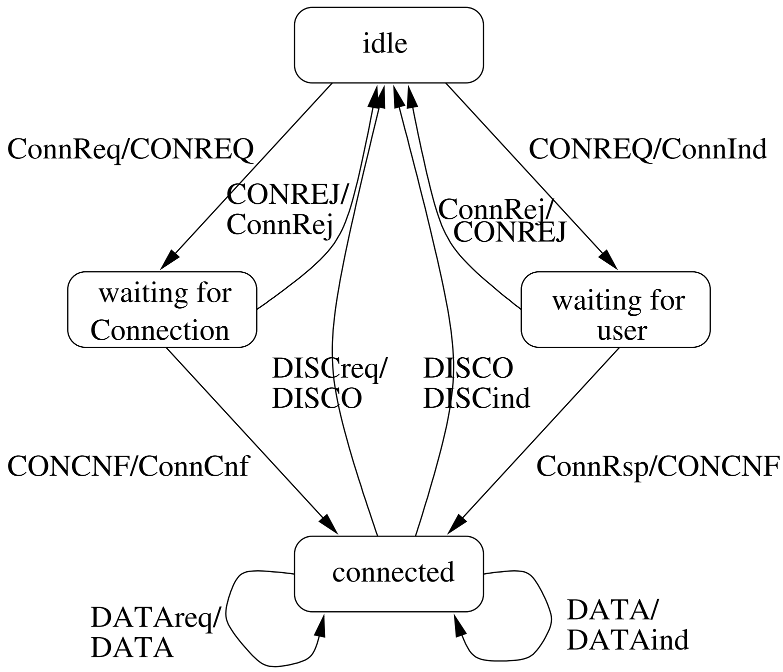
Universität Mannheim
Lehrstuhl für Praktische Informatik IV
Dr. Stefan Fischer
Tel.: 292-1407
Email: stefis@pi4.informatik.uni-mannheim.de

Lösungen Übungsblatt Nr. 2

Aufgabe 1: Estelle/Real-Time Estelle

(a) Erweitern Sie das Protokoll von Blatt 1, Aufgabe 1, um eine Datenphase und den Verbindungsabbau.

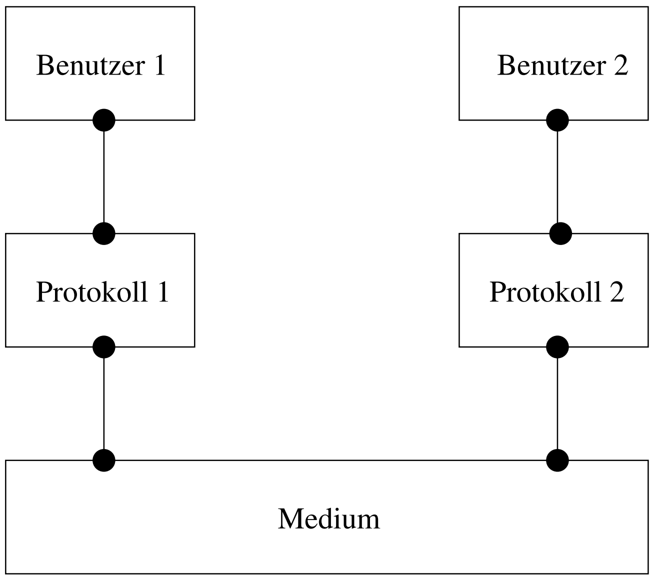
Lösung:



(b) Spezifizieren Sie dieses neue Protokoll in Estelle. Verwenden Sie dabei die Module Benutzer, Protokoll und Kommunikationsmedium.

Lösung:

Modulstruktur:



Spezifikation:

```
specification beispiel;
default indvidual queue;
timescale seconds;

channel protokoll_benutzer(protokoll, benutzer);
by protokoll:
    CONNcnf;
    CONNind;
    CONNcnf;
    DATAind;
    DISCind;
by benutzer:
    CONNreq;
    CONNind;
    CONNrej;
    DATAreq;
    DISCreq;

channel protokoll_medium(protokoll,medium);
by protokoll:
    CONNREQ;
    CONNCNF;
```

```
    CONNREJ;
    DISCO;
    DATA;
by medium:
    CONNREQ;
    CONNCNF;
    CONNREJ;
    DISCO;
    DATA;

module benutzer process;
ip to_protokoll:protokoll_benutzer(benutzer);
end;

body b_body for benutzer; external; (* wird nicht beschrieben *)

module medium process;
ip to_protokoll: ARRAY[1..2] OF protokoll_medium(medium);
end;

body m_body for medium; external;

module protokoll process;
ip  to_medium: protokoll_medium(protokoll);
    to_benutzer: protokoll_benutzer(protokoll);
end;

body p_body for protokoll;
state  idle,  waiting_for_connection,  waiting_for_user,  connec-
ted;

initialize to idle begin end;

TRANS
    FROM idle TO waiting_for_connection
    WHEN to_benutzer.CONNreq begin
        output to_medium.CONNREQ;
    end;

    FROM idle TO waiting_for_user
    WHEN to_medium.CONNREQ begin
        output to_benutzer.CONNind;
    end;

    FROM waiting_for_connection TO connected
    WHEN to_medium.CONNCNF begin
        output to_benutzer.CONNcnf;
    end;

 (* usw. fuer alle Pfeile in der Grafik aus Teil (a) *)
```

```
end; (* of body *)

(* jetzt die Initialisierung des gesamten Systems *)
MODVAR
    b: ARRAY[1..2] OF benutzer;
    p: ARRAY[1..2] OF protokoll;
    m: medium;

initialize
    VAR i:integer;
begin
    init m with m_body;
    FOR i := 1 TO 2 DO begin
        init b[i] with b_body;
        init p[i] with p_body;
        connect b[i].to_protokoll to p[i].to_benutzer;
        connect p[i].to_medium to m.to_protokoll[i];
    end;
end.
```

(c) Überlegen Sie sich sinnvolle Zeitrestriktionen für dieses Protokoll.

Lösung:

1. Eine Protokolldateneinheit darf sich maximal a Zeiteinheiten im Protokoll aufhalten.
2. Die Übertragung einer Dateneinheit von Benutzer 1 zu Benutzer 2 darf maximal b Zeiteinheiten benötigen.
3. Der Verbindungsaufbau darf nicht mehr als c Zeiteinheiten dauern.

(d) Spezifizieren Sie diese Restriktionen in Real-Time Estelle.

Lösung:

1. Formuliert im Protokollmodul

```
FORALL x: TIME; HENCEFORTH (
    RECEIVING OF to_benutzer.DATAreq AT x LEDASTO
    SENDING OF to_medium.DATA At (x < now <= x+a)
);
```

2. Formuliert im Spezifikationsmodul

```
FORALL x: TIME; HENCEFORTH (
    SENDING of b[1].to_protokoll.DATAreq At x LEADSTO
    RECEIVING OF b[2].to_protokoll.DATAind AT (x < now <= x+b)
);
```

3. Formuliert im Benutzermodul

```
FORALL x:TIME; HENCEFORTH (
    SENDING OF to_protokoll.CONNreq AT x LEADSTO
    RECEIVING OF to_protokoll.CONNcnf AT (x < now <= x+c)
);
```

Aufgabe 2: Real-Time Estelle

Überlegen Sie, wie Sie aus einer Real-Time Estelle-Spezifikation automatisch Code für eine Implementierung ableiten könnten, der für eine Einhaltung der spezifizierten Echtzeitrestriktionen sorgt.

Lösung:

s. dazu das sich ebenfalls im Web befindliche Paper sowie die Vorlesung, Kapitel 6