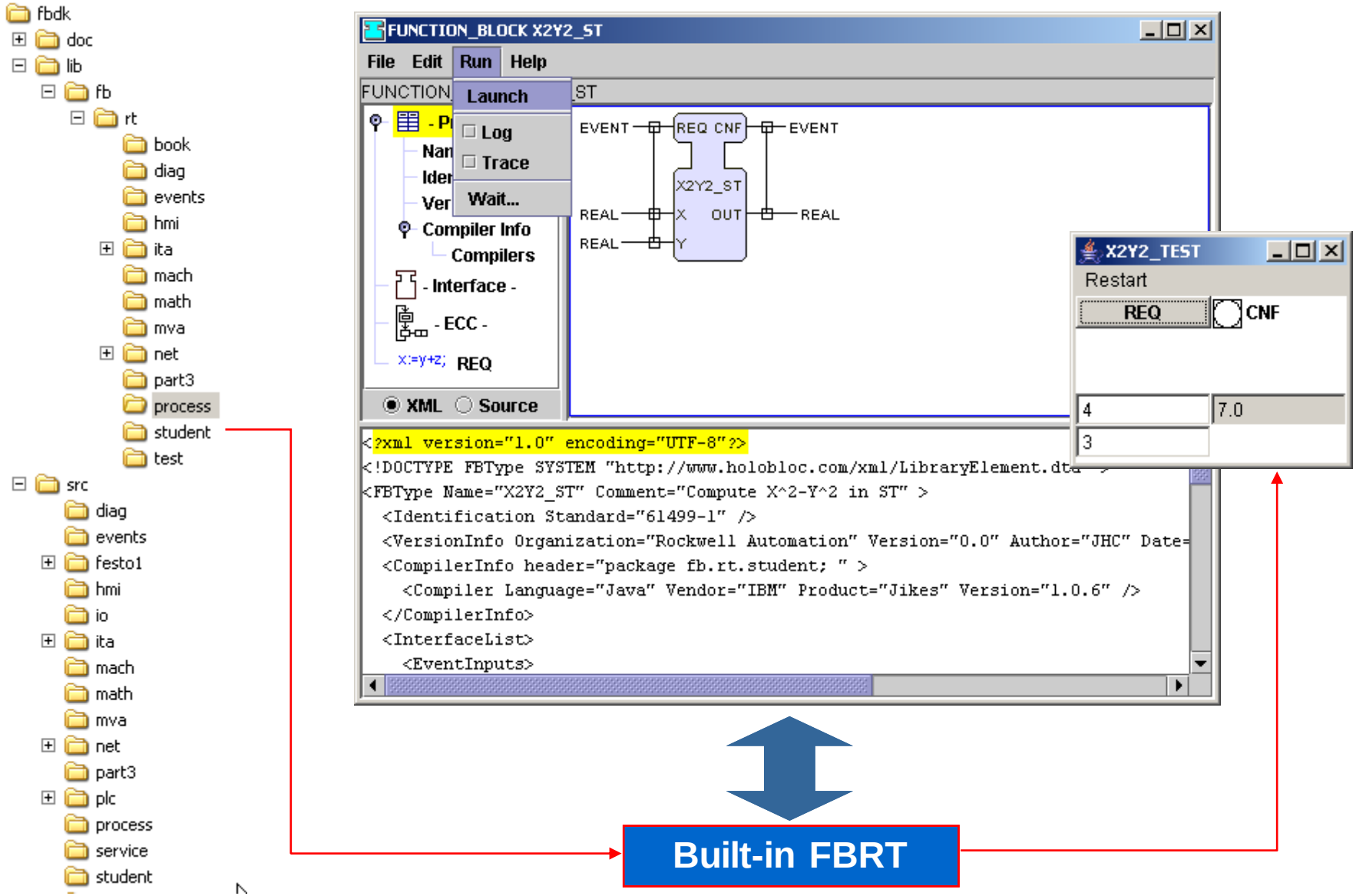


4. Save *.java
5. Compile to *.class

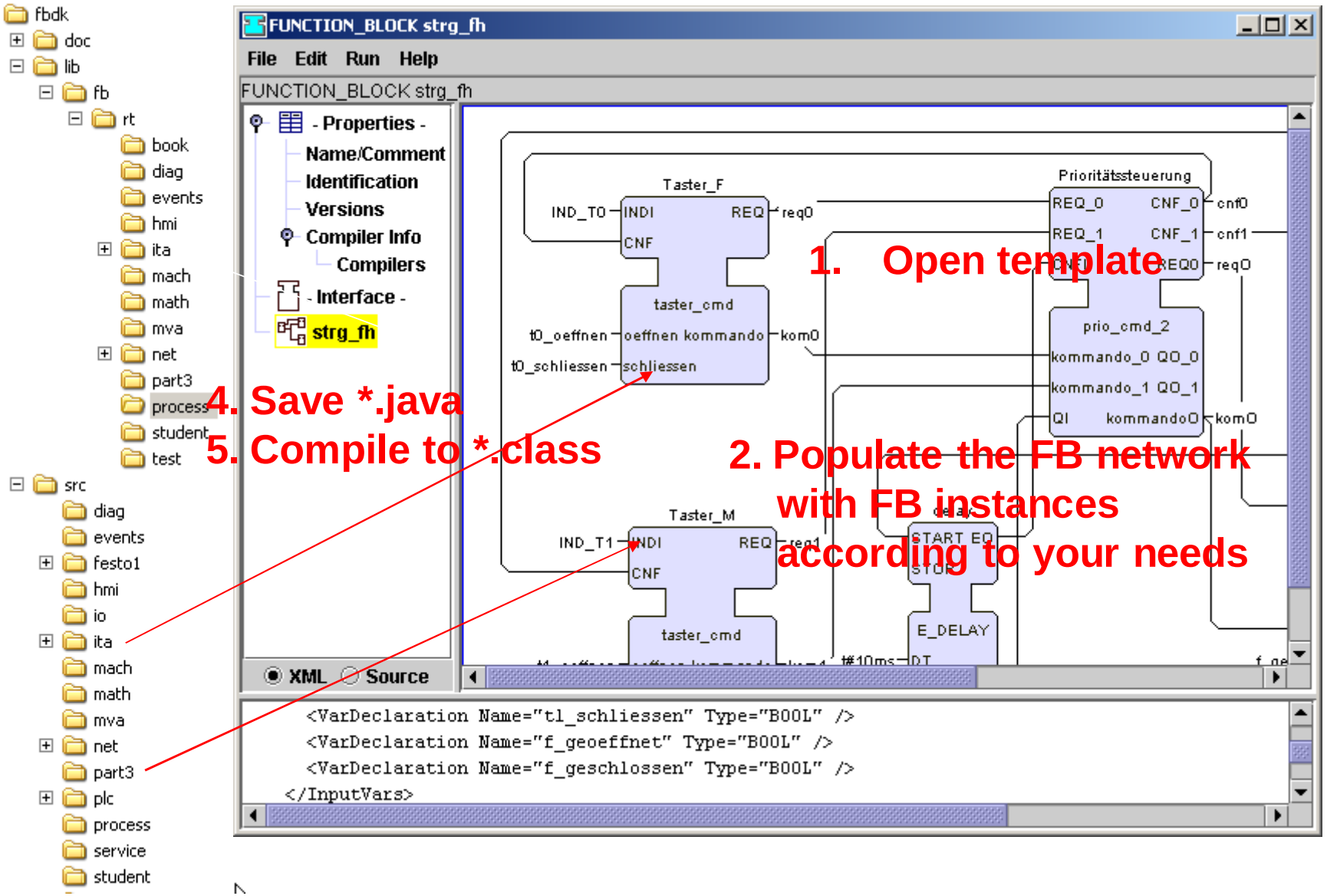
3. Save source



Run Function Block

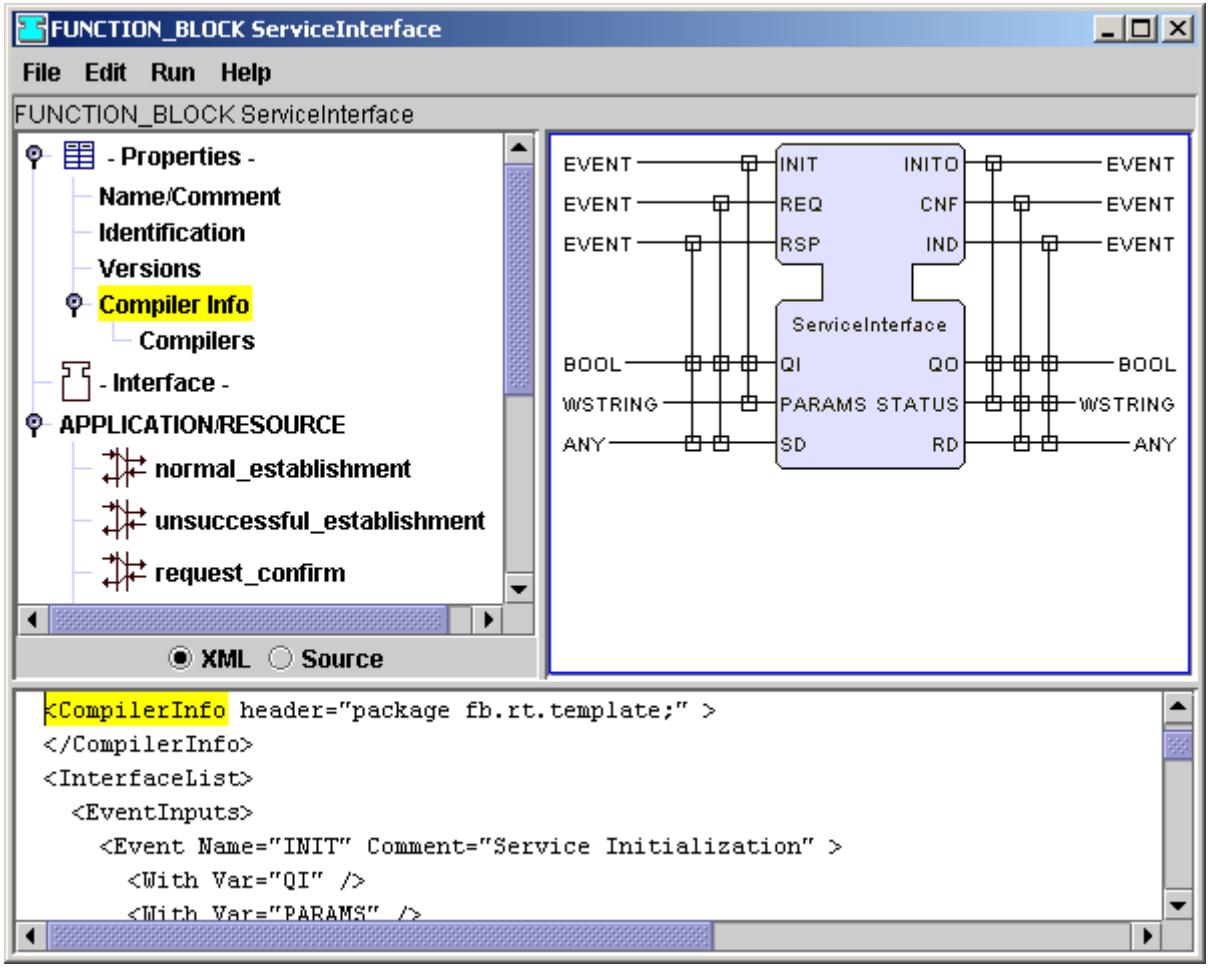


Create a Composite FB (or Application)

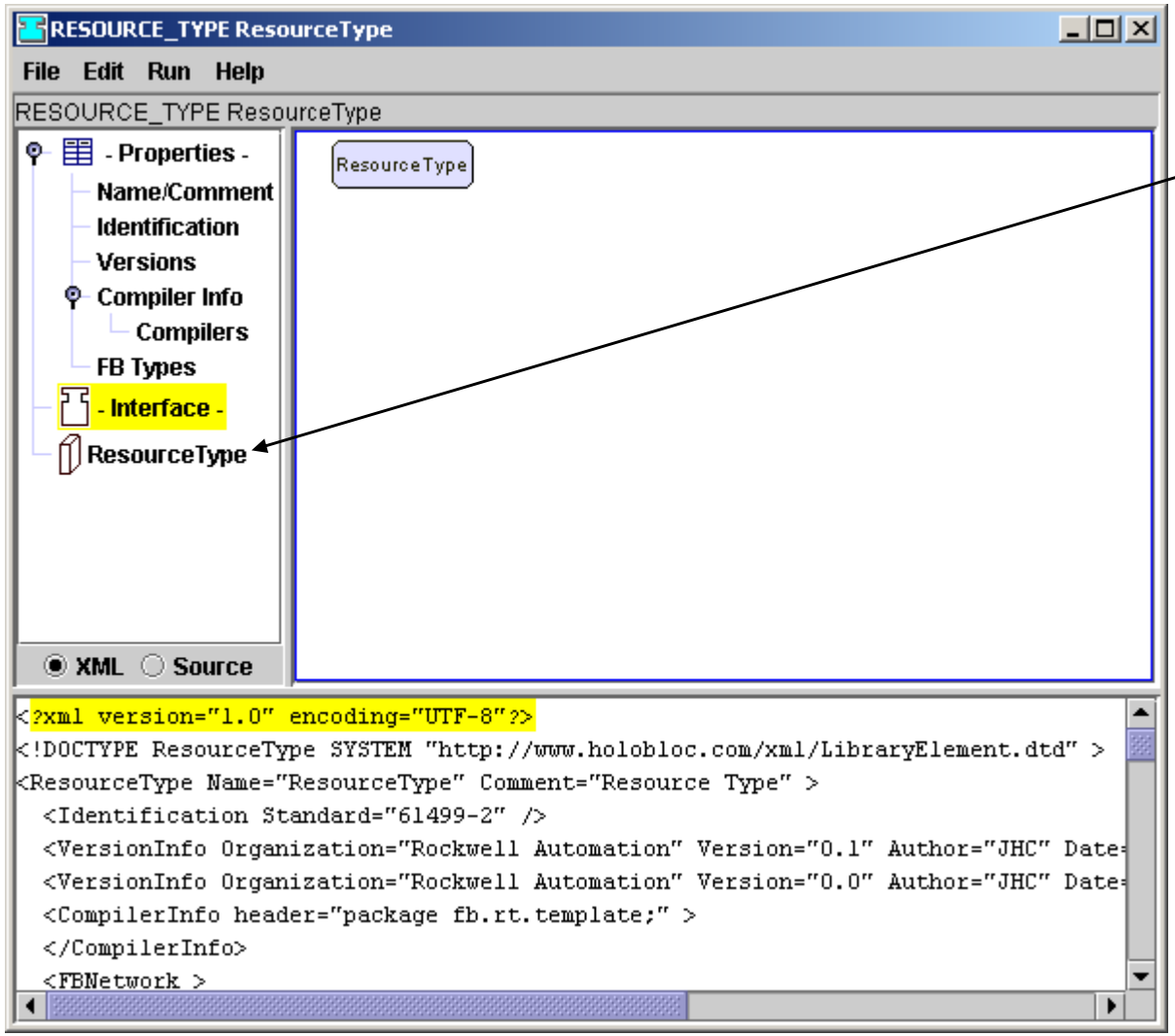


Create a SIFB

1. Start with the template
2. Modify the interface
3. Modify the list of implemented services
4. Save as *.fbt
5. Save as *.JAVA
6. Modify manually JAVA code adding new methods implementing the services
7. Compile *.JAVA into *.Class



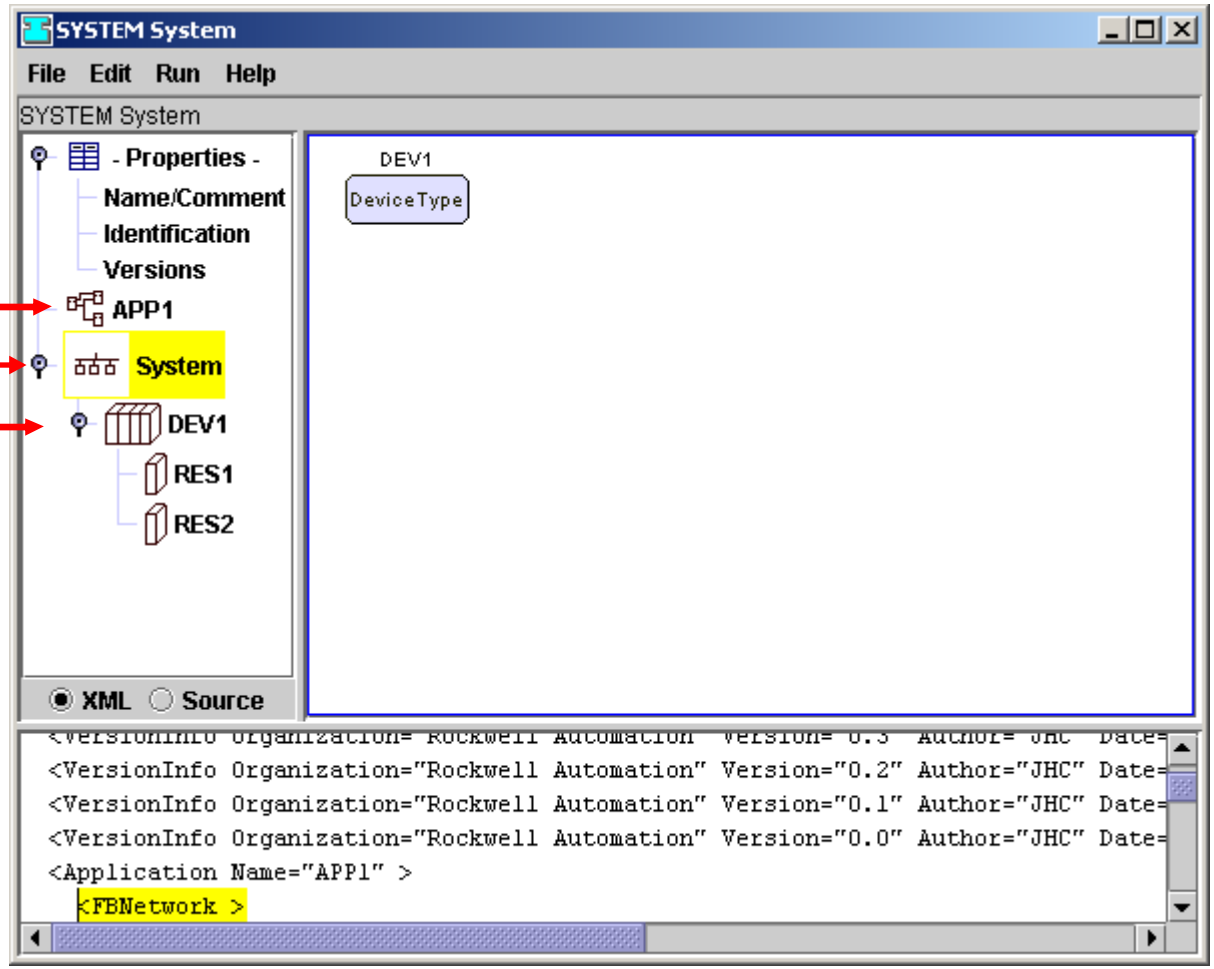
Create a resource type



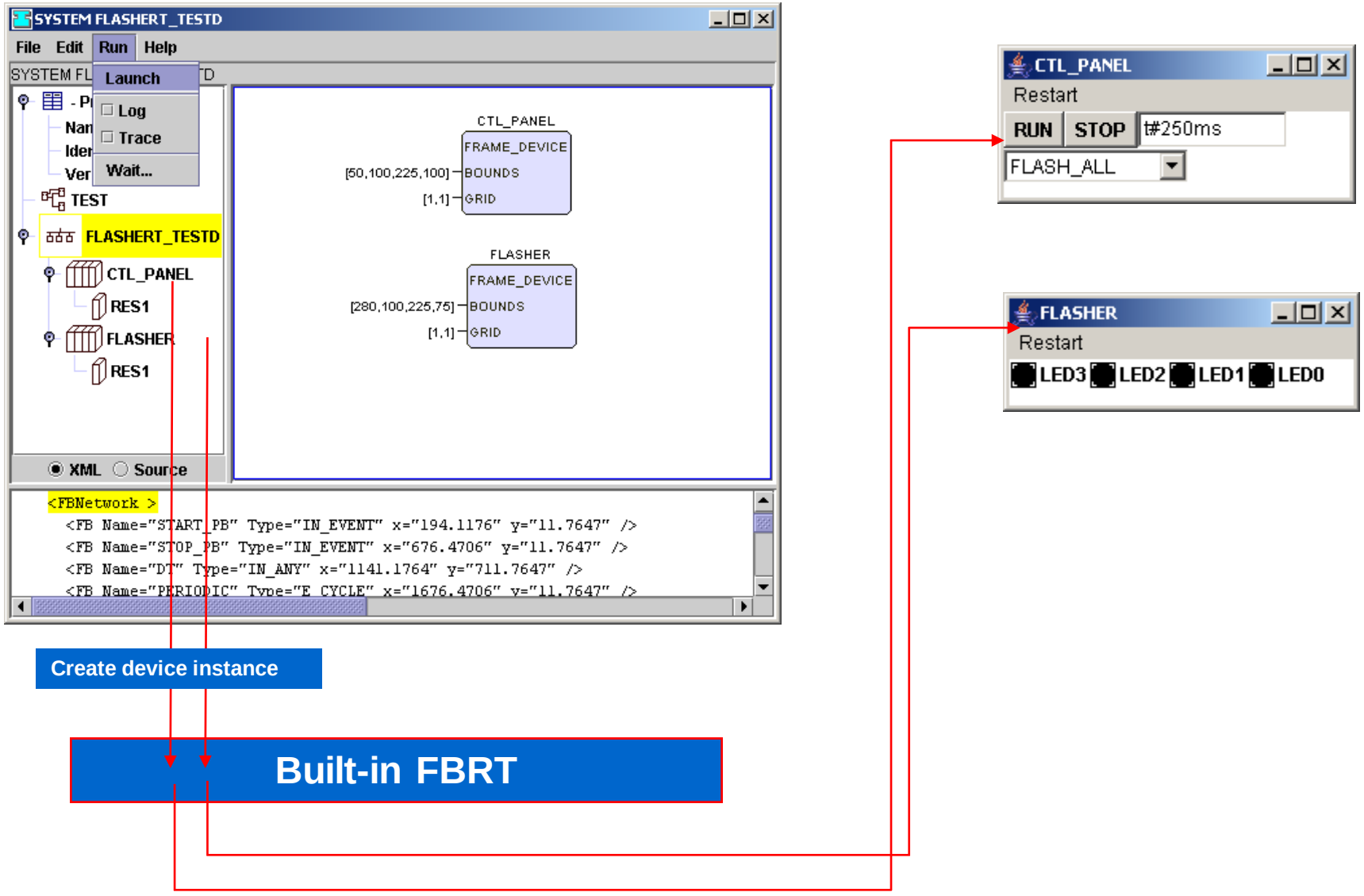
1. Start with resource template
2. Populate with function block instances needed in this resource by default (other function block instances may be added to the resource instance)
3. Save as *.res
4. Save as *.java
5. Compile *.java into *.class

Create a System Configuration

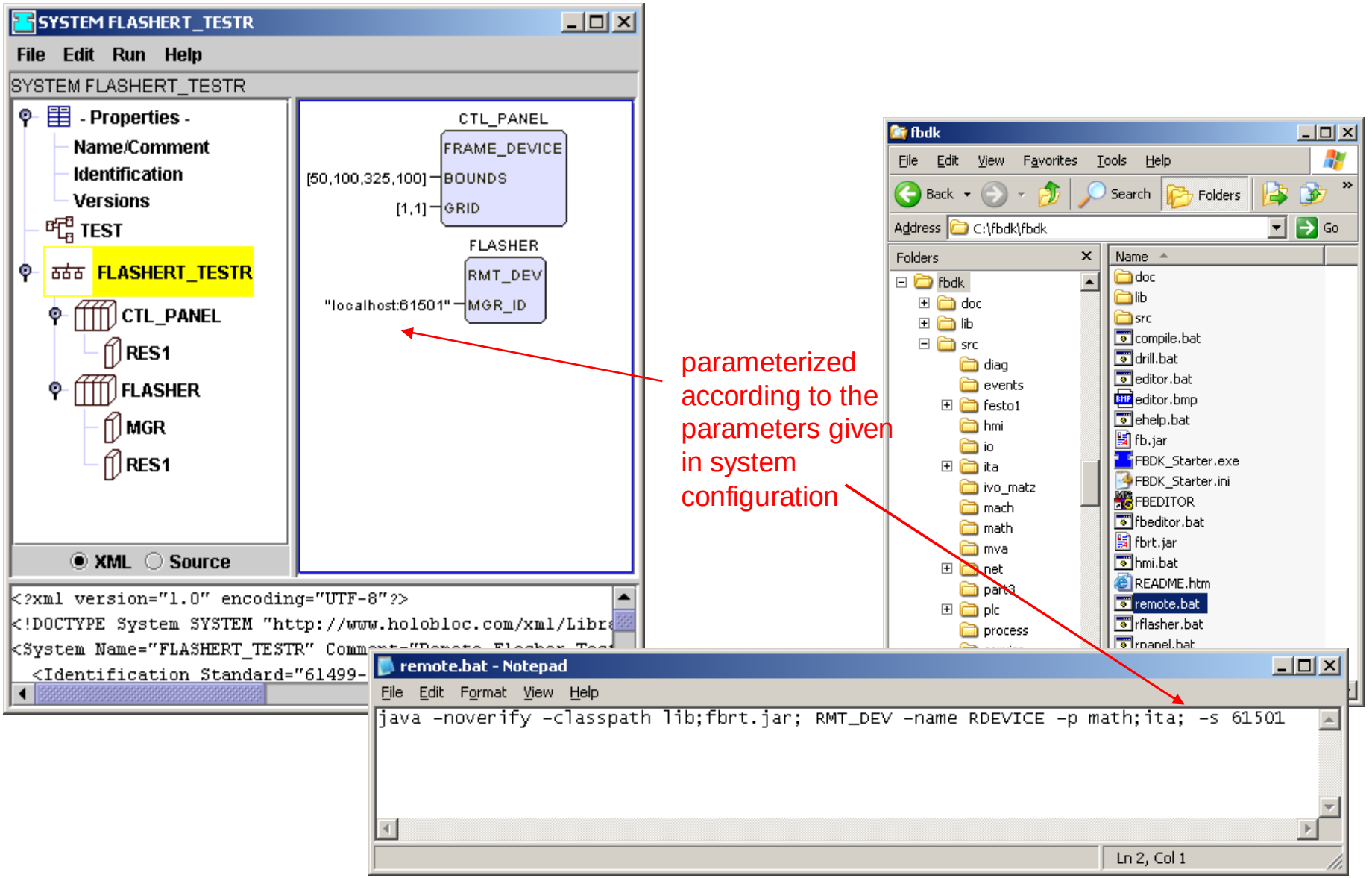
1. Start with the system template
2. Add/create the application
3. Modify/Create the structure of devices by instances of device types
4. Map function blocks from the application to devices
5. Parameterize device instances
6. Save as *.sys
7. RUN!



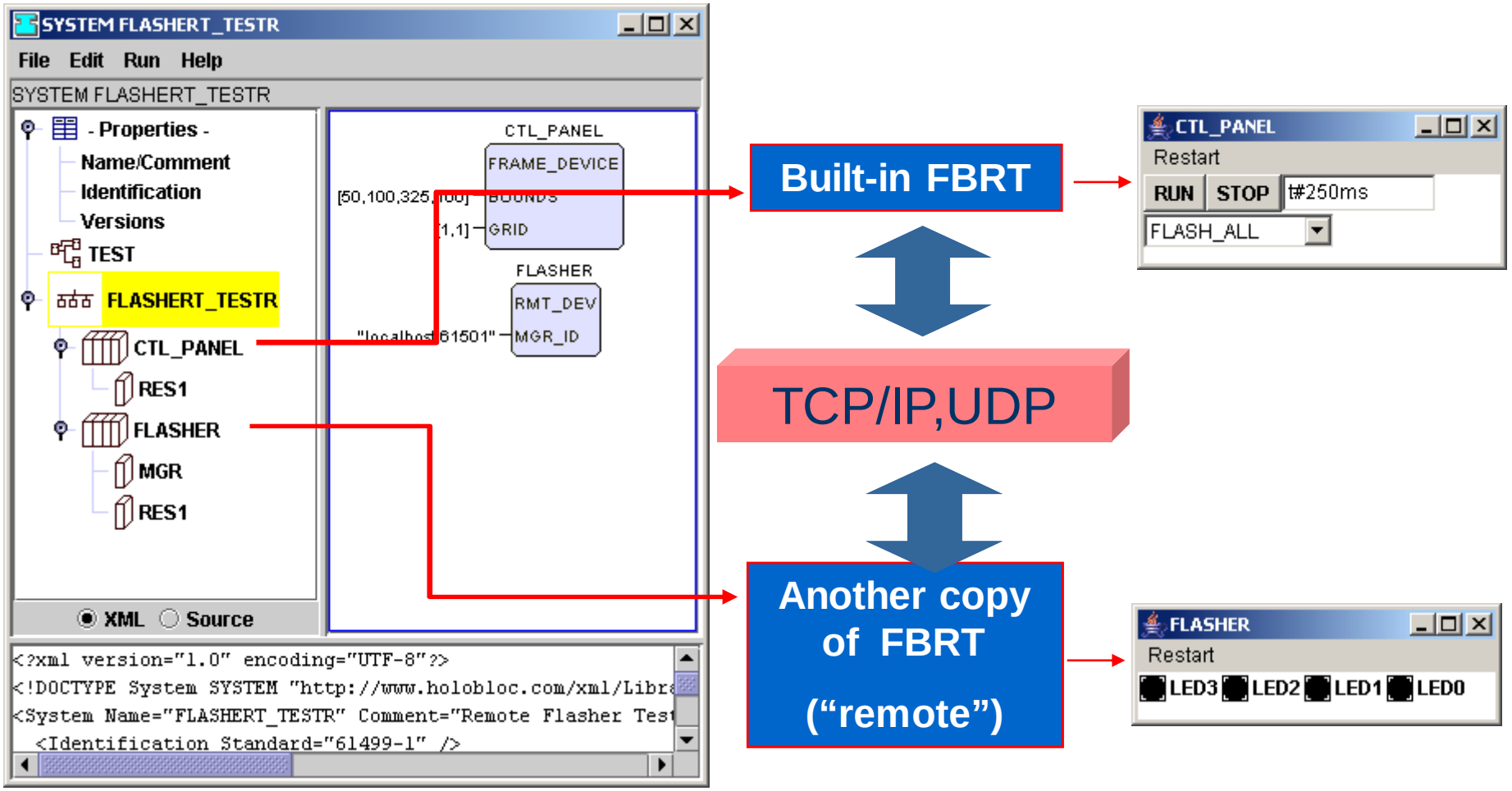
Run System Configuration



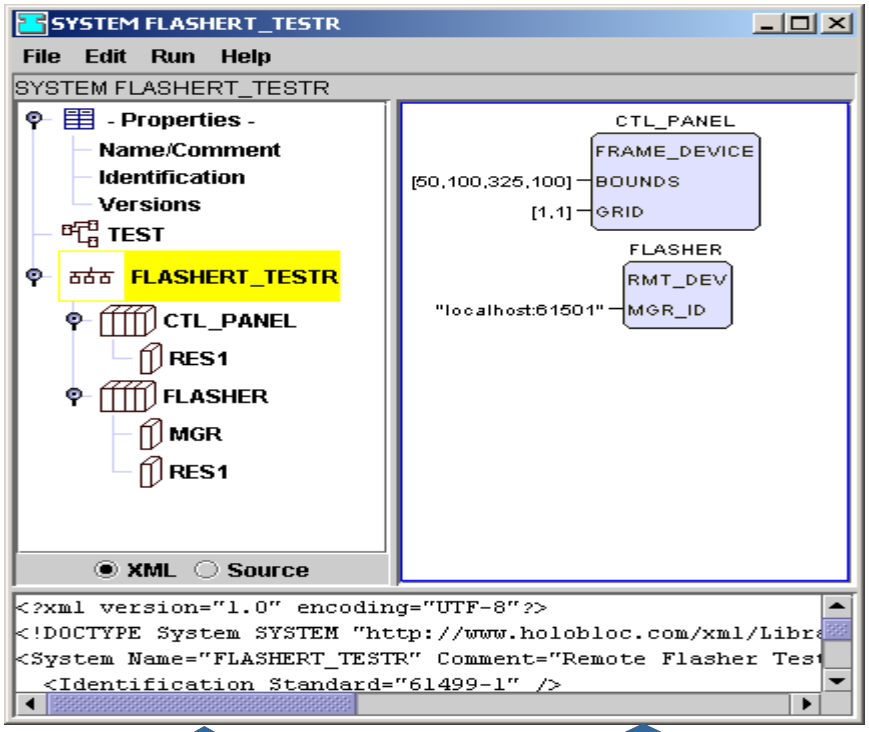
Run a Distributed System Configuration

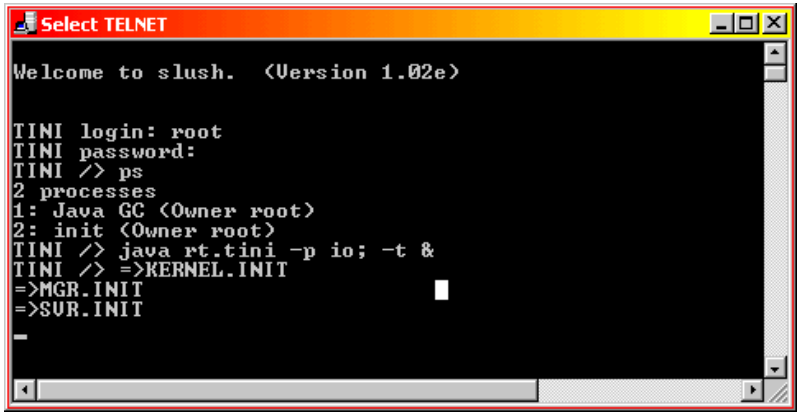


Run a Distributed System Configuration (2)



Run a remote device



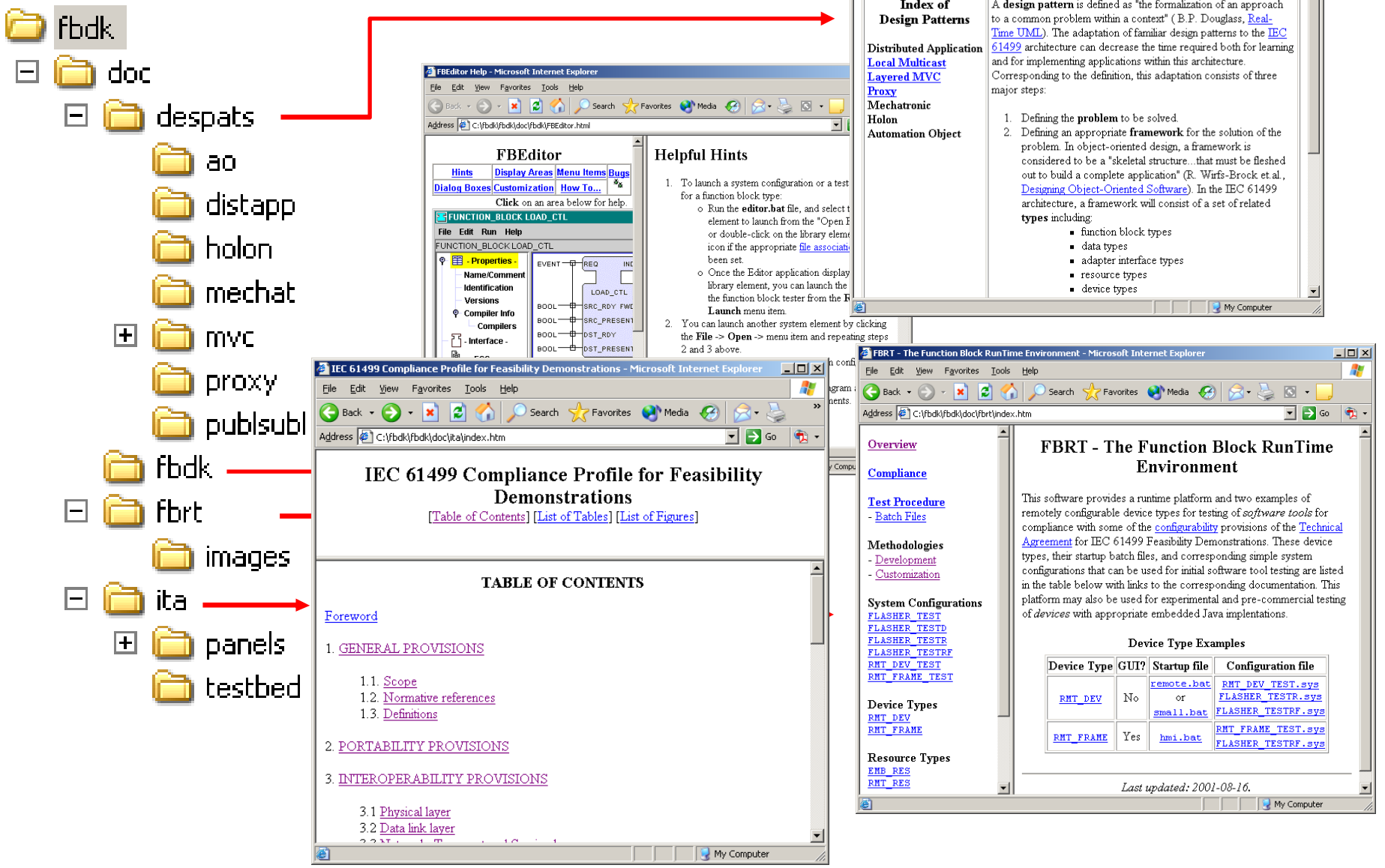


Remote FBRT

Built-in FBRT

Ethernet

Documentation



What is missing in FBDK?

- **Build the system configuration (recompile necessary function blocks)**
- **Dynamic trace of data/algorithms**
- **Dynamic upload of classes into running remote devices (requires also support from the device side)**