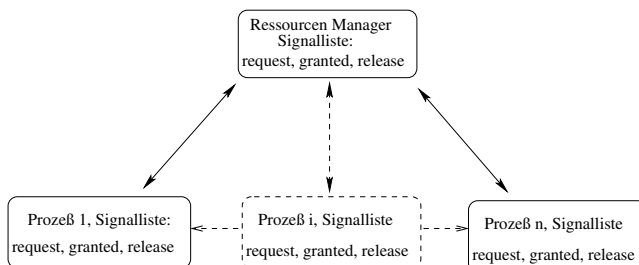


Übung 6: SDL & MSC

Besorgen Sie sich bitte unter www.cinderella.dk eine Testversion der SDL Cinderella Toolsuite, es gibt eine 14-Tages Demo. Mit Cinderella koennen Sie die Systeme aus den folgenden Aufgaben spezifizieren und simulieren.

Aufgabe 1: Mutual exclusion

Es soll ein System modelliert werden, in dem n Prozesse gemeinsam auf eine exklusive Ressource zugreifen wollen, d.h. die Prozesse müssen sich wechselseitig vom zeitgleichen Zugriff auf diese Ressource ausschließen können. Zur Vereinfachung können Sie dabei einen Ressourcen-Manager einsetzen. Dieser Manager teilt dem Prozess die Ressource zu (via der Nachricht granted), der sich als erster um einen Zugriff (via Nachricht request) bemüht hat. Die belegte Ressource wird dann von diesem Prozess nach einer fixen Zeitdauer wieder freigegeben (via Nachricht release). Die Kommunikationsstruktur ist zur Verdeutlichung unten noch einmal graphisch dargestellt. Modellieren Sie dieses System als SDL-System (Blockdiagramm + State charts).



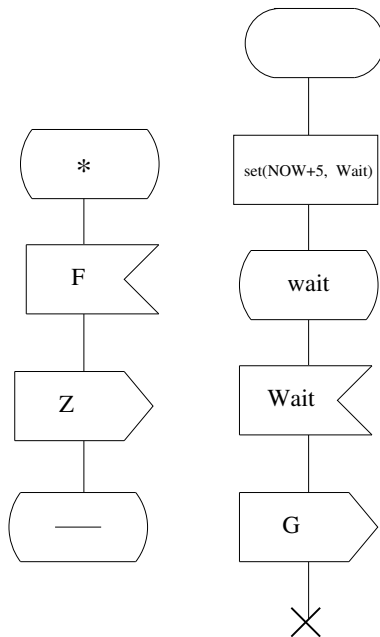
Verfeinern Sie nun ihren Ansatz, so dass

- ein unterbrechendes System mit Prioritäten modelliert wird.
- und ein System, das jedem Prozess eine feste Zeitscheibe zu ordnet (\sim TDMA).

Was sind hier die Probleme, wie kann man diese beheben?

Aufgabe 2: Timer und MSCs

Aus der Vorlesung muesste Ihnen folgende Prozessspezifikation bekannt sein:



1. Geben Sie bitte ein MSC an, dass durch diese Spezifikation realisiert wird.
2. Wie sieht der zugrundeliegende finite state transducer aus (Kapitel 2, Seite 50)?
3. Wie sieht der FST aus, wenn wir ihn in einen einfachen endlichen Automaten überführen, d.h. wenn Eingabe und Ausgabe getrennt werden, also in eigene Transitionen separiert werden? Welche Sprache wird von diesem Automaten erzeugt? Ist sie regulär?

Aufgabe 3: Alternating bit protocol

Es soll eine Variante des “alternating bit protocol” (ABP) mit Hilfe von SDL spezifiziert werden. Das ABP ist eine einfaches Ablaufschema, um zwischen einem Sender und Empfänger den Verlust oder die Verdoppelung von Nachrichten zu entdecken.

1. Das ABP funktioniert wie folgt:
 - (a) Sender und Empfänger besitzen je ein eigenes Kontrollbit, welche zu Begin einer Session den gleichen Wert haben.
 - (b) Der Sender sendet eine Nachricht zusammen mit seinem Kontrollbit an den Empfänger.
 - (c) Nach Empfang einer Nachricht testet der Empfänger ob sein Kontrollbit mit dem der Nachricht übereinstimmt. Ist dies der Fall, akzeptiert er die Nachricht. Nach Empfang einer Nachricht sendet der Empfänger seinerseits ein “acknowledgment” (ACK). Dieses ACK enthält sein Kontrollbit nach Empfang der zuletzt akzeptierten Nachricht. Nach Empfang einer zu akzeptierenden Nachricht und dem Versand des dazugehörigen ACK, komplementiert (toggled) der Empfänger sein Kontrollbit. Bei Ungleichheit zwischem eigenem Kontrollbit und dem der gerade empfangenen Nachricht, das somit zu einem “nicht-akzeptieren” der Nachricht führt, sendet der Empfänger das letzte versendete ACK erneut.
 - (d) Bei Empfang eines ACK überprüft der Sender, ob sein Kontrollbit mit dem des ACKs übereinstimmt. Ist dies nicht der Fall, so wiederholt er das Senden der letzten Nachricht. Hat das Kontrollbit der Nachricht hingegen den gleichen Wert, so toggled der Sender ebenfalls sein Kontrollbit und sendet ggf. die nächste

Nachricht. Diese neue Nachricht wird dabei mit dem neuen Kontrollbit an den Empfänger gesendet und das Verfahren beginnt somit von vorne.

- (e) Um mit dem Verlust von Nachrichten umgehen zu können sollen Sender und Empfänger mit einem Timer ausgestattet werden. Nach dessen Ablauf gilt es dann die letzte Nachricht erneut zu senden.
 - (f) Schlägt das Senden einer Nachricht oder eines ACKs N -mal fehl, so ist davon auszugehen, dass die Leitung dauerhaft gestört ist. Sender und Empfänger signalisieren dies an die Umgebung und gehen in einen Fehlerzustand über, der nur durch Erhalt eines Benutzersignals wieder verlassen werden kann.
2. Um das obige System realistischer erscheinen zu lassen, sollen Sender und Empfänger über einen Mediumsdienst (MD) miteinander verbunden sein und nicht direkt. Hierbei nimmt der MD die zu übermittelnden Nachrichten des Senders entgegen und sendet diese an den MD des Empfängers. Demgegenüber nimmt der MD des Empfängers Nachrichten vom MD des Senders entgegen und reicht sie an den Empfänger weiter. Das Senden / Empfangen von ACKs läuft analog aber in die andere Richtung.
 3. Bei dem MD handelt es sich um die Modellierung eines unsicheren Kanals, d.h. die jeweiligen Nachrichten können durch den MD vernichtet bzw. verfälscht werden. Dieses nicht-deterministische Verhalten soll durch die SDL *any*-Entscheidungsabfrage modelliert werden, es gilt dabei jeweils folgendes zu berücksichtigen:
 - (a) Der MD verliert eine un spezifizierte Anzahl von Nachrichten, dieser Verlust wird durch einfaches Löschen der Nachricht simuliert.
 - (b) Darüberhinaus überprüft der MD eintreffende Nachrichten auf Korrektheit (bspw. durch CRC, das nicht zu implementieren ist). Dabei wird eine nicht spezifizierte Anzahl von Nachrichten als fehlerhaft erkannt. Ist eine Nachricht als fehlerhaft bewertet worden, wird dies dem Sender als auch dem potentiellen Empfänger angezeigt. Diese reagieren dann mit dem erneuten Senden der letzten Nachricht bzw. des letzten ACKs darauf.
 - (c) Es gilt hier das Prinzip der Fairness, d.h. eine Nachricht oder ein ACK können nicht unendlich oft vernichtet oder verfälscht werden, eine Obergrenze bzgl. des wiederholten Vernichtens oder Verfälschens einer Nachricht ist hingegen nicht bekannt.
 4. Spezifizieren Sie für das obige SDL-System das MSC, das folgenden Ablauf beschreibt:

Es wird eine Übertragungsanfrage (*StartIt*) in das System geschickt. Das Medium überträgt das Signal Korrekt, aber bevor die Quittung die Sendeseite erreicht, läuft der Timer ab. Dies führt zu einer erneuten Versendung des Datenpaketes.

Aufgabe 4: Real-time constraints

Spezifizieren Sie einen Mediumsdienst, der die an ihn gesendeten Nachrichten weiterleitet und mindestens alle 10 Zeiteinheiten (ZE) eine Nachricht verschickt. D.h. liegen keine weiterzuleitenden Nachrichten vor, wird einfach ein dummy Datensatz verschickt. Können Sie für Ihre Spezifikation eine Echtzeitgarantie geben? (Begründung ist wesentlicher Teil der Antwort).

Zusatzaufgabe :

Im Netz finden Sie die Artikel

- Ingmar Fliege, Reinhard Gotzhein: Automated Generation of Micro Protocol Descriptions from SDL Design Specifications. SDL Forum 2007; ¹
- Frank Slomka, Matthias Dörfel, Ralf Münzenberger: Generating mixing hardware/software systems from SDL specifications, CODES '01; ²

Diese Artikel sind zum Selbststudium gedacht, da sie den Einsatz von SDL anschaulich darstellen.

Zuständig für diese Übung ist:

Name : Kai Lampka
Gebäude : ETZG, Gloriastr. 35
Raum : G 77
E-Mail : Lampka@tik.ee.ethz.ch

¹www.springerlink.com

²www.acm.org