Use of Tcl/Tk in Railway signalling simulation and maintenance software

Eric Boudaillier
Adrien Peulvast
Sommaire

• Context

• User interface tools

• Tests tools

• Examples on current projects

• demos
Context

- Signalling design has to adapt to different clients
  - New markets with their own practice
  - New system functions

- Internal debugging is necessary:
  - Generic kernel (System functions)
  - Instanciation rules
  - Topology

- System level view of the signalling equipments
  - Identical as the signaller view
  - High level (commands, controls, graphical animation)

- Simplicity to be used by projects team (not software engineers)
- Signalling system definition cycle -

CT = control tables
CP = Production Tool – (chaine de production)

Goals:
- Process guidelines
- Automatisation of topology data.
- System functions verified and agreed.
- Instanciation rules tested.
- Good quality parameters.
- GUI mutualisation: views are integrated in the TT.
Graphical objects editor

• Scanvas: layer over the Tk canvas with tkpath

• Graphical symbols are described in text file. A symbol is:
  • A group of atomic gadgets (polyline, oval, …)
  • Parameterized by attributes
  • Animated through an set of variables

• Graphical symbols are generated in Tcl code.
Graphical objects editor
Topology data extraction

Control Tables generation

Raw Parameters generation
Topology data extraction

- Tcl scripts loaded as plugin of the editor
- Use the Scanvas API and the magnets to build topology
- Use of Tcom for Excel management
WhiteBox Simulator

- Import of production tool data

- Loading of simulation models
  - Automatic links between models
  - Defined or automatic links between models and graphical object
  - Models management with enhanced view (inputs/outputs, filters, graphs, model browser)

- Simulation management
  - Inputs setting / outputs getting.
  - Initialisation phase, environment simulation.
  - Running management (play, step-by-step, pause, fast forward, reset)
Simulation GUI – Graphical view

- Interactivity,
- Overview,
- Graphical objects finding tool
Simulation GUI – Data view

Use TkTable: thousands of variables in the table without any response time issue

- Filters and data selection
- Custom tables
- Change simulation inputs
Simulation GUI – Strip chart

• Custom set of variables,
• Enhanced controls: zoom, value display, etc.
Tests and validation use

• Real time simulation mode: the user controls the running and actions.

• Record/replay mode: scenarii can be recorded and play again.

• Validation Scenarii mode: high level scenario (include macros) run with the help of the simulator.
Maintenance Use: TT GUI

- Track views reused for the maintenance application
- First use TclOO
Other application: BlueBox Simulator

• A field simulator
  – Modbus standard communication with PLC (Triangle Research / Beckhoff)
  – Configuration management (inputs/outputs affectation + Ladder code generation)
  – Use graphical views to map objects in the PLC.
Other application: BlueBox Simulator
Other application: BlueBox Simulator

- Use of TkVideo
Tcl/Tk in the industry: advantages

- Modular development: from a simple software up to a complex one.
- Maintainability, glue language
- Easier to use with TclOO
- Tk scanvas power
- Use of C/Tcl API for C dlls
- Easy deployment thanks to TclKit
Current projects

• SNCF resignalling in France: simulation and TT.

• ERTMS in UK: modelling and simulation up to the training tool and TT. On-board data management tool.

• CTCS in China: modelling and simulation, topology data extraction from the track view, gateway with the production tools.
Demos

• Scanvas editor (objects + views)

• Cambrian simulator + Level crossing simulator + on-board data management tool

• Shitai topology + simulator