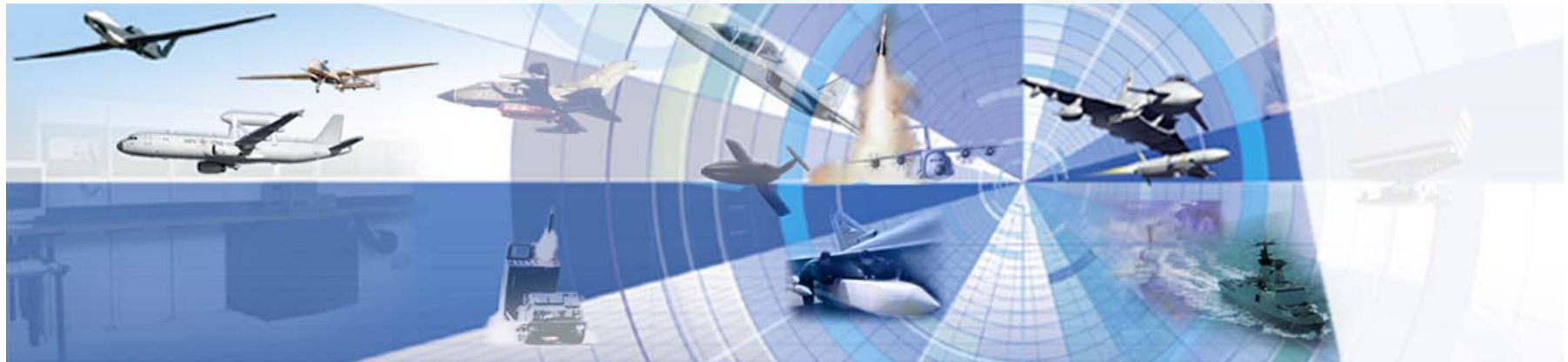


Modellbasierte Entwicklung eingebetteter Systeme



Innovation Forum Embedded Systems, 23. April 2010

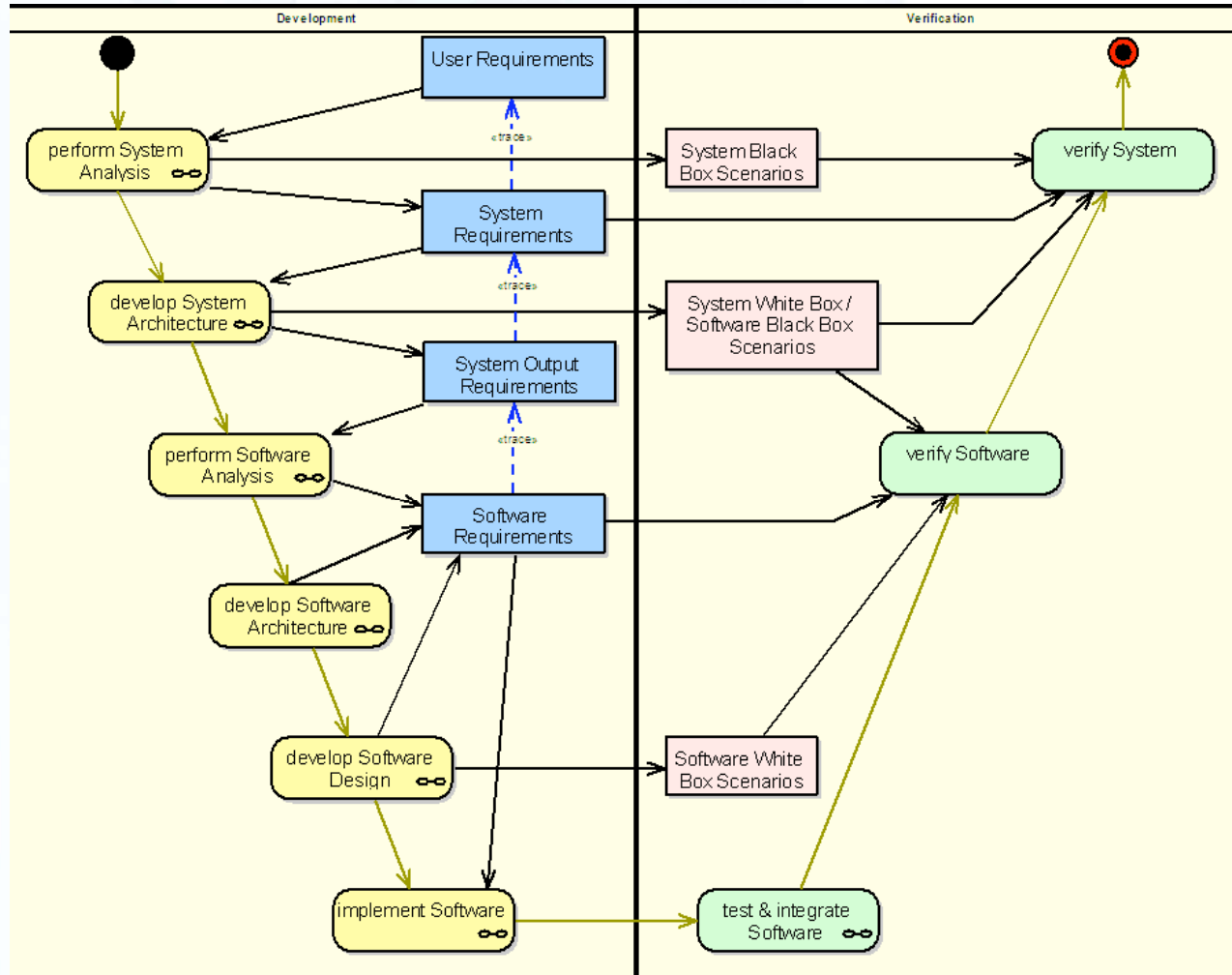
Gundula Blohm

Randbedingungen

- Eingebettete Systeme zeichnen sich durch starke Abhängigkeiten zwischen Software- und Hardware-Anteilen aus.
 - Dem ist bereits in der Systementwurfsphase Rechnung zu tragen.
- Der Übergang in die Software- und Hardware-Entwicklung ein kritischer Punkt, da hier ein Wechsel der Disziplin, Bearbeiter und oft auch Abteilung stattfindet.
- Bei sicherheitskritischen Systemen ist es zusätzlich notwendig, die Anforderungen von der Systemebene bis zu ihrer Realisierung verfolgen zu können.
 - Die Entwicklungsmethodik muss die Zusammenarbeit unterstützen.

Vorgehensweise, RBE, MBE

- RBE: Fokus liegt auf den Anforderungen
- 👉 Unterstützung formaler Nachweisführung (Tracing, Justification)
- 👉 Zusammenhänge oft nicht einfach zu erkennen
- Zusätzliche Beschreibungen notwendig (bisher oft Text-Dokumente)
- Während bei der Software-Entwicklung modellbasierte Vorgehensweisen bereits state-of-the-art sind, stecken diese bei eingebetteten Systemen in der Praxis in den Kinderschuhen.



UML (et al)

- Modellierungssprachen (z.B. UML, SysML, ...) bieten eine Syntax
 - 👍 einheitliche Notation => keine „Bearbeiter“-spezifische Legende zu Diagrammen notwendig
 - 👎 Vorgehensweise und Semantik fehlen
- Modellierungs-Neulinge sind mit der Vielzahl der angebotenen Diagramme, Elemente und Relationen oft überfordert
- Für verschiedene Domänen sind unterschiedliche Beschreibungssprachen optimiert

Der Weg zum Ziel

- Definition der Vorgehensweise und Modellierungssprache
 - In welcher Disziplin entstehen welche Diagrammen, Elemente, Relationen
 - Begrenzung auf disziplin-notwendige Elemente und Relationen
 - Hinterlegen der Syntax mit Semantik (Stereotypen)
 - Schrittweise Verfeinerung, um Bruch zwischen Disziplinen zu vermeiden

=> Durchgängigkeit
- Anpassung der Werkzeuge
 - Template für Modellstruktur
 - Reduktion auf das Wesentliche: welche Diagramme, Elemente und Relationen werden angeboten
- Verfeinerung dedizierter (UML)-Modellelemente durch andere Modellierungssprachen (z.B. Matlab)

Grenzen und Ausblick

- Für wichtige Elemente gibt es oft mehrere Verfeinerungen, dies wird von Werkzeugen nicht unterstützt.
Beispiel: externe Sicht auf einen Use Case mittels Sequenzdiagramm, interne Verfeinerung mittels Aktivitätsdiagramm
- Interfaces müssen bei Verfeinerungen oft manuell nachgezogen werden.
Beispiel: Verfeinerung von Sequenzdiagrammen
- Für System-Schnittstellenbeschreibungen (logisch und physikalisch) stehen keine geeigneten Standardelemente z.V.
Workaround: Detail-Beschreibung in XML / Excel
- Um das dynamische Verhalten von Modellen zu simulieren, muss viel Zusatzaufwand betrieben werden.
Beispiel: manuelle Erstellung von Statecharts aus Sequenzdiagrammen
- Werkzeug-Anpassungen sind oft aufwendig, aber notwendig für Akzeptanz und effizientes Arbeiten.
Workaround: Ein Mitarbeiter (Vollzeit-Job) für Werkzeug-Anpassung und Support/Coaching der Entwickler

Ausblick



EADS-DE

System Development Improvement Workgroup #5: MBSE

EU-Forschungsprojekte

Model-based methods and tools for Avionics and surveillance
embeddeD SystEmS: <http://www.mades-project.org>



Java Environment for Parallel Realtime Development:
Entwicklung eingebetteter Systeme mit Mikroprozessoren und FPGAs