

Modellierung von Echtzeitsystemen

Reaktive Systeme

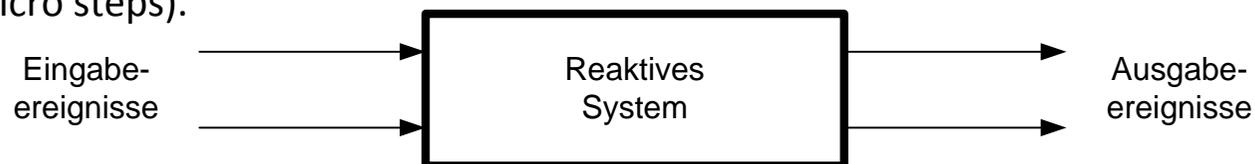
Werkzeuge: SCADE, Esterel Studio

Esterel

- Esterel ist im klassischen Sinne eher eine Programmiersprache, als eine Modellierungssprache
- Esterel wurde von Jean-Paul Marmorat und Jean-Paul Rigault entwickelt um die Anforderungen von Echtzeitsystemen gezielt zu unterstützen:
 - direkte Möglichkeit zum Umgang mit Zeit
 - Parallelismus direkt in der Programmiersprache
- G. Berry entwickelt die formale Semantik für Esterel
- Es existieren Codegeneratoren zur Generierung von u.a. **sequentiellen** C, C++ Code:
 - In Esterel werden (parallele) Programme in **einen** endlichen Automaten umgewandelt
 - Aus dem endlichen Automaten wird ein Programm mit **einem** Berechnungsstrang erzeugt ⇒ deterministische Ausführung trotz paralleler Modellierung.
- SCADE (ein kommerzielles Tool, das u.a. die Esterel-Sprache verwendet) wurde bei der Entwicklung von Komponenten für den Airbus A380 eingesetzt.
- Ein frei verfügbarer Esterel-Compiler kann unter <http://www-sop.inria.fr/esterel.org/files/> bezogen werden (siehe Links auf der Vorlesungs-Homepage).

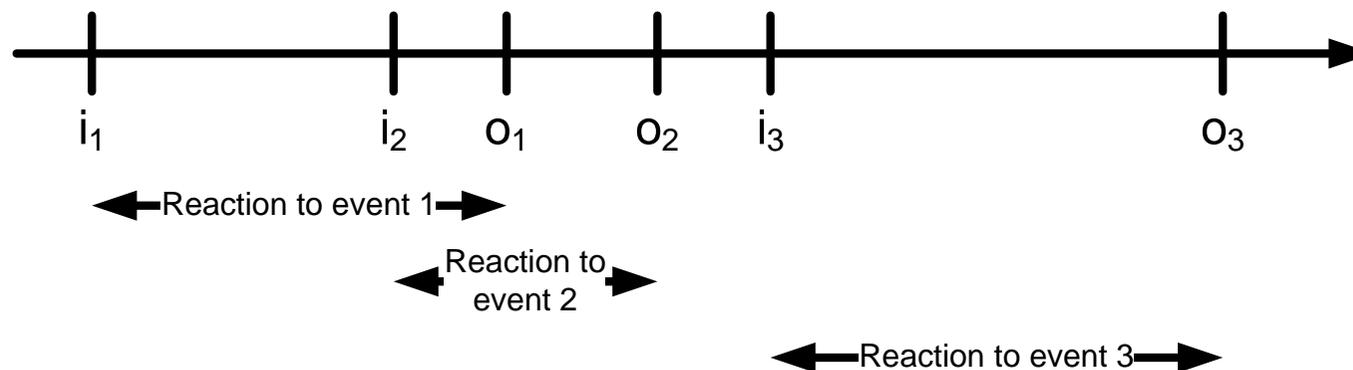
Einführung in Esterel

- Esterel gehört zu der Familie der **synchronen** Sprachen. Dies sind Programmiersprachen, die optimiert sind, um **reaktive Systeme** zu programmieren. Weitere Vertreter: Lustre, Signal, Statecharts
- Bei **reaktiven Systemen** erfolgen Reaktionen direkt auf **Eingabeereignisse**
- Synchroner Sprachen zeichnen sich vor allem dadurch aus, dass
 - Interaktionen (Reaktionen) des Systems mit der Umgebung die Basisschritte des Systems darstellen (**reaktives System**).
 - Anstelle von physikalischer Zeit die **logische Zeit** (die Anzahl der Interaktionen) verwendet wird.
 - Interaktionen, oft auch **macro steps** genannt, bestehen aus einzelnen Teilschritten (micro steps).



Reaktive Systeme

- In reaktiven Systemen (reactive / reflex systems) werden für Eingabeereignisse Ausgaben unter Einhaltung zeitlicher Rahmenbedingungen erzeugt.
- Reaktive Systeme finden u.a. Anwendung in der Industrie zur Prozesssteuerung und zur Steuerung / Regelung in Automobilen und Flugzeugen.
- Schwerpunkte bei der Umsetzung von reaktiven Systemen sind Sicherheit und Determinismus.
- Bearbeitung der Ereignisse kann sich überlappen (i input, o output)



Synchronitätshypothese

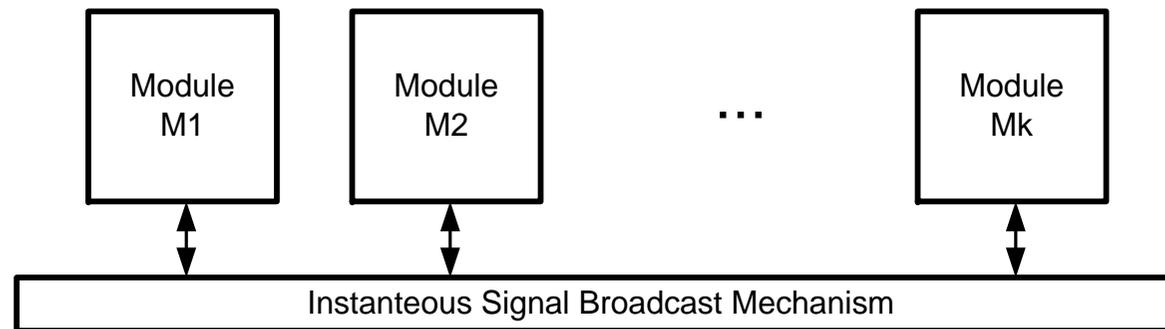
- Die Synchronitätshypothese (synchrony hypothesis) nimmt an, dass die zugrunde liegende physikalische Maschine des Systems unendlich schnell ist.
 - Die Reaktion des Systems auf ein Eingabeereignis erfolgt augenblicklich (ohne erkennbare Zeitverzögerung). Reaktionsintervalle reduzieren sich zu Reaktionsmomenten (reaction instants).
- **Rechtfertigung:** Diese Annahme ist korrekt, wenn die Wahrscheinlichkeit des Eintreffens eines zweiten Ereignisses, während der initialen Reaktion auf das vorangegangene Ereignis, sehr klein ist.
- Esterel erlaubt das gleichzeitige Auftreten von mehreren Eingabeereignissen. Die Reaktion ist in Esterel dann vollständig, wenn das System auf alle Ereignisse reagiert hat.

Determinismus

- Esterel ist deterministisch: auf eine Sequenz von Ereignissen (auch gleichzeitigen) muss immer dieselbe Sequenz von Ausgabe Ereignissen folgen.
- Alle Esterel-Anweisungen und -Konstrukte sind garantiert deterministisch. Die Forderung nach Determinismus wird durch den Esterel Compiler überprüft.
- Durch den Determinismus wird die Verifikation von Anwendungen wesentlich vereinfacht, allerdings birgt er auch die Gefahr, dass Ereignisse „vergessen“ werden, falls sie exakt zeitgleich mit höher priorisierten Ereignissen eintreffen.

Grundlagen der Esterel Sprache: Signale / Sensoren

- Die Kommunikation im System erfolgt durch Signale und Sensoren
 - Sensoren stellen Messwerte zur Verfügung; Es muss immer ein Wert anliegen.
 - Signale können zur Ein- und zur Ausgabe verwendet werden. Es muss nicht immer ein Wert anliegen.
- Man unterscheidet dabei zwei Arten von Signalen:
 - Wertbehaftete Signale (Signal liegt mit einem Wert an oder liegt nicht an)
 - *pure* Signale (Signal liegt an oder nicht)
- Esterel Programme können in mehrere Module aufgeteilt werden
- Kommunikation der Module erfolgt über einen Broadcast-Mechanismus



Grundlagen der Esterel Sprache: Programmsteuerung

- Das Verhalten des Programms wird durch Signale beeinflusst:
 - *emit*: Ein Signal wird ausgesandt.
 - *await*: Auf das Auftreten eines Signals wird gewartet.
 - *present*: Prüft, ob ein Signal anliegt.
- Die Ausführung eines Moduls kann mittels `abort` abgebrochen werden:
 - Syntax: `abort` Body `when` Exit_Condition
- Ein periodisches Ausführen von Anweisungen kann mit `every` umgesetzt werden.
 - Syntax: `every` Occurrence `do` Body `end every`
- Komposition von Anweisungen:
 - Blöcke von Befehlen (Module) können **nacheinander** oder **gleichzeitig** ausgeführt werden.
 - Blöcke von Befehlen (Module) können **wiederholt** ausgeführt werden.
 - Blöcke von Befehlen (Module) können **unterbrochen** werden.