Supervisory Control Language

Applying Tcl To The Realtime Arena

by

James B. Bassich  jbb@cpu.com
Marc Chevis      mmc@cpu.com
Gerald Lester    gwl@cpu.com

Computerized Processes Unlimited, Inc.
4200 South I-10 Service Road Suite #205
Metairie, LA 70001
(504) 889-2784
Background

CPU’s Mission

Computerized Processes Unlimited, Incorporated is an independent control system integrator serving domestic and international Oil and Gas Energy and other process industries with highly competent consulting services, project management and customized problem solving software.

Primary Projects

• the design and implementation of systems to monitor and control processes
• realtime data integration with corporate databases
• network integration

Platforms (client driven)

• Hewlett-Packard 9000/7xx running HP-UX
• Digital VAX
• PC’s

Foundation requirements

A stable, extensible software foundation to build custom solutions for our clients.
Supervisory Control and Data Acquisition (SCADA)

Purpose

- Collect data from field devices and present the data in meaningful form to operators.
- Provide methods for operators to issue commands to field located controllers.
- Operator must be able to:
  - determine the state of the process easily
  - control the process instinctively

Current State of the Industry

- Current SCADA systems place extreme importance on the interactive operator display to help the operator process data from many sources and respond correctly to changes in the process.

- Small to moderate size systems are now PC based and are user configurable. Unfortunately, configuration is rigid and extensibility is limited.

- UNIX and VMS based systems are used for larger applications. These make use of the multitasking, and operator interface features. Custom integration is still required and can be complex.
Hewlett-Packard’s Realtime Application Platform (RTAP)

- Provides a toolkit for building SCADA applications. Configuration is a combination of using interactive tools and C programming.

- Major components are:
  - realtime database with calculation engine
  - data historian to maintain data for longer times
  - time keeper and event manager to support event processing
  - environment configuration and monitoring
  - scan system for acquiring data and issuing commands to and gathering data from remote devices
  - alarm detection, reaction, and display
  - report system for producing hard copy summary reports
  - user interface tools to support the creation and display of interactive schematics
Goals of the SCL Project

- Develop a product that would allow CPU to become more effective at system integration
- Provide complete development and configuration environment
  - Support custom configuration/application by engineer/technician
- Leverage work done by others
  - RTAP/Plus
  - Other third party products
  - Public domain products
- Extensible by:
  - CPU
  - Third parties
  - Users
- Provide appropriate interface for different levels of users
The SCL Family

A Layer Diagram

<table>
<thead>
<tr>
<th>SCL Base *</th>
<th>SCL Schematic View Manager</th>
<th>SCLX Extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>3P General</td>
<td>RTAP/Plus Application Program Interface</td>
<td>3P Graphics</td>
</tr>
</tbody>
</table>

HP/UX, MOTIF
Peer Type Extensibility

SCL(X) Interpreter

PARSER

Extensions Made in "C"

<table>
<thead>
<tr>
<th>SCL Base</th>
<th>RTAP/Plus</th>
<th>X Windows</th>
<th>SVM (UIP)</th>
<th>Other CPU</th>
<th>Third Party</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Access</td>
<td>Alarms</td>
<td>Buttons</td>
<td>Schematics</td>
<td>Dialogs</td>
<td>SYBASE</td>
</tr>
<tr>
<td>Lists</td>
<td>Database</td>
<td>Labels</td>
<td>Symbols</td>
<td>Widgets</td>
<td>Plotting</td>
</tr>
<tr>
<td>Keyed Lists</td>
<td>Data Historian</td>
<td>Menus</td>
<td>X-events</td>
<td>TBD</td>
<td>Widgets</td>
</tr>
<tr>
<td>Math</td>
<td>Event Manager</td>
<td>Graphics</td>
<td>Menus</td>
<td></td>
<td>TBD</td>
</tr>
<tr>
<td>Strings</td>
<td>Environment</td>
<td>Icons</td>
<td>Messages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unix</td>
<td>Plot Display</td>
<td>Entry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XPG/3</td>
<td>Scan System</td>
<td>etc.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time Keeper</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Watchdog</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Extensions Made in SCL

<table>
<thead>
<tr>
<th>SCL Base</th>
<th>RTAP/Plus</th>
<th>X Windows</th>
<th>SVM (UIP)</th>
<th>Other CPU</th>
<th>Third Party</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td>Point Read</td>
<td>Gauge</td>
<td>Config</td>
<td>Point Methods</td>
<td>TBD</td>
</tr>
<tr>
<td>Parsing</td>
<td></td>
<td>Vector</td>
<td>Macros</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Levelmeter</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SCL Plus Process Model

Database Environment

Operator Environment

GMS
SSMS

SM

SVS
DS
Typical Screens

- Process schematic
- Detail panel
- Plot panel
- "Layered" detail panel
CPU’s TCL Activity

C Code:
27,000 lines

Tcl/Tk Code:
120,000 lines
Conclusion

- SCL has allowed CPU to become more effective at integrating systems.
  - drastically reduced development time by:
    almost eliminating C code programming
    eliminating linking and compiling
    reducing the need for the script writer to be concerned with memory allocation and other operating system "baggage"
  - greater reusability of applications since libraries are easier to build and maintain
  - debugging and testing is made simpler by the interactive interface
  - all of the above results in a substantially reduced turnaround time
Desired Future Directions
for Tcl/Tk

• Better support for multiple interpreters
  • Multiple interpreter support for Tk
  • Standard method of resolving signals when using multiple interpreters

• Further development of canvas
  • partial fill of objects
  • drawing tool for creating objects and defining bindings

• Compiler

• Windows NT

• Continued unencumbered license (no Copy Left)
Appendix A

SCL RTAP Extensions
**Alarm System**

- rrtas_alarm_ac
- rtas_close
- rtas_config_connection
- rtas_open
- rtas_update_msg

**Database**

- rtdb_close
- rtdb_config *item*
  - ADD_NULL_PT, ADD_SCALAR, ADD_TABLE, ADD_VECTOR, ALIAS, ATTR_NAME, CATEGORIES, COPY_ATTR, COPY_BRANCH, COPY_POINT, DEFINITION, DEL_ATTR, DEL_BRANCH, DEL_BR_CHK, EXP_ORDER, GROUPS, MOVE_POINT, PT_CLASS, PT_NAME, RESIDENCE, SET_RECORD_CNT
- rtdb_control *item*
  - CE_ORDER, DISABLE_SNAPS, ENABLE_SNAPS, LOCK_PT, REL_CFI, RUN_CE, SET_CFI, SET_CWP, SET_USAGE, SNAPSHOT, UNLOCK_PT, XFER_LOCK_PT
- rtdb_match_pts
- rtdb_multi_read,
- rtdb_multi_write
- rtdb_open
- rtdb_query *item*
  - ALIAS, ALPHA_ATTRS, ATTRIBUTE, ATTR_ACCESS, ATTR_CNT, ATTR_NAMES, ATTR_ORDER, CATEGORIES, CATEG_NAMES, CE_DEP_REF, CE_DEP_UPD, CE_OPER, CONN_INFO, DEFINITION, DE_TYPE, DIRECT, DIRECT_ATTR, EVENT, EXPR_ORDER, FIELD_NAMES, FIRST_CHILD, GROUPS, GROUP_NAMES, LRL, NEXT_SIBLING, PARENT, PTS_IN_CLASS, PT_CLASS, RESIDENCE, SYM_ABS, SYM_ALIAS, SYM_REL, USAGE
Database (cont.)
   rt
db_read
   rt
db_set
   rt
db_write
   rt
db_unit_write

Historian
   rt
dh_close
   rt
dh_config item
      AUTOREARM, COPY_ABS_POINT, COPY_REL_POINT,
      DELETE_TABLE_POINT, RECORD_DATA,
      TABLE_NAME, TABLE_RESIDENCE, TABLE_SIZE
   rt
dh_control item
      ARM_TABLE, AUTOARM_DISABLE, AUTOARM_ENABLE,
      CLEAR_TABLE, DATAWRAP_DISABLE,
      DATAWRAP_ENABLE, DISABLE_TABLE,
      DISARM_TABLE, ENABLE_TABLE, ONESHOT_TABLE
   rt
dh_open
   rt
dh_query item
      AUTOARM, AUTOREARM, DATAWRAP, OUTPUT_TRIGGER,
      RECORD_DATA, TABLE_CONN_PLIN, TABLE_LIST,
      TABLE_LIST_CNT, TABLE_NAME, TABLE_RESIDENCE,
      TABLE_SIZE, TABLE_STATE
   rt
dh_read
   rt
dh_set

Event Manager
   r
tem_attach_event
   r
tem_change_event
   r
tem_detach_event
Environment System
  rtenv_bind_msg_handler
  rtenv_break_dispatch
  rtenv_dispatch_msg
  rtenv_get_env_dir
  rtenv_get_error
  rtenv_get_my_name
  rtenv_get_option item
    DEBUG, PRECISION, READ_BUFFER, READ_WRITE_STAT
  rtenv_get_proc_name
  rtenv_get_proc_num
  rtenv_get_unix_pid
  rtenv_log_error
  rtenv_msg_recv
  rtenv_msg_send
  rtenv_print_error
  rtenv_query_msg_handler
  rtenv_sched_process
  rtenv_set_my_name
  rtenv_set_option item
    DEBUG, PRECISION, READ_BUFFER, READ_WRITE_STAT

Plot System
  rtpd_control item
    CLOSE_VIEW, CONFIGURE_PLOT, COPY_PLOT,
    COPY_PLOT_UNDER, DELETE_PLOT, HOUR_GLASS_OFF,
    HOUR_GLASS_ON, ICONIFY, ICONIFY_VIEW,
    OPEN_VIEW, OPEN_VIEW_AT, PRINT_PLOT,
    PRINT_PLOT_TO, REFRESH, SET_LIST_BY_PARENT,
    SET_LIST_BY_SIBLING, SWITCH_VIEW, UNICONIFY,
    UNICONIFY_VIEW
  rtdp_query item
    GET_CONTEXT, GET_VIEW_STATUS
**Scan System**

- `rtss_close`
- `rtss_control item`
  - `COLD_RTS_DEVICE`, `COMM_PORT_MODE`,
  - `DISABLE_SS`, `DISABLE_CP`, `DISABLE_SD_SI`,
  - `DISABLE_SD_SI_PT`, `DISABLE_SD_SO`,
  - `DISABLE_SD_SO_PT`, `ENABLE_SS`, `ENABLE_CP`,
  - `ENABLE_SD_SI`, `ENABLE_SD_SI_PT`, `ENABLE_SD_SO`,
  - `DISABLE_SD_SO_PT`, `FORCE_POLL`,
  - `FORCE_POLL_TYPE`, `FORCE_PRBX`,
  - `FORCE_PRBX_TYPE`, `POLL_PERIOD`, `POLL_TYPE`,
  - `PRBX_PERIOD`, `PRBX_TYPE`, `SET_TIME`, `SNAP`,
  - `SNAP_WITH_VERIFY`, `WARM_RST_DEVICE`

- `rtss_open`
- `rtss_query item`
  - `SYSTEM_STATE`, `TASK_STATE`
- `rtss_read`
- `rtss_set`
- `rtss_write`

**SCL Initialization**

- `scl_init`

**Time Keeper System**

- `rttk_cancel_timer`
- `rttk_delay`
- `rttk_start_timer`

**Watchdog**

- `rtwd_cancel_monitor`
- `rtwdcontrol_server`
- `rtwd_report_condition`
- `rtwd_start_monitor`
Appendix B

SVM Extensions
Schematic View Manager

svm_bind
svm_control item
    POLL
    REFRESH
    RUN
    STOP
svm_config_menu
svm_config_menu_item
svm_config_sch
svm_config_sym
svm_control
svm_create_menu
svm_create_sch
svm_destroy_sch
svm_message
svm_query
svm_query_sch
svm_query_sym
svm_set