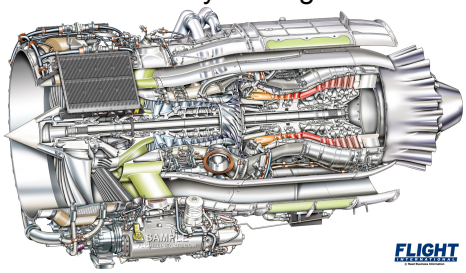
CS486C – Senior Capstone Design in Computer Science

Project Description

Project Title: Next Generation BTU proof of concept	
Sponsor Information: Honeywell	Harlan Mitchell, Systems Technical Manager HTF7K Controls Systems Integration https://aerospace.honeywell.com/en/products/engines/htf7000-turbofan-engine https://en.wikipedia.org/wiki/Honeywell_HTF7000 Honeywell Harlan.mitchell@honeywell.com 602-206-7223

Project Overview:

Honeywell is known for small to mid-sized engines including auxiliary power units (APUs), propulsion engines, and turbo-chargers (note: vehicle turbo-chargers are similar in some ways to gas turbine engines). Honeywell is the largest producer of gas turbine APUs found on many leading aircraft with more than 100,000 APUs produced and more than 36,000 in service today. Within the propulsion engines group Honeywell has applications on helicopters, business jets, turbo props, military jets, and even the US Army Abrams Tank.



Within the Honeywell engines enterprise our group, the Controls Systems Integration (CSI) group, is responsible for all aspects of engine control including the ECU (Engine Control Unit). In addition to the ECU itself the control system is comprised of temperature/pressure sensors, valves, the fuel pump, and other hardware components. The largest current project we are working on is the re-application of the HTF7000 engine on the new Cessna Citation Longitude business jet which will be doing its first flight this month. The Cessna Citation Longitude site is: http://cessna.txtav.com/citation/longitude

As part of engine certification (some engines are certified separately from the aircraft), requirements-based testing is required by the cert authorities (FAA, EASA, etc.); this includes system level requirements. In order to accomplish this verification, the Controls Systems organization employs different methods, one of which is simulator-based target testing, which is a form of "hardware in the loop" testing in which the Honeywell engine controller (the "hardware") is connected to a rack-mounted simulator that emulates the full aircraft/engine system. In this way, the controller essentially thinks it is mounted in an actual aircraft, and can be thoroughly tested in a variety of simulated situations.

A key part of the simulator based testing environment is the BTU (Bench Test Utilities) system which allows Control Systems Engineers to develop and run automated and semi-automated scripts that verify system level requirements. The BTU system is a critical tool because it allows Mechanical Systems Engineers to generate complex Python scripts without knowing Python or having to hand code scripts. The BTU system also keeps track of test/requirement linkage as well as many other functions.

The primary functions of the BTU are to:

- Expedite and standardize the test script generation process.
- Organize, warehouse, and manage test scripts.
- Store test procedure results and system bench configuration data.
- Enable statistical analysis and data mining to report on testing progress.

A typical BTU usage scenario is:

1. The engineer enters test instructions into the BTU using a graphical user interfaces (GUI).
2. The BTU is then used to render these test instructions in a script containing Python® syntax.
3. Using the BTU GUI the script is executed in the test environment.
4. Using BTU functions, the script executing generates tab delimited log files of the test results.
5. Test results are imported back into the BTU where they can be analyzed using GUI tools.
6. In the final step test results are exported to a database where certification results are stored.

While the current BTU meets our basic needs and is being used daily by 10-15 engineers it has grown out of date and needs a major overhaul. The Windows XP/Access 2003 architecture on which the BTU is based can no longer be supported and needs to be completely be re-envisioned.

This project is to establish and validate a modern GUI and architecture for the NG BTU. The primary objective of the project is to demonstrate the GUI and script generation from the integrated data base. Honeywell will provide the BTU code and documentation to the project team. Generated scripts can be compared to existing BTU scripts to verify proper function of the project deliverable therefore no access to the Honeywell test environment will be necessary. While many improvements and extended functionality could be included in the project the primary objective is to focus on the architecture proof of concept.

Knowledge, skills, and expertise required for this project:

As with all projects, the team will be expected to learn the knowledge and skills required for this project early on. Beyond standard senior-level capabilities in programming and software design, helpful skills will include:

- Familiarity with different COTS database systems.
- Experience with Python.
- Skills in interface design, testing and refinement.

Equipment Requirements:

No special equipment should be required beyond a standard development platform (your laptop), as well as freely available environments and software tools.

Software and other Deliverables:

1. Software implementing the functionalities outlined above.
2. Professionally documented codebase.
3. Test results document demonstrating Python script compatibility with current system test environment.
4. A strong as-built document that details the design and implementation of the software. This must be robust enough to allow a future development team to easily pick up where you left off.