



Versuch:

SDL 1 - Grundlagen



Studiengänge:

Ausbildungsziel:

Ausbildungsinhalte:

Dokumente:

Gerätetechnik:

Vorkenntnisse:

- Informationstechnik, Multimediatechnik, Elektrotechnik
- Erlangung von praktischen Fertigkeiten bei der Spezifikation und Beschreibung von Software mittels SDL - Specification and Description Language - einschließlich der Simulation.
- Sichere Anwendung der SDL-Grundkonzepte:
 - Architektur,
 - Kommunikation,
 - Beschreibung des dynamischen Prozessverhaltens,
 - Abstract Data Types - ADT.
- Arbeit mit einem professionellen SDL-Tool.
- Kennen lernen der integrierten Entwicklungsumgebung (IDE) "SDL- und TTCN-Suite" von Telelogic bezüglich der Arbeitsschritte:
 - Anlegen von Projekten, Umgebungsvariable der IDE (Verzeichnisse, Parameter, ...) einstellen
 - Editieren typischer SDL-Systeme mit Syntaxüberprüfung und Fehlerbeseitigung.
 - Simulation der Systeme (Umgang mit dem Simulator, Text- und MSC-Darstellung der Simulation).
- Ausgabe einer Spezifikation in SDL-PR und deren Interpretation.
- Vorlesung [SDL](#)
- SDL- und TTCN-Suite von Telelogic. (Diese Software ist auf allen Rechnern in 6-401 installiert.)
- Kenntnisse über die SDL Grundkonzepte:
 - Architektur,
 - Kommunikation,
 - Beschreibung des dynamischen Prozessverhaltens,
 - Abstract Data Types - ADT.

Versuchsvorbereitung

- Verschaffen Sie sich einen allgemeinen Überblick zu Anwendungsgebieten und wesentlichen Merkmalen der Sprache anhand Vorlesung SDL (PDF-Format: 0,2 MB)
- Erarbeiten Sie sich die SDL-Grundkonzepte:
 - Architektur: Environment, System, Block, Process, Procedure.
 - Kommunikation: Signal, Signallist, Signalroute, Signalpath.
 - Prozessbeschreibung: Start, Task, State, Input, Save, Decision, Output, Create, usw.
 - Datendeklaration:
 - Umgang mit vordefinierten Typen: Boolean, Integer, Real, Nature, Character, Charstring, Pid, Time usw.
 - Der Deklaration neuer Typen mittel NEWTYPE, der Ableitung von Typen mittels SYNTYPE,
 - Der Deklaration komplexer Typen mittel ARRAY und STRUCT einschließlich der Verwendung.
- Überlegen Sie sich ein eigenes Beispiel aus der Kommunikations- oder Steuerungstechnik, um dieses mit SDL zu beschreiben, syntaktisch zu prüfen und simulieren zu können.

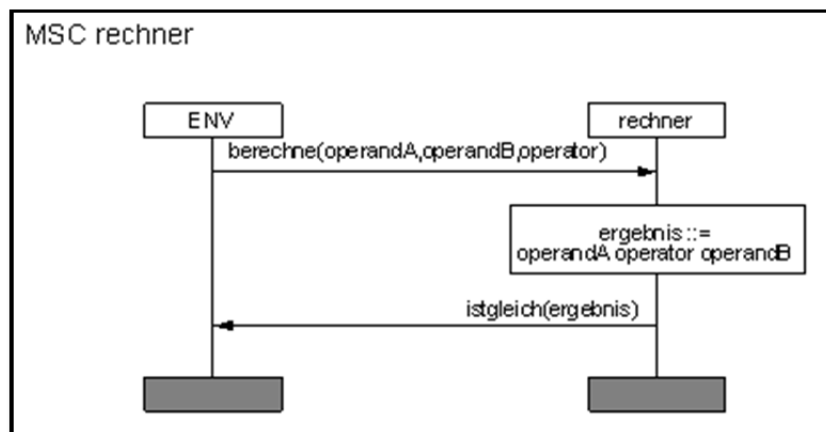
Allgemeiner Hinweis

- Stellen Sie nach dem Öffnen der Suite die Arbeitsverzeichnisse so ein, das diese auf einen Ordner "...\\SDL" im Laufwerk "I:" verweisen.
- Legen Sie für jedes Projekt im Ordner ...\\SDL einen neuen Ordner an.

Aufgabe 1

Entwerfen Sie ein System „rechner“ und simulieren Sie es. Das System habe folgende Eigenschaften:

- Das System rechner besteht aus einem Block rechner. Der Block rechner aus einem Process rechner. Das System rechner kommuniziert mit ENV über 2 Signale:
 - von ENV zu System: `berechne(operandA,operandB,operator)`,
 - von System zu ENV: `istgleich(ergebnis)`.
- Die beiden Signale werden wie folgt deklariert:
SIGNAL `berechne(Real,Real,Character)`, `istgleich(Real)`.



- Sehen Sie die Operationen: Addition '+', Subtraktion '-', Multiplikation '*', Division '/' vor.

Simulieren Sie das System!

- Starten Sie die Simulation über den Menü-Button „Simulate“!
- Führen Sie folgende Simulationsvarianten durch:
 - (1) Simulation über Kommandozeile,
 - (2) Simulation mittels vorhandener Buttons,
 - (3) Simulation mittels eigener Buttons

(1) Simulation über Kommandozeile

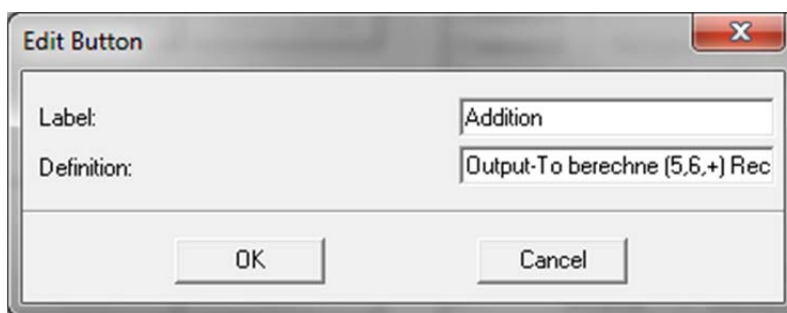
- Starten Sie den MSC-Editor und Zeichnen Sie die Simulation mit dem Tracewert 2 auf! Geben Sie hierfür folgenden Befehl Eingabefeld „Command“ an!
 - Start-Interactive-MSC-Log 2
- Senden Sie für jede Rechenoperation das Signal „berechne“ mit beliebigen Werten an den Block „Rechner“!
Achtung: Nach jeder Eingabe eines Befehls, den Befehl „Go“ ausführen, damit der Befehl verarbeitet wird
Beobachten Sie den Verlauf im dargestellten MSC!
 - Z.B. Output-To berechne (5,3,'+') rechner

(2) Simulation mittels vorhandener Buttons

- Nutzen Sie die Buttons der Simulationsoberfläche, um Signale an das System zu senden!
- Senden Sie für jede Rechenoperation mittels des Buttons „SendTo“ im Modul „Send Signal“ das Signal „berechne“ an den Block „Rechner“! Beobachten Sie den Verlauf im dargestellten MSC!

(3) Simulation mittels eigener Buttons

- Die Simulationsoberfläche bietet die Möglichkeit, eigene Buttons zum Senden von Signalen zu definieren.
- Zum Erstellen eigener Buttons zuerst über die Option „Add Group“ im Menü „Buttons“ eine neue Gruppe anlegen. Anschließend im Dialog einen Namen für das neue Modul eingeben, z.B. Rechner. Jetzt können eigene Buttons in diesem Modul erstellt werden. Dazu auf den Button „Group“ klicken und dann „Add“ wählen. Im Dialogfenster bei Label den Namen des Buttons eingeben (z.B. Addition) und bei Definition den Befehl, der ausgeführt werden soll. z.B.



- Erzeugen Sie für jede Rechenoperation einen eigenen Button!
- Stellen Sie den Tracewert für die Simulation auf 6! (Aufzeichnung aller SDL-Aktionen, Resultate und Parameter)
 - Set-Trace System Rechner 6
- Starten Sie einen GR-Trace!

Durch den GR-Trace wird die Reihenfolge der Abarbeitung, der einzelnen Schritte, im SDLDiagramm anschaulicher dargestellt. Die Symbole, die bei Abarbeitung eines Befehls durchlaufen werden, werden gelb hinterlegt. Der Tracewert 2 bedeutet, es werden alle benutzten Symbole im SDL-Editor angezeigt.

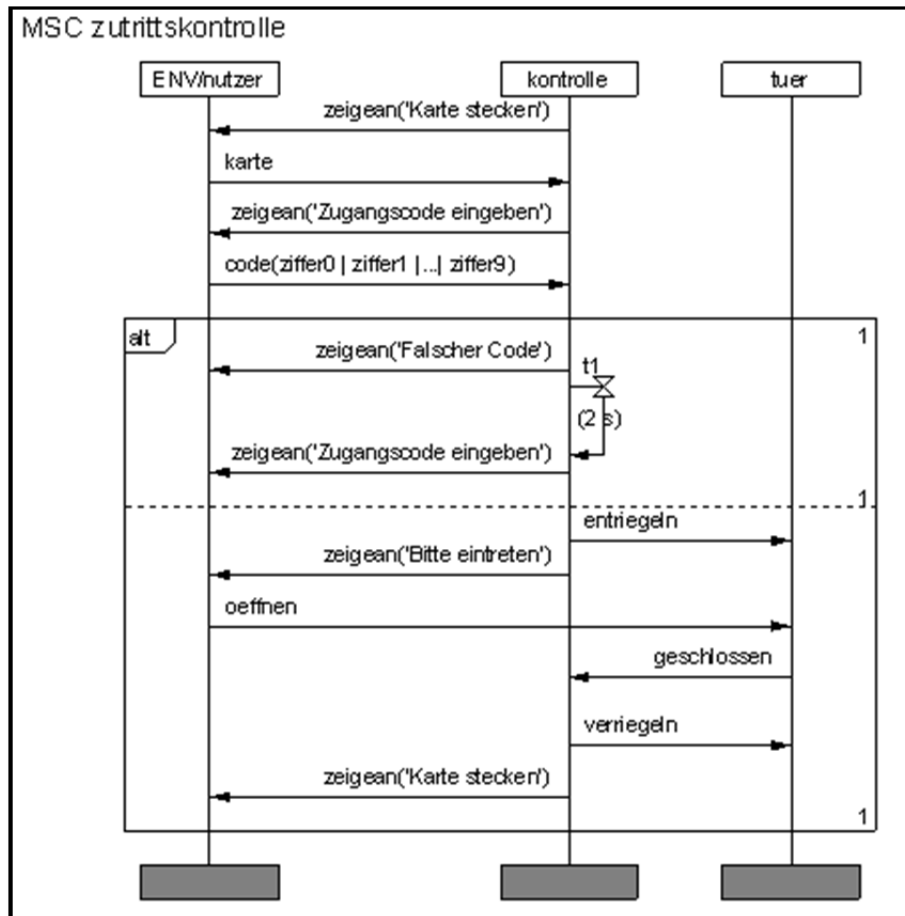
 - Set-GR-Trace System Rechner 2
- Starten Sie die Simulation und beobachten Sie den SDL-Editor!

Tipp: Die gelb hinterlegten Symbole können über View→Temporary Colors→Remove wieder entfernt werden.

Aufgabe 2

Entwerfen und simulieren Sie ein SDL-System „zutrittskontrolle“ mit folgenden Eigenschaften:

- System „zutrittskontrolle“ besteht aus den Blöcken „kontrolle“ und „tuer“.
- In jedem der Blöcke existiert ein gleichnamiger Process.
- Realisieren Sie das im MSC beschriebene Verhalten.
- Simulieren Sie das System.



Simulieren Sie das System mittels MSC- und GR-Trace!

- Führen Sie folgende Simulationsvarianten durch:
 - Simulation mittels vorhandener Buttons,
 - Simulation mittels eigener Buttons
 - Legen Sie hierfür die 4 Buttons „Karte“, „Code NotOK“, „Code OK“ und „Oeffnen“ an!
 - Speichern Sie die Buttonkonfiguration über Buttons→Save As... in Ihrem Verzeichnis!
- Erweitern Sie das System wie folgt: trifft 5 s nach dem Signal „entriegeln“ kein Signal „geschlossen“ ein, wird die Tür verriegelt.