PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

The preliminary design of the G-CLEF spectrograph instrument device control system

Ian N. Evans, Daniel Durusky, Janet DePonte Evans, Thomas Gauron, Cem Onyuksel, et al.

Ian N. Evans, Daniel Durusky, Janet DePonte Evans, Thomas Gauron, Cem Onyuksel, David A. Plummer, "The preliminary design of the G-CLEF spectrograph instrument device control system," Proc. SPIE 10707, Software and Cyberinfrastructure for Astronomy V, 107071F (6 July 2018); doi: 10.1117/12.2312242

Event: SPIE Astronomical Telescopes + Instrumentation, 2018, Austin, Texas, United States

The preliminary design of the G-CLEF spectrograph instrument device control system

Ian N. Evans, Daniel Durusky, Janet DePonte Evans, Thomas Gauron, Cem Onyuksel, and David A. Plummer

Harvard-Smithsonian Center for Astrophysics
60 Garden Street, MS-81, Cambridge, MA 02138, USA

ABSTRACT

The Giant Magellan Telescope (GMT)-Consortium Large Earth Finder (G-CLEF) is a fiber-fed, precision radial velocity optical echelle spectrograph. The preliminary software design incorporates a hierarchical, multi-level state machine. At the lowest level, the state machine utilizes GMT-provided frameworks to communicate with the hardware. At higher levels of abstraction, the design makes extensive use of State Chart Extensible Markup Language (SCXML) representations to define the operation of the instrument. The functionality of the design can be validated by executing these representations. The incorporation of an interpreter to directly execute the SCXML as a component of the control system is being investigated. The approaches used to develop the preliminary software design concept are described, the use and utility of SCXML for instrument control is discussed, and the application of the preliminary design to a subset of G-CLEF subsystems is demonstrated.

Keywords: control system, G-CLEF GMT, instrument control, model-based systems engineering, software design process, SCXML

1. INTRODUCTION

The first major scientific instrument selected for development for the Giant Magellan Telescope (GMT) is the GMT-Consortium Large Earth Finder (G-CLEF),[1–4] a fiber-fed, precision radial velocity (PRV) optical echelle spectrograph. G-CLEF is a general-purpose instrument with multiple observing modes that will address many fundamental questions in the areas of exoplanet studies, stellar astrophysics, and cosmology.[5] However, the instrument design is strongly driven by the requirements for superb radial velocity measurement stability,[6] with the capability of making single PRV measurements with a precision of 40–50 $\mathrm{cm\,s^{-1}}$, and an ultimate PRV stability requirement (by combining multiple measurements) of 10 $\mathrm{cm\,s^{-1}}$.

The G-CLEF instrument device control subsystem (IDCS) controls all aspects of the instrument hardware, including active feedback loops that are required to meet the G-CLEF PRV stability requirements. The IDCS must integrate tightly with the GMT observatory control system, and is required to conform to a reference architecture[7] defined by the GMT Organization (GMTO) Software and Controls (SWC) group. This integration is achieved in part by developing the G-CLEF IDCS using a set of device control frameworks that are being designed and developed by the GMTO. Advantages of this approach are many, including coordinated electronics cabinet power management, tight integration of G-CLEF instrument and GMT facility safety systems, uniform user interfaces, and access to GMTO-developed functionality such as alarm, logging, and telemetry services, and science data archiving.

The control system utilizes an EtherCAT[*]-based hardware architecture to control all instrument functions, with the exception of science and technical camera control and data acquisition. The preliminary software design concept incorporates a hierarchical, multi-level state machine that coordinates instrument operations. A model-based systems engineering methodology[8] is being used to develop this design.

At the lowest level, the G-CLEF control system state machine utilizes GMTO-provided hardware control and I/O frameworks to communicate with the EtherCAT hardware and manage commanding and telemetry at

Send correspondence to ievans@cfa.harvard.edu

[*] https://www.ethercat.org

Software and Cyberinfrastructure for Astronomy V, edited by Juan C. Guzman, Jorge Ibsen, Proc. of SPIE Vol. 10707, 107071F · © 2018 SPIE · CCC code: 0277-786X/18/$18 · doi: 10.1117/12.2312242

the component level. At higher levels of abstraction, the preliminary design makes extensive use of State Chart Extensible Markup Language (SCXML)[†] representations to define the operation of the instrument. These representations are executed using a SCXML interpreter to validate the functionality of the design. The incorporation of an interpreter that directly executes the SCXML as a component of the operational IDCS is presently being investigated.

This preliminary software design concept is presently being developed further and validated through application to several G-CLEF subsystems, including the vacuum systems, thermal control system, and active leveling system. These subsystems were selected partly because their hardware design and operational procedures are sufficiently mature, and also because they exercise several of complex state machine operational requirements. The latter include fault tolerant operations, and possible independent state modification by an autonomous Safety Integrity Level[9] (SIL) 3 safety system that operates independently of the main control loops. The thermal control system design is particularly challenging since it must integrate telemetry from large numbers of temperature sensors and control the loads applied to dozens of heater panels, to maintain the G-CLEF optical bench temperature with a precision of $\pm 0.001^\circ$ C even under the circumstance of failed sensors and/or heaters.

In this paper, the approaches used to develop the preliminary software design concept are described, the use and utility of SCXML for instrument control is discussed, and the application of the preliminary design to a subset of G-CLEF subsystems is demonstrated.

2. G-CLEF HARDWARE DESCRIPTION

The G-CLEF instrument is composed of several major subsystems located at different stations on the GMT, as shown conceptually in Figure 1. Each of these subsystems, with the exception of the passive optical fiber chains, requires software control and provides housekeeping telemetry and/or science data.

The G-CLEF front-end assembly (GCFEA) is responsible for directing light from the telescope to the spectrograph for analysis. The GCFEA is split into two subassemblies: the GCGIR and the GCIP. The GCGIR, mounted on the GMT *Gregorian Instrument Rotator* (GIR), includes a deployable tip-tilt tertiary mirror to intercept the telescope optical beam, and relay optics to direct and focus the light onto a set of optical fibers on the GCIP. During most GMT operations, the GIR rotates to correct for the field-rotation introduced by the alt-az telescope mount. However, when G-CLEF is observing, the GIR must be locked in position so that the optical beam can be directed to the GCIP, which is mounted on the fixed portion of the GMT *Instrument Platform* (IP).

The GCIP includes a fiber selector system that allows any of the various G-CLEF instrument modes to be selected by translating the appropriate optical fiber input station into the telescope focus. Each fiber input station includes a science fiber (pupil-sliced for the PRV modes) and a pair of sky fibers (all with associated shutters) that can be used to record the sky background spectrum (only a single sky fiber aperture can be utilized at one time since the spectral orders overlap). A pair of calibration light injector masts can be positioned in front of either the science or sky fibers to inject calibration light from the G-CLEF calibration light source assembly (GCCLS) into the optics. If a measurement of the sky background is not required, then calibration light may be injected into a sky fiber during a science observation, providing a simultaneous calibration capability. Also located on the GCIP is the flexure control camera (FCC), which provides the telescope control system with guide and focus corrections due to flexure between the GMT *Acquisition, Guiding, and Wavefront Sensing* (AGWS) system and the GCIP focal plane. The FCC is also used to image the local field of view for the purpose of target acquisition.

Optical fiber chains (GCSFIB+GCCFIB) relay the optical signals from the G-CLEF front-end to the spectrograph, and also transmit light generated by the GCCLS for injection into the GCIP optics.

The G-CLEF Spectrograph Assembly (GCSPECT) is mounted at the GMT *Gravity Invariant Station* (GIS) located on the rotating GMT azimuth disk. GCSPECT includes the spectrograph optical bench and associated hardware. To minimize coefficient of thermal expansion (CTE) effects on the spectrograph optics, the optical bench must be maintained at a constant temperature ($20.0 \pm 0.001^\circ$ C) by means of an active precision thermal

[†] https://www.w3.org/TR/scxml/

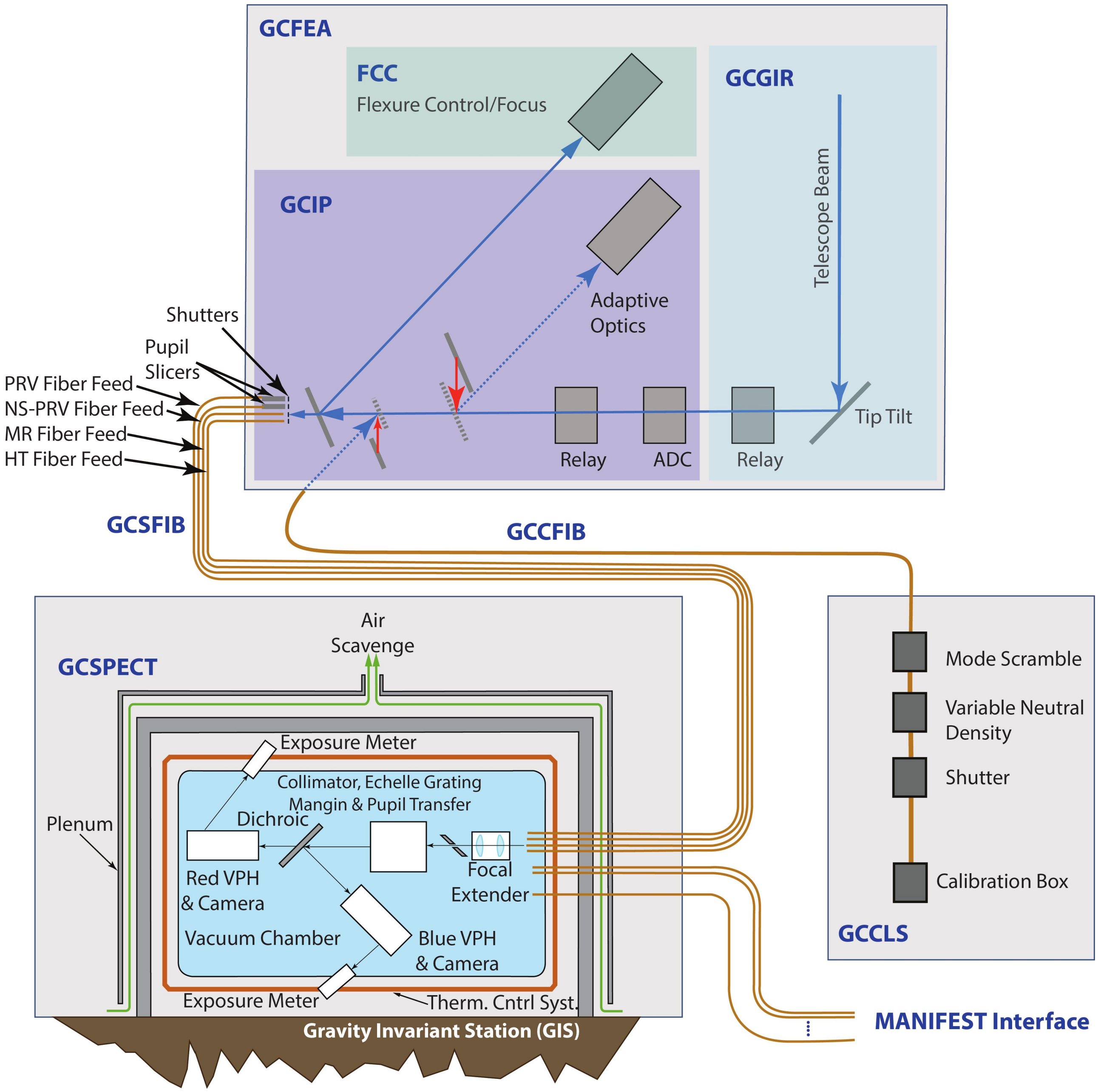

Figure 1. The G-CLEF instrument major functional subsystems.

control system.[10] The optical bench is installed in a vacuum chamber within a larger thermal enclosure. The latter is maintained at $17.5 \pm 0.5^{\circ}$ C via a HVAC system in order to cold-bias the vacuum chamber. The precision thermal control system applies heat to to maintain the optical bench at 20.0° C via a set of 48 heater panels surrounding the vacuum chamber, plus additional sets of heaters located on support feet and vacuum feedthroughs, since the latter provide direct thermal conduction paths to the exterior. Although the spectrograph is mounted at the GMT GIS, residual non-verticality of the GMT azimuth axis, together with distortions induced in the GMT azimuth disk as the latter rotates, will result in unacceptable changes in the direction of the gravity vector at the level of precision required by G-CLEF. To mitigate these effects, two of the three feet that support the vacuum chamber are controlled by an active leveling system to ensure that the optical bench remains in a constant plane during PRV mode operation.

Mounted immediately below the GIS on the GMT *Auxiliary Utility Platform* (AUP) is the GCCLS. The GCCLS includes several built-in wavelength calibration lamps, as well as optical fiber inputs for external light sources, which may include a laser-driven optical continuum source, an optical frequency comb, and a Fabry-Perot Etalon source. Any two lamps or external sources can be selected at one time, and their light is transmitted using optical fiber to the spectrograph front-end assembly, via a pair of shutters and variable neutral density filter wheels.

The GCSPECT+GCCLS, GCIP, and GCGIR subsystems are electrically remote from each other in the sense that their control signals pass through different sets of cable wraps and are combined in the G-CLEF *Device Control Computer* (DCC) that is located in a remote GMT electronics room.

3. CONTROL SYSTEM HARDWARE

Following the GMT IDCS requirements,[11] almost all G-CLEF components are controlled via EtherCAT interfaces. EtherCAT is well suited to large scale industrial automation systems, and many components used in the G-CLEF control system (e.g., motion drives, vacuum gauges, flow controllers, a-to-d and d-to-a converters), are readily available with EtherCAT interfaces. However, in many ways G-CLEF is more like a piece of precision laboratory equipment rather than an industrial robot. There are a significant number of G-CLEF components, principally designed for laboratory use, that do not have available EtherCAT interfaces. The majority of them use serial (RS-232 or RS-485) interfaces with custom communications protocols, and they are connected to the fieldb us using EtherCAT to serial interface modules. This provides cable redundancy up to the EtherCAT interface (but not beyond), but does not allow us to take advantage of the "hardware adapters" that are included in the GMTO SWC frameworks. A small number of G-CLEF control components, as well as science and technical camera image buffer output, require Ethernet connections via TCP/IP, and dedicated network connections will be used for these components. Finally, the commercial off-the-shelf CCD imagers that are incorporated in the FCC, pupil alignment camera, and exposure meters are only controllable through USB interfaces. These devices will be connected to the G-CLEF DCC using dedicated fiber transceivers, with the control software interfacing using a vendor-supplied Linux device driver library.

Most of the G-CLEF components will be controlled by a DCC running a realtime version of the Linux operating system and a set of software frameworks provided by the GMTO. The frameworks that run on the DCC provide many of the capabilities needed to implement the state machines, control loops, hardware adapters, and EtherCAT communication that are required to operate the G-CLEF hardware. This design is somewhat different from the typical industrial automation approach where control software is uploaded to one or more Programmable Logic Controllers (PLCs) that are embedded directly in the EtherCAT buses.

We do use dedicated PLCs for control in a few cases where operation independent of the DCC is required. Because of the large thermal mass of the spectrograph, a significant amount of time is required to achieve thermal equilibrium with the stability necessary for PRV mode observations. Under normal circumstances, we want to operate the precision thermal control system at all times, even if communication with the DCC is lost (e.g., due to the need to perform a software update). Therefore, we plan to upload the control software into a dedicated PLC that is embedded in the precision thermal control EtherCAT bus. A similar approach is used for power management of various electronics cabinets.

System safety is hardware-based, and uses a set of dedicated hardware inhibits that are activated using SIL 3 rated Safety-over-EtherCAT components. Each safety bus includes a safety-rated PLC that reacts to safety inputs and triggers safety outputs as required. Safety triggers that are purely internal to G-CLEF, for example vacuum system valve inhibits or CCD heater thermal protection, are handled directly by the local safety bus. External safety triggers such as Emergency Stop, or G-CLEF safety triggers that require a GMT-wide response (e.g., a telescope motion inhibit) are communicated via a SIL 3 rated interface with the GMT Interlock and Safety System.

The G-CLEF control system hardware package includes more than 250 sensors (~ 50 different types) and roughly 170 actuators ($\sim$70 different types).

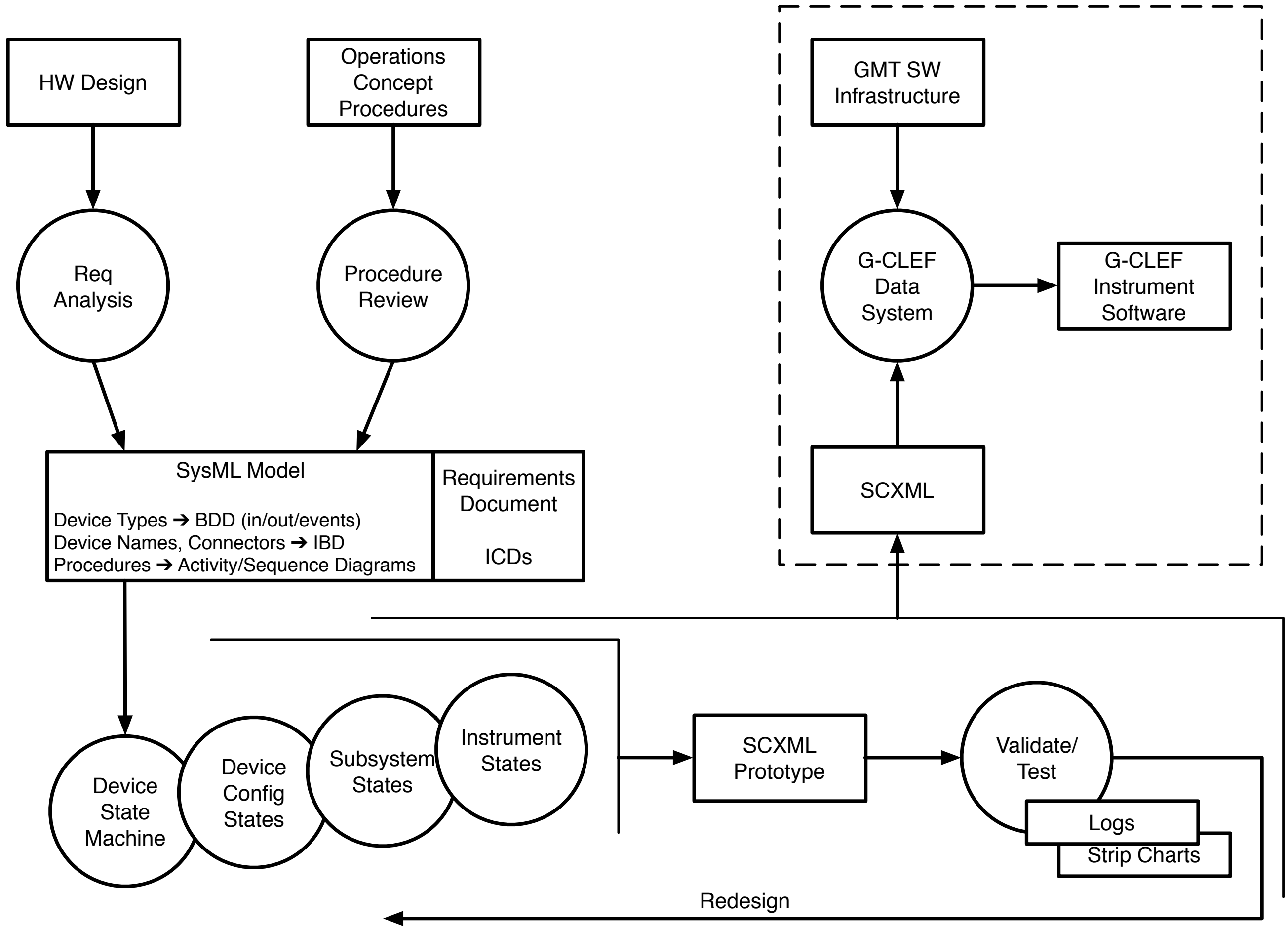

Figure 2. A conceptual outline of the G-CLEF control system software design process. The elements included in the dashed box are applicable if the SCXML state machines are included directly in the control system implementation.

4. PRELIMINARY DESIGN METHODOLOGY

To comply with the GMT IDCS requirements, the G-CLEF control system preliminary design uses a state machine description, where the desired instrument configuration at any time is specified by a set of goal states. The control system preliminary design is being developed using a model-based systems engineering approach. A conceptual outline of the G-CLEF control system software design process is shown in Figure 2. Requirements included in a high-level IDCS requirements document,[12,13] together with a functional requirements analysis developed from the G-CLEF instrument hardware design,[14] are combined with detailed operational sequences developed from a review of the operations concept procedures[15,16] to model the control system in SysML. The state behavior of each subsystem is extracted from the SysML model to create prototype state machine representations in the State Chart Extensible Markup Language (SCXML). The SCXML model can be executed using an SCXML interpreter, and this enables the state machine behavior to be tested and verified as part of the preliminary design. Iterative updates to the model based on feedback from the SCXML execution allow incremental improvements to the model as the sophistication of the simulation is increased, for example supporting fault-tolerant design and understanding of failure modes.

The G-CLEF preliminary software design consists of a hierarchical, multi-level state machine. At the lowest level, for each state variable the control loop combines the outputs from one or more sensors to estimate the current value of that state variable, and that value is compared with the current goal value for the state variable. A corrective action is then identified that is applied through the action of (usually) a single actuator, and the

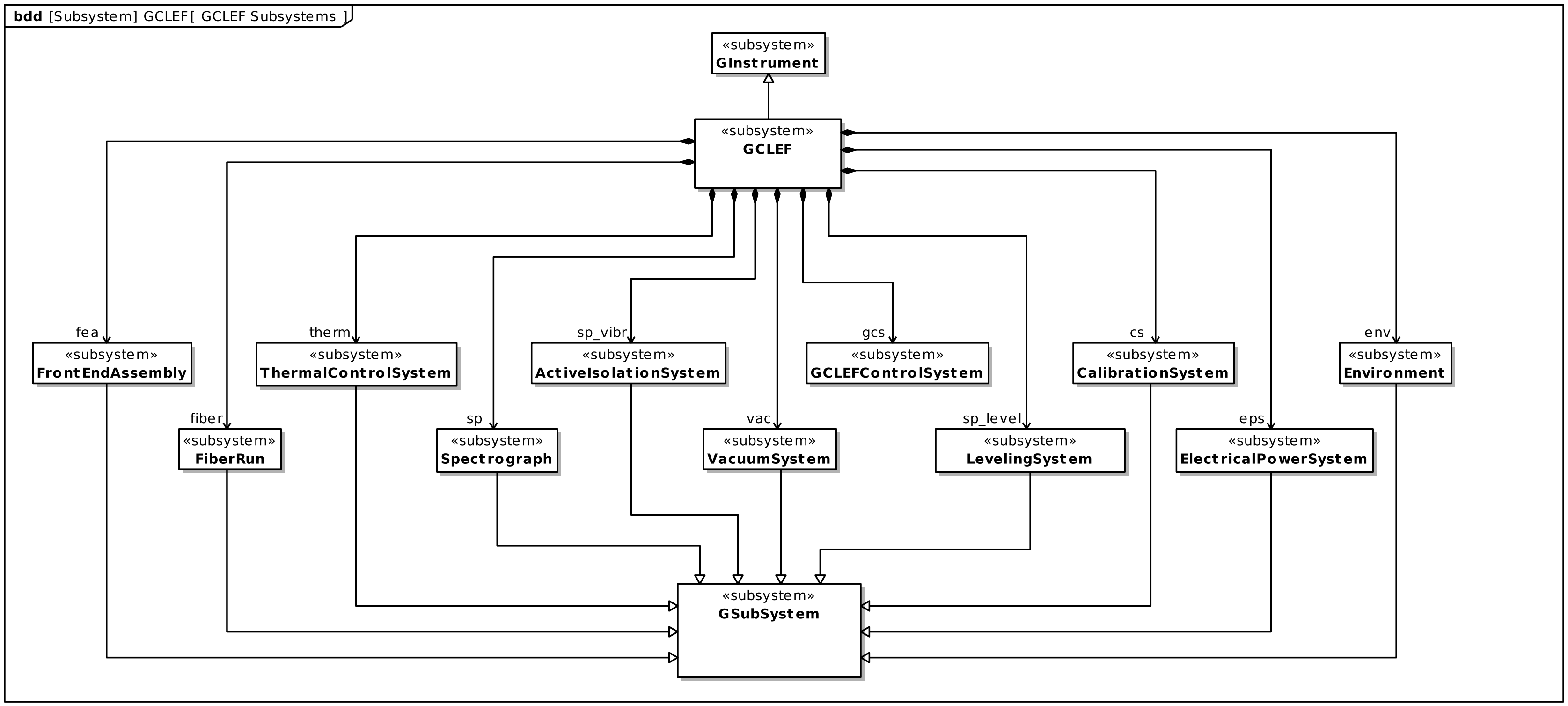

Figure 3. The top-level G-CLEF control system software components.

loop repeats at a specified rate. Control at the single actuator level is typically necessary for fault diagnosis and maintenance activities, and applying the state machine approach down to the lowest level enables these "manual" activities to be handled in software using the same mechanism as is used by the automated system. Additionally, this design allows the control system to recover automatically from manually commanded states that may never occur under normal circumstances.

At higher levels of abstraction, the control loops manage each of the lower-level loops by specifying the current goal value for each of the lower level state variables and monitoring their actual values. When all of the next-lower level subsystems' state variables match their goals, possibly within some tolerance that depends on the subsystem, then the current subsystem has achieved its goal. This hierarchy is repeated for as many levels as is necessary to represent the G-CLEF system to the level at which the overall observatory control system interfaces to the instrument.

The top-level G-CLEF control system software components are shown in Figure 3.

As an example, the precision thermal control system is responsible for maintaining the spectrograph optical bench temperature at a constant temperature. The current values from a set of precision temperature sensors are combined with additional outputs, such as the operational status of each sensor, to estimate the current temperature of the optical bench (the state variable of interest). If the current temperature is less than the goal temperature of $20.0 \pm 0.001^{\circ}$ C, then the control response is to power-on a set of heaters for an interval that is determined using a set of calibrated, nested PID loops. The appropriate heater profile is dependent on a lower-level state variable that wraps up the status of the heater system, considering heater locations (*e.g.*, heaters attached to vacuum chamber feet or feedthroughs require a different profile from the heater panels wrapping the vacuum chamber) and any identified heater failures.

At the time of writing, the preliminary design analysis is complete at the lowest level for several of the G-CLEF subsystems, and the preliminary design for the higher-level state machines for these subsystems has started. This is achieved through a combination of careful analysis of the lower-level control loops and requirements to identify the best choice for goal state variables at each level of abstraction, and also through prototyping of the hierarchical state machines using SCXML.

5. SCXML STATE MACHINES

SCXML is an XML-based markup language that is designed to encode state machine definitions. Furthermore, several implementations of SCXML support interpretive execution of the state machine, and can be readily configured for simulation. The G-CLEF preliminary software design process currently uses the *JSSCxml* JavaScript State Chart interpreter[‡] to develop and validate the state machine design through prototyping.

The executable SCXML state machine prototypes include three main components: the command handlers, the simulator, and the prototype hierarchical state machine itself.

The command handlers are further subdivided into an internal command handler and an external command handler. Both command handlers execute a loop that reads a command from the input stack, validates the command, and then executes the appropriate component of the state machine. The external command handler takes a goal state as an argument and initiates the specified state machine at the appropriate level of abstraction. The internal command handler is responsible for executing the steps that implement an action, and then waiting for any response that results from that action (for example, a sensor readout). The response action is placed on the internal command execution queue and managed by the internal command handler.

The simulator component instantiates the "hardware" responses to state machine actions. In its simplest form, this response may be to change the state of a sensor output based on the executed action, for example changing an output from "disabled" to "enabled" when an "enable" action is executed. The simulator may also implement more complex responses if the corresponding sensor values are used for state variable estimation. Enabling a power supply may require monitored output voltages and currents be set to their nominal values, for example. The level of sophistication of each simulator component depends on the level of fidelity required to validate the state machine design robustly. For example, to accurately simulate the behavior of the precision

[‡] https://www.jsscxml.org

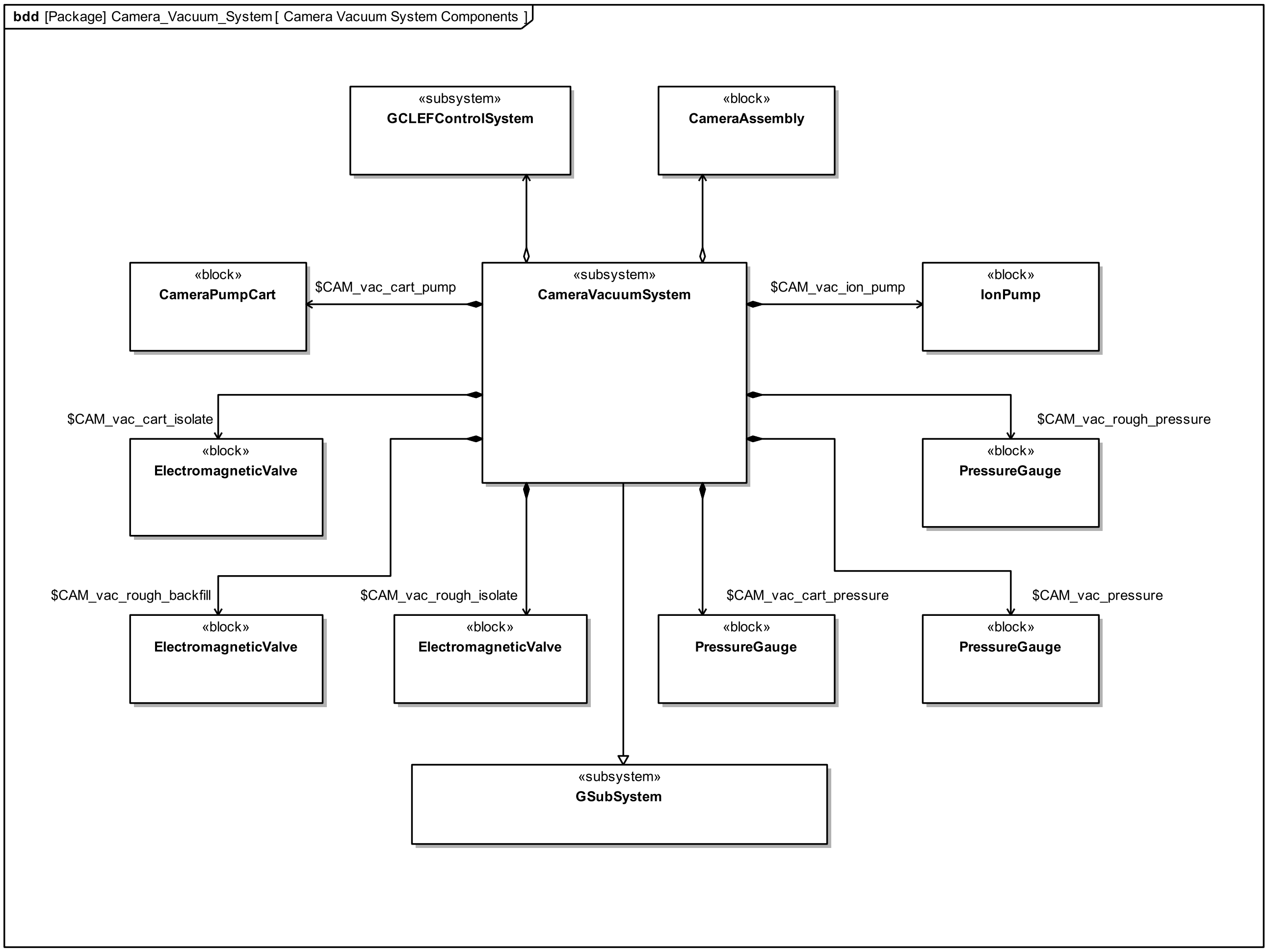

Figure 4. The G-CLEF science camera vacuum control system components.

thermal control system an accurate thermal model may be required. If this level of complexity is necessary to validate the operation of the state machine, then hardware prototyping may be a more appropriate solution. Simulator transitions are normally instantaneous. If time-resolved responses are required (*e.g.*, tracking the drop in pressure in a vacuum system as a function of time), then a model of the temporal behavior of the system at each step must be incorporated into the simulator. Unless a specific temporal response profile is necessary, a simple linear or logarithmic model (*e.g.*, pressure dropping by a factor of 2 at each step) is implemented by the simulator. The simulator component supports validation of state machine fault tolerance and safety by providing simulated hardware failure responses for the former, and simulated safety inhibit signals for the latter (although system safety is based on an independent SIL 3 rated hardware system, the control system software should recognize safety inhibits and react accordingly).

The prototype hierarchical state machine is the design implementation of the state machine under test, and includes all of the state machine components at each level of abstraction. As an example, consider the G-CLEF camera vacuum control system. The G-CLEF instrument includes independent blue and red science cameras that record the echellogram when a science exposure is taken. Each camera includes a $10\,\mathrm{K} \times 10\,\mathrm{K}$ pixel CCD detector that is maintained at -120° C in a vacuum. The G-CLEF camera vacuum control system components for the red science camera are shown in Figure 4. There are several activities related to the camera vacuum system that must be implemented by the G-CLEF control system. One such activity is the vacuum pump down sequence shown in Figure 5. For simplicity, fault tolerance and safety actions are not included in this example.

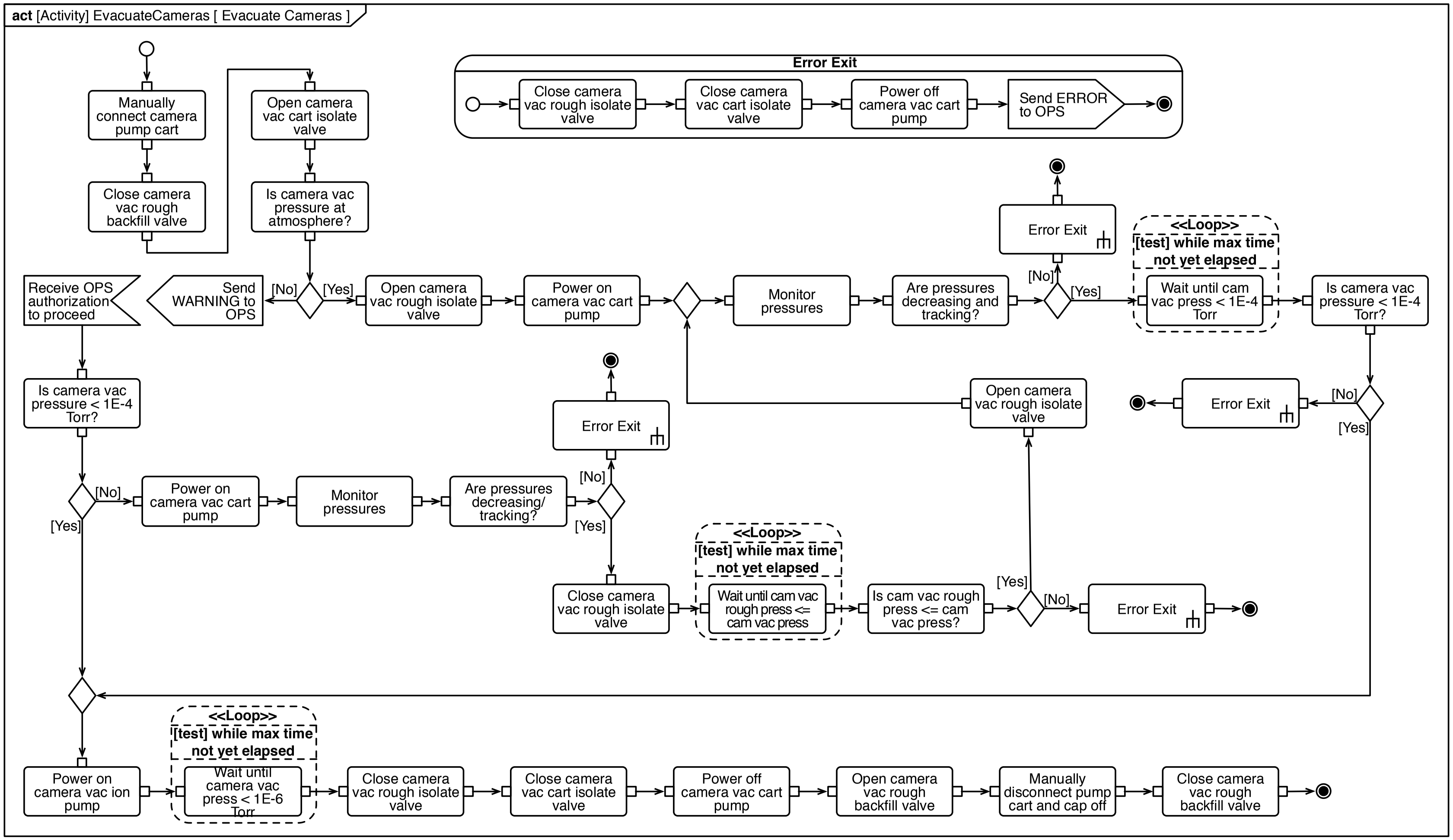

Figure 5. The G-CLEF science camera vacuum control system “Evacuate Cameras” activity.

```xml
<!--  red_vac_ion_pump  -->
▼<parallel id="red_vac_ion_pump:pump">
 ▼<state id="red_vac_ion_pump:status" initial="red_vac_ion_pump:Status_Unknown">
  ▼<onentry>
     <log label="state_entered" expr="'red_vac_ion_pump:status'"/>
   </onentry>
  ▼<!--
     record pump status data, transistion to appropriate state
   -->
  ▼<transition event="red_vac_ion_pump_readout" cond="_event.data === 'On'" type="minimal" target="red_vac_ion_pump:On">
     <log label="red_vac_ion_pump_readout" expr="_event.data"/>
     <assign location="red_vac_ion_pump_readout" expr="_event.data"/>
     <raise event="red_vac:new_pump_data"/>
   </transition>
  ▶<transition event="red_vac_ion_pump_readout" cond="_event.data === 'Off'" type="minimal" target="red_vac_ion_pump:Off">...</transition>
   <!--  unexpected input - go to Unknown, trigger fault  -->
  ▶<transition event="red_vac_ion_pump_readout" type="internal" target="red_vac_ion_pump:Status_Unknown">...</transition>
   <!--  handle commands  -->
  ▼<transition event="process_internal_command" cond="gclef_internal_command_stack.ProcList[0] === 'red_vac_ion_pump:On'">
    ▼<if cond="In('red_vac_ion_pump:On')">
       <!--  no-op, already in this state  -->
       <log label=" already in state" expr="'red_vac_ion_pump:On'"/>
      ▼<script>
         gclef_internal_command_stack.done_proc('red_vac_ion_pump:On')
       </script>
       <else/>
       <log label=" sending red_vac_ion_pump_cmddev" expr="'On'"/>
      ▼<send event="red_vac_ion_pump_cmddev">
         <content>"On"</content>
       </send>
       <!--  move command to Waiting array  -->
      ▼<script>
         gclef_internal_command_stack.sent_proc('red_vac_ion_pump:On')
       </script>
     </if>
   </transition>
  ▶<transition event="process_internal_command" cond="gclef_internal_command_stack.ProcList[0] === 'red_vac_ion_pump:Off'">...</transition>
   <!--  states  -->
  ▶<state id="red_vac_ion_pump:Status_Unknown">...</state>
  ▼<state id="red_vac_ion_pump:On" precondition="red_vac:Pressure_Low_Enough_for_Ion_Pump">
    ▼<onentry>
       <log label="state_entered" expr="'red_vac_ion_pump:On'"/>
       <raise event="gclef_new_data"/>
     </onentry>
   </state>
  ▶<state id="red_vac_ion_pump:Off" precondition="">...</state>
 </state>
 <!--  end red_vac_ion_pump:status  -->
```

Figure 6. The SCXML that implements the action “Power on camera vac ion pump” that is part of the “Evacuate Cameras” activity. For simplicity, fault tolerance and safety actions are not included.

Part of the SCXML state machine definition for one single action ("Power on camera vac ion pump") that is included in this activity is shown in Figure 6. The example SCXML incorporates several different functions. When the state machine is first instantiated, the ion pump status is initialized to "Unknown." If the state machine receives a state transition request with a goal state of "On," then the current state of the ion pump is evaluated. If the state is already "On" then no action is taken. Otherwise the SCXML interpreter verifies that any preconditions are satisfied. In this example, the precondition requires that the "red_vac:ion_pump_operational_zone" state variable is in the state "red_vac:Pressure_Low_Enough_for_Ion_Pump" to ensure that the pressure is in a safe range to power-on the ion pump. If the precondition is met then a command is sent to the simulator component to turn on the ion pump. The success (or otherwise) of this command will be reported through a subsequent readout event from the simulator component (equivalent to a sensor readout on the actual hardware). If the readout event indicates that the pump is now on, then the current value of the state variable will be transitioned to the "On" state.

The *JSSCxml* interpreter can be used to visualize execution of the SCXML state machine in a graphical user interface. Figure 7 displays the visualization for this example once a state in which the ion pump is operating has been achieved. In this figure, the display of most of the hierarchical state machine details have been minimized to reduce clutter. The current value of each state variable is shown in green; other values are shown in white. The interpreter allows the user to trigger specific state transitions and actions manually to allow investigation of the steps required to achieve a specified goal state. This capability is especially useful when triggering simulated faults.

SCXML allows logging of events to a file, and the G-CLEF prototype makes extensive use of this capability to record simulated command execution, sensor readings, and state transitions. The contents of the log file may be plotted for review by the software developers, and analyzed for correct behavior by engineering and science team personnel. An example of a subset of the information recorded for the "Evacuate Cameras" activity is shown in Figure 8.

Developing the prototype state machine hierarchy as part of the G-CLEF preliminary design enhances understanding of the operations of the instrument subsystems and how they interact, in particular under circumstances when the hardware safety system is activated or in the presence of faults when fault tolerant behavior is required. When combined with the software systems engineering process, the executable SCXML facilitates the optimal choice of state machine variables, especially at higher levels of abstraction.

Conceptually, the state machine hierarchy developed as part of the preliminary design, and enhanced with additional detail in later design phases, could be integrated with a SCXML interpreter as part of the actual G-CLEF control system. The advantages of using an actual implementation of the state machines that is validated as part of the design process, rather than developing and verifying a separate, conforming implementation, should be obvious in terms of cost and schedule. Howver, the disadvantages are less clear. The performance of the SCXML interpreter may be a limiting factor under some circumstances, but the G-CLEF hardware, in general, does not require execution of high-speed control loops since response-time and bandwidth requirements are not typically driving factors. Further evaluation of the performance and robustness of the available SCXML interpreters will be conducted as part of the preliminary software design process, to establish whether this is a viable operational approach.

ACKNOWLEDGMENTS

This work has been supported by the GMTO Corporation, a non-profit organization operated on behalf of an international consortium of universities and institutions: Arizona State University, Astronomy Australia Ltd, the Australian National University, the Carnegie Institution for Science, Harvard University, the Korea Astronomy and Space Science Institute, the São Paulo Research Foundation, the Smithsonian Institution, the University of Texas at Austin, Texas A&M University, the University of Arizona, and the University of Chicago.

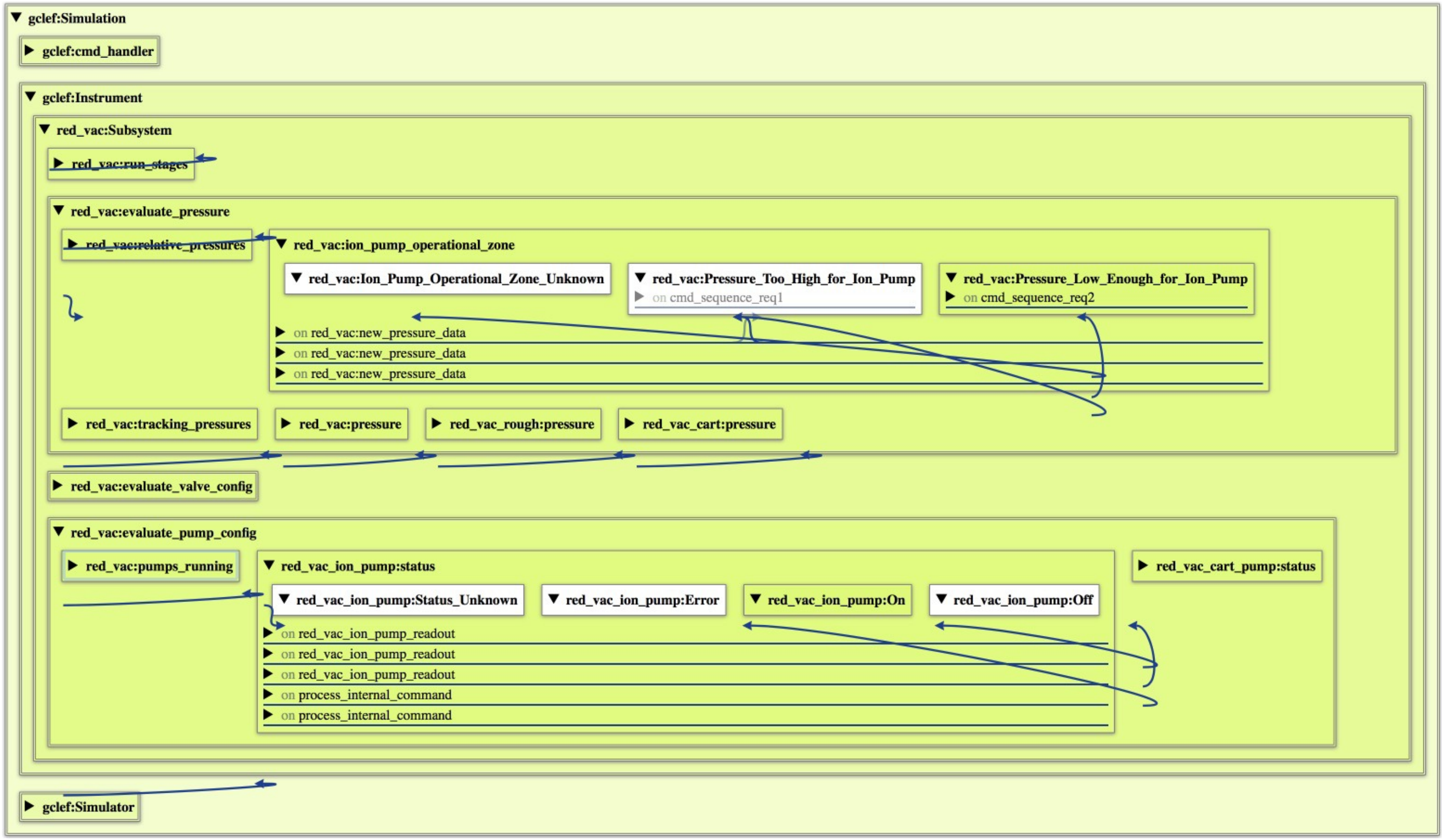

Figure 7. The *JSSCxml* interpreter can be used to visualize execution of the SCXML state machine in a graphical user interface. This example shows the camera vacuum subsystem when it has a achieved a state where the ion pump is operating. To reduce the footprint of this example, most of the hierarchical state machines have been minimized.

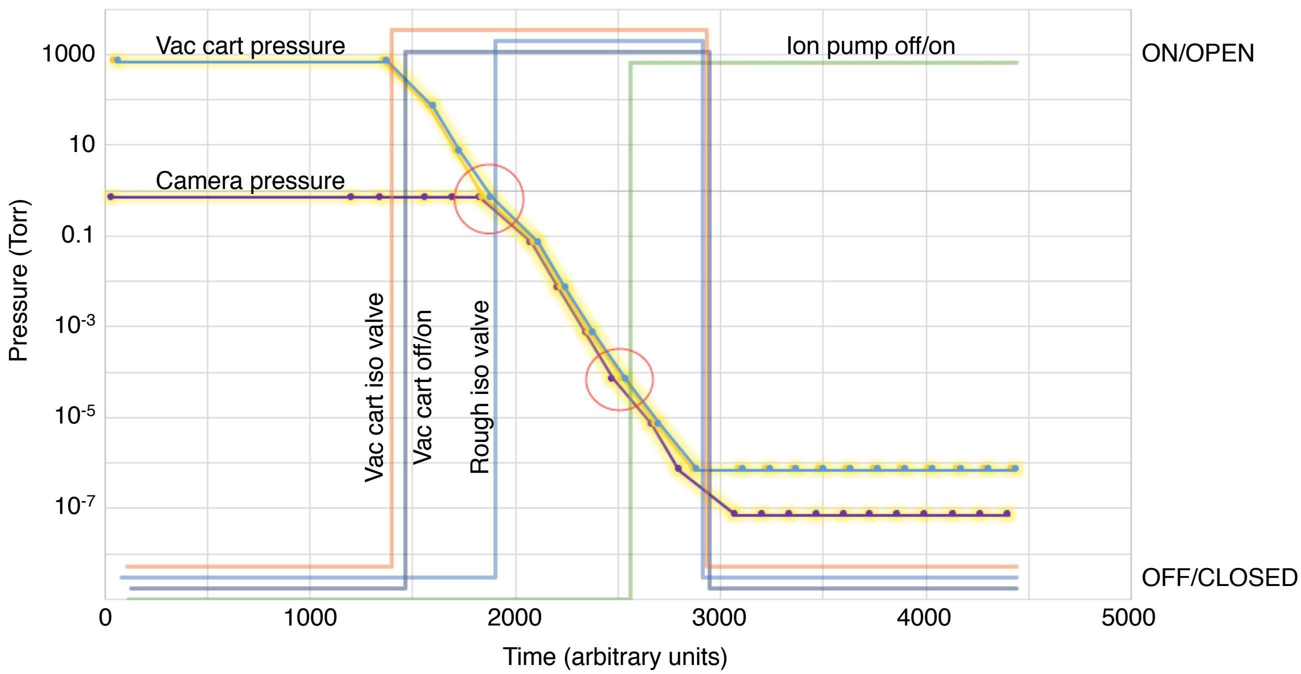

Figure 8. A subset of the information logged by the SCXML state machine prototype for the "Evacuate Cameras" activity. In this example, the camera pressure was manually set to 1/10th atmosphere at the start of the simulation. The individual vacuum pressure samples are indicated by the dots. The individual mechanism traces are shifted vertically slightly for visibility.

REFERENCES

[1] Szentgyorgyi, A. et al., "The GMT-CfA, Carnegie, Catolica, Chicago Large Earth Finder (G-CLEF): a general purpose optical echelle spectrograph for the GMT with precision radial velocity capability," in [*Ground-based and Airborne Instrumentation for Astronomy IV*], McLean, I. S., Ramsay, S. K., and Takami, H., eds., *Proc. SPIE* **8446**, 84461H, SPIE, Bellingham, WA (2012).

[2] Szentgyorgyi, A. et al., "A preliminary design for the GMT-Consortium Large Earth Finder (G-CLEF)," in [*Ground-based and Airborne Instrumentation for Astronomy V*], Ramsay, S. K., McLean, I. S., and Takami, H., eds., *Proc. SPIE* **9147**, 914726, SPIE, Bellingham, WA (2014).

[3] Szentgyorgyi, A. et al., "The GMT-Consortium Large Earth Finder (G-CLEF): An optical echelle spectrograph for the Giant Magellan Telescope (GMT)," in [*Ground-based and Airborne Instrumentation for Astronomy VI*], Evans, C. J.and Simard, L. and Takami, H., eds., *Proc. SPIE* **9908**, 9908–76, SPIE, Bellingham, WA (2016).

[4] Szentgyorgyi, A. et al., "The GMT-consortium large earth finder (G-CLEF): an optical echelle spectrograph for the Giant Magellan Telescope (GMT)," in [*Ground-based and Airborne Instrumentation for Astronomy VII*], Evans, C. J.and Simard, L. and Takami, H., eds., *Proc. SPIE* **10702**, 10702–63, SPIE, Bellingham, WA (2018).

[5] Frebel, A. and Szentgyorgyi, A., "G-CLEF Science Requirements Document," Tech. Rep. SAO-INST-REQ-00006, Rev. A, Smithsonian Astrophysical Observatory, Cambridge, MA (2015).

[6] Szentgyorgyi, A. and Podgorski, W., "G-CLEF System Requirements," Tech. Rep. SAO-INST-REQ-00015, Rev. A, Smithsonian Astrophysical Observatory, Cambridge, MA (2015).

[7] Filgueira, J. M., Bec, M., Soto, J., Liu, N., and Peng, C. Y., "GMT software and controls overview," in [*Software and Cyberinfrastructure for Astronomy II*], Radziwill, N. M. and Chiozzi, G., eds., *Proc. SPIE* **8451**, 11, SPIE, Bellingham, WA (2012).

[8] INCOSE, "Systems Engineering Vision 2020," Tech. Rep. INCOSE-TP-2004-004-02, Version 2.03, INCOSE, San Diego, CA (2007).

[9] International Electrotechnical Commission, "Functional safety of electrical/electronic/programmable electronic safety-related systems," Tech. Rep. IEC 61508-1, Edition 2.0, IEC (2010).

[10] Mueller, M. A. et al., "Precision thermal control of the GMT-consortium large earth finder (G-CLEF)," in [*Ground-based and Airborne Instrumentation for Astronomy VII*], Evans, C. J., Simard, L., and Takami, H., eds., *Proc. SPIE* **10702**, 10702–368, SPIE, Bellingham, WA (2018).

[11] Filgueira, J. M., "GMT Software and Controls Standards," Tech. Rep. GMT-SWC-REF-00029, Version 2, GMTO Corporation, Pasadena, CA (2018).

[12] Evans, I. N., "G-CLEF Instrument Device Control Subsystem Requirements," Tech. Rep. SAO-INST-REQ-00039, Version 2.0, Smithsonian Astrophysical Observatory, Cambridge, MA (2018).

[13] Evans, I. N., Budynkiewicz, J. A., Evans, J. D., Miller, J. B., Onyuksel, C., Paxson, C., and Plummer, D. A., "Software requirements flow-down and preliminary software design for the G-CLEF spectrograph," in [*Software and Cyberinfrastructure for Astronomy IV*], Chiozzi, G. and Guzman, J. C., eds., *Proc. SPIE* **9913**, 99131Q, SPIE, Bellingham, WA (2016).

[14] Plummer, D. A., "G-CLEF Instrument Device Control Subsystem Software Requirements," Tech. Rep. SAO-INST-REQ-00094, Version 2.0, Smithsonian Astrophysical Observatory, Cambridge, MA (2018).

[15] Szentgyorgyi, A., "G-CLEF Operations Concept Document," Tech. Rep. SAO-INST-DOC-00166, Smithsonian Astrophysical Observatory, Cambridge, MA (2018).

[16] Evans, I. N., Evans, J. D., and Budynkiewicz, J. A., "G-CLEF Operations Concept Definition Document," Tech. Rep. SAO-INST-DOC-00065, Version 2.0, Smithsonian Astrophysical Observatory, Cambridge, MA (2018).