

Supporting Embedded Software Development

Doing the Heavy Lifting with Tcl/Tk

Andrew Mangogna

amangogna@modelrealization.com
17th Annual Tcl/Tk Conference
Oakbrook Terrace, Illinois

October 13-15, 2010

It's a Small World After All

It's also a Harsh World

- Small Memory
 - ▶ 32 KiB to 128 KiB Flash
 - ▶ 4 KiB to 16 KiB RAM
- Dedicated Functionality
 - ▶ Runs Only One Program
 - ▶ Low Rate of Field Updates
- Safety Critical
 - ▶ Rigorous Development Process
 - ▶ Specific Testable Requirements
 - ▶ Documented Quality
- Ultra-low Power
 - ▶ Standby Current in μA
 - ▶ Dynamic Current in mA
 - ▶ 99% Asleep

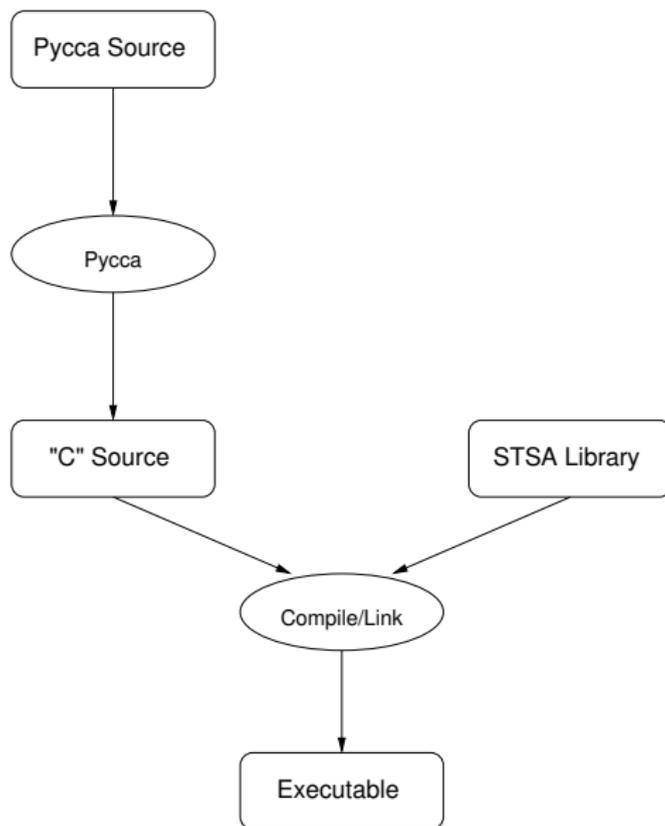
Counter Measures

Operate at a Higher Level of Abstraction

- Specify More, Program Less
- Factor Data Management and Execution Policies Into One Place
 - ▶ Debug, Test and Reuse
 - ▶ Stop Coding Execution Sequencing Over and Over
- Single Threaded Execution Architecture
 - ▶ Single Theaded, Foreground / Background Concept
 - ▶ Closely Match Processor Execution Capabilities
 - ▶ Event Driven
 - ▶ State Machine
 - ▶ Interrupts
- Bind Behavior Before Run Time Using Data

- Pycca Supports Specifying the Data Required by the Execution Architecture
- Pycca Processes a Domain Specific Language
 - ▶ Specify Data Structures
 - ▶ Specify Relationships Between Data Structures
 - ▶ Specify State Machines
 - ▶ State Machine Action Code is "C"
 - ▶ Functions and State Actions are Repackaged and Passed Through
 - ▶ Generates Required Data Structures (e.g. Transition Matrix)
- Written Tcl (no Python, despite the name)
- Delivered as a Starpack

Pycca Workflow



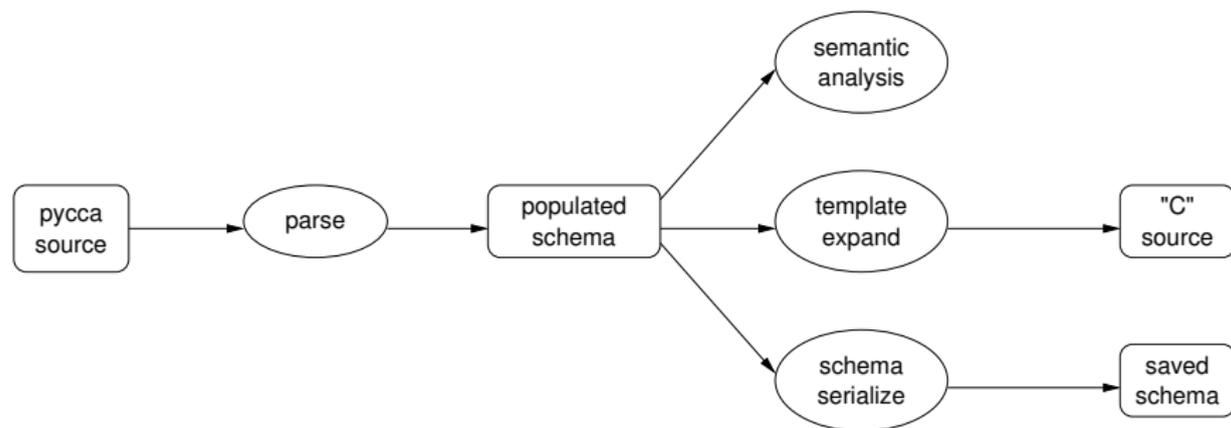
Example Pycca Source

A Very Small Extract

```
machine
  state Idle() {
    ///  
    ///  
    PYCCA_generate(Stop, Motor, self->R2, self) ;
  }
  transition Idle - Run -> FillingForWashing
  state FillingForWashing() {
    ///  
    ///  
    PYCCA_generate(Open, Valve, self->R3, self) ;
  }
  transition FillingForWashing - Full -> Agitating
  .....
end
```

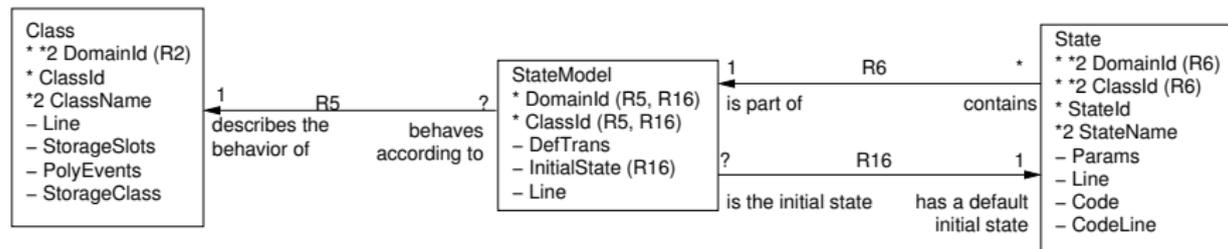
- Parsing
- Internal Data Structures
- Semantic Analysis
- Template Expansion
- Serialization

Pycca Internal Data Flow



Pycca Internal Schema

A Small Extract Only



TclIRAL Implementation of Schema

```
relvar create StateModel {
  DomainId int
  ClassId int
  DefTrans string
  InitialState string
  Line int
} {DomainId ClassId}

relvar association R6\
  State {DomainId ClassId} *\
  StateModel {DomainId ClassId} 1

relvar association R16\
  StateModel {DomainId ClassId InitialState} ?\
  State {DomainId ClassId StateName} 1

relvar create State {
  DomainId int
  ClassId int
  StateId int
  StateName string
  Params list
  Line int
  Code string
  CodeLine int
} {DomainId ClassId StateId}\
  {DomainId ClassId StateName}
```

Semantic Analysis by Query

Finding Isolated States

```
# Find isolated states, i.e. states that have no outgoing or
# incoming transitions
set noIns [relation semiminus $::Transition $::State\
    -using {DomainId DomainId ClassId ClassId NewState StateName}]
set noOuts [relation semiminus $::NormalTrans $::State\
    -using {DomainId DomainId ClassId ClassId StateName StateName}]
set isoStates [relation intersect $noIns $noOuts]
relation foreach isolated $isoStates {
    relation assign $isolated Line StateName
    reporterror "state has no incoming or outgoing transitions"\
        $Line $StateName
}
```

Friends of Pycca

If you've got it, then use it

Pycca (via TclRAL) can save the schema population which other programs can then access.

[pyccaexplore](#) View State Machine Transition Matrix

[pycca2dot](#) Layout State Machine Graphs

[mechtrace](#) State Machine Tracing Package

The Moral to Our Story

- Pycca is a language processing program that looks like a database application.
- The use of relational data structures provides a flexible and powerful way to structure the internals of the program.
- The pycca language is a convenient syntax for populating relational data structures.
- Relational data has a single consistent access algebra that operates on a **set at a time** basis.
- Tcl has many different relation data extensions to match the scale of the application.
- You will be happy with yourself if you encode the rules of your application in data.

- <http://tcl-cm3.sourceforge.net>
- <http://tclral.sourceforge.net>