Spacecraft Control & Monitoring Systems

TSC & CCS Presentation, June 2018
Software Tools for Spacecraft Monitoring

- Single-User
  - SCOE
  - P/L EGSE

- Multi-User
  - CCS
  - SCS

Unified Monitoring & Control
- data model...
- procedures...
- displays...
- archives...
Motivations & History

Compatibility between AIT & Operations:
Common systems used in several missions (e.g. SCOS2000, OC, CGS)

But with legacy systems
• maintenance increasingly difficult, becoming uneconomic, unfeasible
• flexibility & performance not enough for new missions
• installation & upgrade processes very onerous

Assumptions:
• Adopting complete new standards requires familiarization by end users
• Terma too small to set or impose industry-wide standards

Decision: new compatible products
• **TSC** = single user: Instruments, Payloads & SCOE AIT
• **CCS** = multiuser: Satellite level AIT and Operations
Assembly, Integration & Test (AIT)

Satellite Checkout & Operations

Operations

Operations

SCS5

P/L Acceptance

Unit Acceptance

OBSW Acceptance

Spacecraft Acceptance

© The Terma Group 2018, Images courtesy & © ESA, NASA
Technology

All-new implementation (C++, Qt, QML, Tcl)

Compatible with ESA SCOS2000 (MIB, TOPE)
Standard Compliant (CCSDS, ECSS)
Configurable Plugins & Drivers for specific mission protocols
Easy to Install & Set Up

1. Download
2. Install
3. Configure

Immediate downloads for:
- Windows 32/64-bit (.exe)
- Linux 64-bit (.rpm)

Install or Upgrade in Minutes
Configuring CCS

Simple configuration choices:
- e.g. protocols, databases, file locations
Standards Support

Fully-functional TMTC
- Follows & enhances features specified in SCOS2000 MIB (6.9+)
- ECSS PUS *(note - PUS not mandatory!)*
- CCSDS TMTC Packets
- CCSDS TM Frames & TC CLTU, COP1
- ESA & CCSDS Authentication & Encryption

Flexible configuration options, e.g. PUS & CCSDS options, time format…

Enhanced features, e.g.:
- Unlimited MIB field lengths
- Global support for Variable Packets and Parameters
- Deduced parameters in both TM and TC
- Fully configurable length indicators for variable-length parameters
- Different endiainity support
- Different synthetic (derived parameter) languages
External Interfaces

Configurable EGSE packet level plugins:
- EDEN Protocol
- C&C Protocol
- CMDVS (TSC + Windows only)

Protocol control & status available directly from TOPE

One EGSE protocol must be selected, they cannot be mixed in the same program

Configurable MCS frame level protocols:
- NIS (+ optional SLE)
- Cortex
- NDIU LITE

Low level access (e.g. COP1 status) available directly from TOPE

Any combination of MCS protocols can be selected
EGSE and MCS protocols can be combined

Allows e.g. interfacing to satellite onboard computer at frame level using a modem,
At the same time, checkout interfaces at packet level using an EGSE protocol
Each external interface is a named instance, e.g. “Kiruna”, “Svalbard”, “Maspalomas”
Protocol on each interface depends on configured driver class, e.g. “NisDriver”, “CortexDriver”
Driver instances can be connected & disconnected as desired
Protocol drivers are scripted (TOPE) so new protocol can be implemented very quickly
User Interfaces

Flexible displays
- Standard Qt table & tree views
- Custom TOPE Tcl/Tk user interfaces
- Alphanumeric Displays
- Custom QML Displays
- Track
Standard User Interface

TSC & CCS console user interfaces, written in Qt/C++

- Detailed views of TM / TC packets & parameters, guided TC population
- View TOPE sequence status, enter manual TOPE commands
- Support for drag & drop, copy/paste
User can develop custom user interfaces in TOPE (Tcl/Tk)
Comprehensive, indexed, context-sensitive online help
Synoptic (schematic) picture editor & animator
Custom theme (colours & widget styling) can be applied instantly to all windows
Alphanumeric display is a simple tabular view of TM parameter values and states
Enhanced & modernised equivalent to SCOS2000 telemetry desktop
Fluid animation, touch enabled (written in QML)
Allows dynamic creation of new alphanumeric displays (drag & drop)
Includes groups, graphs, sequences, custom QML, synoptic displays & TC stack executor
Custom QML Displays

Allows development of advanced user interfaces (e.g. inspired by sci-fi movies)
Requires QML scripting, results depend on effort invested
System provides QML modules to access TMTC data
Show in TSC/CCS console, or in separate application
(images courtesy & © GISTDA, SSTL)
Presentation of orbit in 3D with call-outs for satellite & data links status, forthcoming events
Accepts live TM updates from CCS/TSC as well as replay from files
Orbit event (AOS/LOS at ground station, eclipses) & swath visualisation
View single satellite or complete fleet, in various projections
Add-on product, see: https://track.terma.com
TOPE : Test & Operations Procedure Executive

Based on Tcl/TK – graphical MMI, numerous plugins
Expressive, robust, good performance, easy to learn, well-documented
Support tools: debugger, syntax checker
TOPE Detailed Features

Syntax and basic commands derived from SCOS2000 TOPE

- Send TC, fetch & subscribe to TM parameters & packets, …
- Start & control status of TOPE sequences
- Shared variables

Numerous enhancements, e.g.:

- Full control of starting & stopping sessions
- Direct access to all properties of archived objects (packets, events)
- TM packet simulation
- TC subscription, modelling
- Detailed control of COP1
- Local packet extraction
- Full access to frame-level coding & decoding

Note: main difference when migrating from SCOS2000 TOPE is increased speed
TOPE in CCS

Global status of all sequences in the whole system

- Sequences can be manipulated
e.g. suspended, resumed, restarted – if permissions allow it
- Actions distributed, including responding to prompts
## TOPE Debugger

Visual debugger for TOPE

- Set breakpoints
- Examine & set variables

### Stack Frames

<table>
<thead>
<tr>
<th>Stack Frames</th>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 global</td>
<td>argc</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>argv</td>
<td>C:/Users/aha/Documents/TESTENV/TSEQ/Rep...</td>
</tr>
<tr>
<td></td>
<td>argc0</td>
<td>(...)</td>
</tr>
<tr>
<td></td>
<td>auto_index</td>
<td>(...)</td>
</tr>
<tr>
<td></td>
<td>auto_indexarglist</td>
<td>(...)</td>
</tr>
<tr>
<td></td>
<td>auto_path</td>
<td>(C:/program files/CCS/tcl/lib/tcl8.6) {...}</td>
</tr>
<tr>
<td></td>
<td>env</td>
<td>(...)</td>
</tr>
<tr>
<td></td>
<td>errorCode</td>
<td>NONE</td>
</tr>
</tbody>
</table>
|              | errorInfo | Error: can't connect to debugger!

### Code Snippet

```
# File: $Id: ReplayTest.tcl 20655 2017-11-06 09:09:56Z ab $  
# Description: Verifies the several ways of replaying a session  
# Last edited by: $Author: ab $ on $Date: 2017-11-06 10:09:56 +0100 (Mon, 06 Nov 2017) $,  
# (c) Terma 2014 - All rights Reserved  

namespace import AUTO/TEST:4
-- namespace import TOOLS:*  
12
13 namespace eval ::ReplayTest ()  
14
15 # automatically set the revision, do not edit this  
16 setrevision [$Id: ReplayTest.tcl 20655 2017-11-06 09:09:56Z ab $]  
17
18 testheader "ReplayTest"  
19 newTest SESS_CTRL_SWITCH_TO_RPL  
20 # map SKFID to the packet id's used by the simulator  
21 array set ::simFktId [1000 1111 2000 2222 3000 3333 4000 4444 5000 5555]  
22 set seeds [lsort -integer [array names simFktId]]  
```

**Code: OK Result:**

```
| stopped | ReplayTest.tcl : 10 |
```
Performance

Needed for…

High-rate interfaces used in EGSE (e.g. SpaceWire)
X-band & Ku-band RF data rates & enabled applications (e.g. CFDP)
Floods of errors during checkout (e.g. massive MIB errors)
Large constellations
Maximum Performance

<table>
<thead>
<tr>
<th>Data</th>
<th>Maximum Rate</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM Parameters</td>
<td>~ 150,000 update/s</td>
<td>With dependencies</td>
</tr>
<tr>
<td>TM Packets</td>
<td>~ 1500/s</td>
<td></td>
</tr>
<tr>
<td>Bandwidth</td>
<td>~ 5 Mbit/s</td>
<td>Monitored housekeeping</td>
</tr>
<tr>
<td></td>
<td>~ 80 Mbit/s</td>
<td>Unmonitored, e.g. large science packets</td>
</tr>
<tr>
<td>Telecommands</td>
<td>~ 500/s</td>
<td>TSC, with verification</td>
</tr>
<tr>
<td></td>
<td>~100/s</td>
<td>CCS, with verification</td>
</tr>
</tbody>
</table>

Legacy systems
• Struggle to meet current performance requirements, especially in checkout & AIT
• Expensive to maintain or upgrade
Different ways to access the CCS archive

- CCS history browsers
- Retrieval from TOPE scripts
- Retrieval tools
- Backup and restore tools
Indexes stored in an RDBMS:
- Choice of MySQL, Maria DB, PostgreSQL *(note - TSC uses SQLITE)*
- Enables searching for records using plain SQL
- Easy to integrate with other systems & languages (PHP, Java, Python, …)
- TM parameters stored on change

Raw data stored in segmented binary files:
- Header contains “properties” (Qt serialized data format)
- Body contains raw packet
- Nanosecond time stamp resolution
- Every TC stage recorded (detailed history of TC verification stage timing)

Live data signaled via broadcast UDP “Heart Beat” datagram
- No “round robin” distribution to client applications
- Client applications can skip records if data arrives too fast
- Raw file access benefits from OS file system optimization (buffer cache)
- System relatively unaffected by addition of new clients
Open any session and browse history *(note CCS does not have to be running)*
- Packets (TM & TC) including parameter values & locations
- Events
- TOPE sequence live status & history
Information is derived from archive (independent of MIB)
CCS Archive Retrieval Tools

Configurable output formats (e.g. tsv, csv, xml, txt, json)
- Retrieve filtered requests across several sessions
- No need to start CCS
- Retrievals can be called from TOPE

**PART / TM Parameter Retrieval**

**DART / Mixed Data Retrieval**
CCS Archive Session Management

RAT : Results Archive Tool

- Manages backup & restore of CCS sessions
- Actions can be automated from TOPE
Special Features

Add flexibility for test & operation phases

Standard features mimic behaviour of ESA SCOS2000

Additional features motivated by experience during AIT
MIB databases may come from different suppliers, arriving at different times, changing
- Need to merge from different sources (Platform, Payloads, Instruments, EGSE)
- Drop / load / continue (no need to restart)
- Online access / directly view MIB contents with search & filter
Clean separation of data from different sources
SVF (Software Validation Facility) often includes a simulator and onboard computer emulator

- The simulation time frame may vary (pause, resume, speed up, slow down)
- In SVF mode, TC verification timers and TOPE sequence timers vary with the simulated time

For Terma emulator (TEMU), see also: https://temu.terma.com
TM packets can be simulated according to the MIB definition (inverts TM processing)

- Acquire data from a device then publish it in the layout of the spacecraft (platform simulation)
- Generate SCOE telemetry (act as SCOE controller)
- Test development and debugging (exercise displays, exercise TOPE scripts)
CCS Web Services

Allows remote clients to use a REST API to:
- Start/stop sessions (& run automated tests)
- Retrieve TM parameters

Allows increased automation (e.g. automated production line, automated operations)
End customers (e.g. ESA) sometimes ask for test evidence, e.g. usage of MIB objects

- Tests may involve a large number of sessions and large number of objects
- Some sessions are formally significant, while many sessions are “scratch”
- Due to potential data size, selecting and merging has to be very efficient
- Different missions and companies have different report formats
To model a TC, activate verification for a TC, even if we did not send it
Cyclic TC: completes when repeat count property reaches zero.
Verification (re)opens according to model (e.g. based on orbit telemetry) in TOPE
Optionally models verification of nested TC (commands within commands)
Complex Use Cases
Advanced applications for AIT
Typical configuration:

- High-end workstation (e.g. 8 cores) running TSC
- Electrical front-end equipment (e.g. Power + MIL 1553 + SpaceWire)
- MATLAB used for graphing and visualization (mission customized)
- TSC controls front ends & processes instrument TM & TC using instrument MIB
Instrument Control Unit with MIL-1553 front-end for Housekeeping TM and TC
Science data: 4 x SpaceWire interfaces, 8 x CCD source
TOPE interfacing to MATLAB
TOPE interfacing to Eurosims Simulator
Typical configuration:

Mission P/L and platform suppliers are different companies
P/L and platform each tested with the other absent, until payload delivered
Platform uses CCS, payload uses TSC, configured with same EGSE protocol (e.g. EDEN, C&C)
TM simulation capabilities allow platform to be simulated & platform TC to be sent to P/L
Allows coordination of MIB contents & inter-site testing prior to shipment
User Roles & Deployment

Issues considered:

User roles & privileges (protection & security)
Speed of deployment (quick, secure, configuration of different users)
User Roles & Deployment Detail

Login relies on operating system (password validation, file permissions)

Access to application features is based on role, e.g. “Operator”, “Guest”
- Capabilities define access to a feature e.g. “canSendTelecommand”
- Capabilities are configured on/off by role.
- Unless prevented, users can change role instantly

Default set of “typical” roles & capabilities is usually configured at installation
- New roles can be defined, or custom capabilities of an existing role
- Specific capability “canSwitchMode”; defines whether user can change to another role
- Role changes, if allowed, are applied instantly

“Global Settings” feature:
- Defines all roles and capabilities for all users, differentiated by group ID
- Speeds up deployment & configuration for big systems
- Improves security (prevents users from configuring their own capabilities)
Constellation Support

Issues considered:

Support for multiple spacecraft in one host

Avoid complex IT setup, preserve easy installation & configuration

Fleet may be homogeneous (identical, similar), or all different

User interface design for managing large fleets
Multiple Spacecraft: Fundamentals

One console (TSC/CCS) filters for one SCID
Multiple parallel TM & TC streams
COP1 flow control, if applicable

But… within one console instance:
- Monitoring is for one (selected) spacecraft
- S/C has context, e.g. on board time

Problem: if S/C is switched, history switches; the alarms raised at a certain time are for a single spacecraft.

Solution: multiple consoles each selecting one SCID separate monitoring context and archive history

See “SimLiteDemo” test pack
Preserve speed & simplicity of installation:

- Install, configure, click constellation option, start
- Configure S/C name, and base, and number desired
- Configure up to 9999 (typical maximum per host = 64, depends on platform)
- Same TCP/IP ports as single-spacecraft CCS, simple host & license setup
Constellation User Interface

Operators must not be overwhelmed
• Call attention only when necessary
• Highly automated operations
• Summarise status at high level

Mission-custom interface

NATS messaging infrastructure [www.nats.io](http://www.nats.io)

Direct interface available for
• Contingency & detailed investigation
• Direct commanding

(images courtesy & © GISTDA)
Cluster of VM’s used for large constellations

- E.g: 4 VM, 50 small S/C per VM = total 200 S/C
- One fully independent TSC for each S/C, N instances within each VM
- Archiving to single common RDBMS, CCS HIST viewers available
- Management MMI within each VM

Minimizes: CPU / Memory of a single server

- Standard “retail” hardware or cloud VM
- Keeps platform costs under control

Minimizes: IT, installation and setup complexity

- One VM = one system, configured once, one IP address
- Avoids allocating N x different sets of IP ports for different S/C
- Keeps IT and engineering costs under control

Maximizes: flexibility

- Within 1 VM, N x S/C are similar but can be completely different
- No fixed number of operator consoles
- Constellation lifetime may be for decades; it will evolve
Flexible: arbitrary number of S/C per VM, arbitrary number of operator WS

For small S/C, moderate TM/TC rates, consider e.g: 1GB RAM, 1 vCPU per S/C

Suitable for cloud deployment
Highly automated maintenance processes

Daily build & automated test (functional and partial testing of user interface)
Daily build of documentation & online help
Continuous gathering of quality metrics
Formal testing every ~3 months
System is Terma IPR

Uses limited number of 3rd-party tools:

- Included in installation:
  - Qt 5 (commercial license)
  - Tcl 8.6 (BSD license)
  - NATS (Apache 2.0 license)
  - SQLITE (Public domain)
- Not included in installation:
  - RDBMS (MySQL, Maria DB, PostgreSQL, AWS Aurora)

License is perpetual:

- Warranty (e-mail support) 1 year included in license
- Subsequent years: 20% of license cost
- Support WIKI available to customers with active licenses
Find out more…

ccs.terma.com
tgss.terma.com

About Terma:

www.terma.com/press/newsletter
www.linkedin.com/company/terma-a-s
www.youtube.com/user/TermaTV