Relevant Milestone: Demonstrate scalable analytic verification technology on a major subsystem for Aerospace avionics.

Shown: The application of model checking to the DEOS real-time embedded aerospace operating system from Honeywell to discover a subtle error not uncovered using the testing techniques required for FAA certification. This impact of this error during flight could have been starvation of critical real-time flight calculations. Indicate the scaling of model checking by showing the average factor of increase in lines of code (yellow) and state-space handled (white) by each technique developed and, in the middle, a graph indicating the impact of these techniques with respect to the time taken to analyze a 1000 lines of code.

Accomplishment / Relation to Milestone and ETG: Development of the Java PathFinder model checker, with accompanying set of synergistic verification technologies (including, abstractions, slicing, partial-order reduction, intelligent search and environment generation techniques) to enable the efficient analysis of object-oriented, concurrent programs such as those found in the next generation of avionics systems (e.g. the DEOS O/S for Integrated Modular Avionic systems). These model checking technologies have significantly reduced the effort required to analyze avionics software: currently we analyze 1000 lines of code per day compared to state of practice of 50 LOC/day in 1998.

Future Plans: Develop techniques to allow guarantees for correct behavior under certain assumptions that can be checked during actual execution using run-time program monitoring. Also, development of “learning” algorithms whereby the model checker’s search strategy can be adapted according to the structure of the program being analyzed.

ETG: Provide increased confidence and lower the cost of development of next generation avionics software.
Strategic Investments Research Program
High-Assurance Software Design

**Strategic Investments Research Program**

**High-Assurance Software Design**

---

**Case 0:**

```
new();
```

**Case 1:**

```
Stop();
```

**Case 2:**

```
Remove();
```

**Case 3:**

```
Wait();
```

---

**Slicing**

**Abstraction**

**Environment Generation**

---

**Repair**

Combined techniques allows

\[ O(10^2) \] source line and

\[ O(10^6) \] state-space increase over state of practice

---

**DEOS**

10000 lines to 1500

**3x Slicing**

Property preserving

---

**Case 0:**

```
new();
```

**Case 2:**

```
Remove();
```

---

**JPF Model Checker**

- **Infinite state to 1,000,000 states**
- **Semi-automated and requires domain knowledge**

---

**KLOC/day**

- **1997: 0.00**
- **1998: 0.03**
- **2000: 0.33**
- **2001: 1.00**

---

**Bandera code-level debugging of error-path**

- **Spurious error elimination during abstraction**
- **2x Heuristic search**
- **10x Focused search for errors**
- **State compression**
- **2x 15x**

---

**Combined techniques allows**

- **5x Abstraction DEOS**
- **100x Environment Generation**

**Partial-order reduction**

- **2x 10x**

---

**Case 0:**

```
new();
```

**Case 2:**

```
Remove();
```