

# Studentisches Projekt

## ACOP – Aircraft Operating System

Jan Peleska und Kirsten Berkenkötter  
AG Betriebssysteme/Verteilte Systeme

{jp,kirsten}@tzi.de  
11. Mai 2005

### 1 Projektbeschreibung

In der Norm [1] wird das *Application Program Interface (API)* für ein Betriebssystem beschrieben, welches besonders für den Einsatz in verteilten Steuerungssystemen der Luftfahrt geeignet ist. Diese Norm ist in den vergangenen Jahren von einigen Herstellern umgesetzt worden, so dass erstmals im neuen Airbus A380 Controller mit einem solchen Betriebssystem ausgestattet sind.

Abgesehen von der Neuheit und Aktualität hat das API mehrere interessante Eigenschaften, die es für den Einsatz bei verteilten Embedded Systems besonders geeignet machen und von konventionellen (Echtzeit-)Betriebssystemen zumindest in der Gesamtheit nicht unterstützt werden:

- Die API-Spezifikation fordert *Partitioning*, das heisst, dass sicherheitskritische Applikationen nicht durch andere, weniger kritische, in ihrem Ablauf gestört werden dürfen, auch wenn sie auf der selben CPU aktiv sind.
- Mit *Blackboards* und *Ports* wird ein vereinheitlichtes Kommunikationskonzept definiert, welches sowohl die (verteilte) Interprozesskommunikation als auch den Datenaustausch zwischen Controller und Peripherie unterstützt. Besonders interessant ist hierbei der Ansatz, eine Schnittstellenabstraktion anzubieten, welche nahezu vollständig vom unterliegenden Device abstrahiert: Daten werden mit Hilfe von *Functional Groups*, *Data Sets* und *Signals* identifiziert; die Umsetzung der Signaldaten auf konkrete Busse, digitale oder analoge Schnittstellen erfolgt transparent durch unter dem API angesiedelte Dienste.

- Das Gesamtsystem ist vor der Inbetriebnahme so konfigurierbar, dass die Verfügbarkeit von Ressourcen (CPU-Zuteilung, Kommunikationsbandbreite) im Umfang der Konfiguration garantiert und nicht vom zufälligen Aufeinandertreffen mehrerer Ereignisse ungünstig beeinflusst werden kann. Dies wird unter anderem auch durch die Normung eines verteilten Kommunikationsdienstes unterstützt [2], das *Avionics Full Duplex Network*, welches ebenfalls im Airbus A380 erstmalig zu Einsatz kommt.

Trotz der attraktiven Merkmale des Betriebssystem-APIs gibt es bisher nur Implementierungen für Spezialcontroller der Luftfahrt; eine auf dem PC nutzbare Umsetzung fehlt noch. Die soll Gegenstand des hier beschriebenen Studentischen Projektes *ACOP – Aircraft Operating System* sein.

Das Gesamtprojekt wird in drei Teilaufgaben gegliedert:

- **Implementierung des ARINC 653-APIs** als *zusätzliches, parallel nutzbares API* im Linux Kernel: Die Funktionen des APIs werden als neue Linux-Systemaufrufe registriert und durch Erweiterungen des Kernels realisiert. Die Schnittstellenabstraktion wird für Ethernet, Digitale I/O, Analoginterfaces, CAN, ARINC 429 und AFDX realisiert. Dies sind die wichtigsten heute in der Luftfahrt gebräuchlichen Schnittstellen.
- **Formale Spezifikation der Wirkungsweise aller API-Funktionen:** Um den informalen, im Manual-Page Stil abzufassenden API-Beschreibungen ein eindeutiges Verhaltensmodell zuzuordnen, werden formale Spezifikationen mit Hilfe von UML2.0, Z und/oder Timed CSP erstellt. Diese sollen als eindeutige Referenzspezifikation und gleichzeitig als Input für die Testfallerstellung dienen.
- **Hardware/Software-Integrationstest der Implementierung:** Für den Test der API-Implementierung werden sog. *Testapplikationen* entwickelt. Diese laufen auf dem zu testenden Rechner und werden im *Hardware-in-the-Loop Test* durch einen separaten Rechner – die *Test Engine* – kommandiert. Die Testapplikation meldet die lokale Wirkung der Systemaufrufe an die Test-Engine zurück; letztere prüft zusätzlich noch die Ausgaben des Testlings auf den Hardwareinterfaces auf Korrektheit.

## 2 Wer kann teilnehmen?

Das Projekt wird für Diplom-, Master-<sup>1</sup> oder Bachelor-Studenten angeboten. Für den Diplomstudiengang ist die Teilnahme an allen vier Semestern erforderlich. Für Bachelor-Studierende sind nur die Projektsemester 1+2 erforderlich, dabei werden nur die Themenstellungen API-Implementierung und API-Spezifikation behandelt, denn die Testaktivitäten werden vornehmlich in den Projektsemestern 3+4 abgewickelt.

## 3 Begleitende Lehrveranstaltungen

Die Projektmitglieder müssen parallel zum Projekt folgende Lehrveranstaltungen besuchen:

**WS2005/2006** 1. Betriebssysteme 1, Vorlesung und Übung (2+2 SWS)

2. Betriebssysteme 2, Vorlesung und Übung (2+2 SWS)

**SoSe2006** Spezifikation eingebetteter Systeme, Vorlesung und Übung (2+2 SWS)

**WS2006/2007** Testautomatisierung 1, Vorlesung und Übung (2+2 SWS)

**SoSe2007** Testautomatisierung 2, Vorlesung und Übung (2+2 SWS)

*Bitte beachten:* Die Teilnahme an den Lehrveranstaltungen ist Voraussetzung für den Erwerb des Projektscheins. Die Noten für Scheine/Modulprüfungen, welche in den Lehrveranstaltungen erzielt wurden, fließen allerdings nicht in die Projektnote ein.

## 4 Scheinkriterien

Der Projektschein und seine Note setzen sich aus folgenden Punkten zusammen:

- Teilnahme an o. g. Lehrveranstaltungen (Pflicht, aber kein Einfluss auf die Note)

---

<sup>1</sup> Gibt es solche?

- Teilnahme am wöchentlichen Projektplenum (Pflicht, aber kein Einfluss auf die Note)
- Beitrag zum Projekt-, System-, Konfigurationsmanagement und zum Internet-Auftritt des Projekts (Pflicht, aber kein Einfluss auf die Note)
- Umfang und Qualität der erstellten
  - Spezifikationen
  - Software zur Implementierung des ACOP-APIs
  - Tests zur Prüfung der Implementierung
- Umfang und Qualität des Beitrags zum Projektbericht
- Qualität der Präsentation auf dem Projekttag.

## Literatur

1. ARINC 653 *AVIONICS APPLICATION SOFTWARE STANDARD INTERFACE*. AIRLINES ELECTRONIC ENGINEERING COMMITTEE, OCTOBER 16, 2003.
2. ARINC 651 *Avonics Full Duplex Network*.