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

Übungsblatt Nr. 1

Aufgabe 1: Softwarelebenszyklus, Systementwicklung

Sie sollen ein Protokoll entwickeln, das den Verbindungsaufbau zwischen zwei kommunizierenden Instanzen vornimmt. Beschreiben Sie ausführlich die verschiedenen Entwicklungsschritte anhand der Phasen des Softwarelebenszyklus. Sie sind bzgl. des Designs dieses Protokolls völlig frei. Gehen Sie zumindest auf die Phasen Anforderungsanalyse, Systemspezifikation, Verifikation und Implementierung ein.

Lösung:

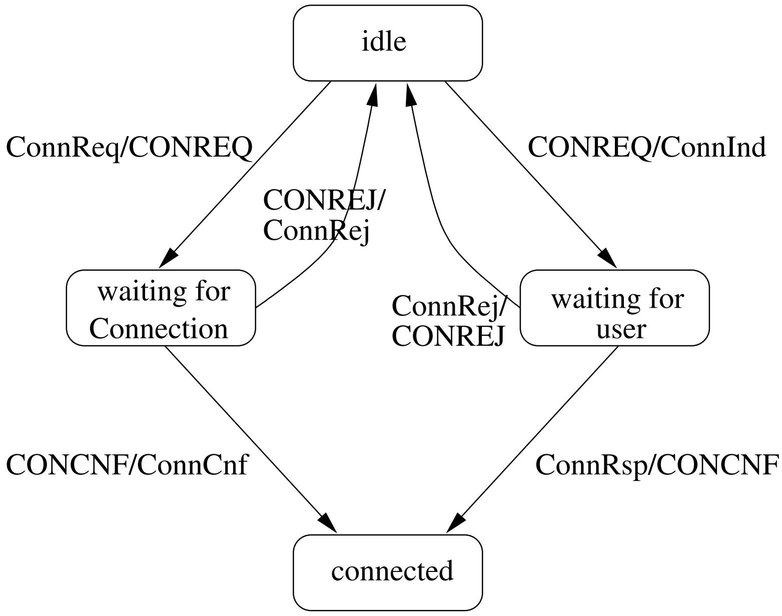
1. Anforderungen an das Protokoll:
- Es soll der Verbindungsaufbau zwischen zwei Benutzern durchgeführt werden.
- Beide Benutzer sollen denselben Dienst verwenden können (symmetrisch).
- Wenn ein Benutzer einen Verbindungsaufbauwunsch absetzt, dann erhält er entweder eine Bestätigung des Aufbaus oder eine Ablehnung
- Wenn dieser Benutzer eine Bestätigung erhält, dann steht die Verbindung und kann sofort zur Datenübertragung genutzt werden.
- Wenn ein Benutzer eine Verbindungsanfrage erhalt, kann er diese entweder annehmen oder ablehnen.
- Wenn ein Benutzer eine Bestätigung oder Ablehnung schickt, ohne daß er eine Anfage bekommen hat, ändert sich am Zustand des Dienstes nichts, d.h., eine solche Nachricht wird verworfen.

2. Formale Spezifikation
- wird aufgeteilt in zwei Phasen
- erste Phase: Spezifikation in Logik
- zweite Phase: Spezifikation als Automat

2.1. Logik-Spezifikation

Formuliere die Diensteigenschaften mit Hilfe temporaler Logik

2.2. Automatenspezifikation



3. Verifikation

Model Checking: Generierung von Zustandsfolgen mit Hilfe der Automaten, Überprüfung der Folgen

4. Implementierung

Implementierung kann mit Hilfe von CASE-Tools erfolgen, die eine Automatenspezifikation in ein laufendes Programm umsetzen.

Sei ein Automat z.B. in Tabellenform gegeben:

Zustand/ Ereignis	z1	z2	z3
e1	a1/z2	---	---
e2	---	a2/z3	---
e3	---	a3/z1	a4/z1

Dann kann ein Automateninterpretierer etwa wie folgt aussehen:

```
TYPE
    states = enum{z1,z2,z3};
    events = enum{e1,e2,e3};
    outputs= enum{a1,a2,a3};
```

```
VAR nextstate: ARRAY[states][events] OF states;
    ausgabe: ARRAY[states][events] OF outputs;

    aktueller_zustand : states;
    aktuelles_ereignis: events;

PROCEDURE send(out:outpits);
FUNCTION hole_eingabe: events;

aktueller_zustand := z1;

while (TRUE) DO begin
    aktuelles_ereignis := hole_eingabe;
    send(ausgabe[aktueller_zustand][aktuelles_ereignis]);
    aktueller_zustand :=
        nextstate[aktueller_zustand][aktuelles_ereignis];
end;
```

Aufgabe 2: Multimedia-Workstation

Nennen Sie Beispiele für die verschiedenen Komponenten einer Multimedia-Workstation, z.B. anhand Ihnen bekannter PC-Produkte.

Lösung:

Anwendung:

Programmierung:

Betriebssysteme:

Kommunikationssysteme:

Hardware:

Aufgabe 3: Datenraten

Welche Datenrate ist zur digitalen Übertragung eines Telefongesprächs notwendig? Warum ist das so? Welche Datenrate stellt ISDN zur Verfügung?

Lösung:

- Zur übertragung der menschlichen Stimme genügt es, etwas mehr als 3000 Hz des Frequenzraums zu übertragen

- Nach dem Abtasttheorem von Shannon genügt es dann, mit der doppelten Rate abzutatsten. Zur Qualitätsverbesserung wählt man gewöhnlich 8000 Hz als Abtastrate.
- Jeder Abtastwert wird mit 8 bit kodiert, d.h., es gibt 256 versch. mögliche Werte.
- Damit ist die Übertragungsrate $8000\frac{\text{Samples}}{\text{s}} \cdot 8\frac{bit}{\text{Sample}} = 64\frac{kbit}{s}$.
- Die Datenrate von ISDN is ebenfalls 64 kbit/s.
- Die Datenrate ist jedoch unkomprimiert. So ist etwa Bildtelefonie mit 2 Kanälen zu je 64 kbit/s schon recht gut durchführbar. Dabei wird die Tatsache ausgenutzt, daß sich das Bild recht wenig verändert („moving head“).

Aufgabe 4: QoS Mapping

(a) Ein Benutzer wünscht die Übertragung eines Videofilms in „sehr guter Qualität“, also mit hoher Farbauflösung, hoher Bildwiederholrate und voller Bildschirmdarstellung. Geben Sie sinnvolle Ober- und Untergrenzen für die benötigte Bandbreite an.

Lösung:

Beispiel: Bildschirmgröße 800x600, 24 Bit Farbtiefe, 30 Bilder pro Sekunde

Das ergibt eine obere Grenze für die Datenrate von

$$800 \times 600 \cdot \frac{\text{Pixel}}{\text{Bild}} \cdot 24 \cdot \frac{\text{Bit}}{\text{Pixel}} \cdot 30 \cdot \frac{\text{Bilder}}{\text{s}} = 345,6 \cdot \frac{\text{MBit}}{\text{s}}$$

MPEG-Kompression: sehr gute Qualität läßt sich ab 4 MBit/s aufwärts realisieren.

(b) Nehmen Sie nun an, dieser Benutzer ist ein Chirurg, der einem Kollegen im einem anderen Krankenhaus live mit Hilfe eines Videosystems Hinweise zu einer komplizierten Operation gibt. Welche Anforderungen hat der Helfer an die Videoübertragung? Welche Transportparameter werden von diesen Anforderungen auf welche Weise beeinflußt?

Lösung:

Wichtige Anforderungen sind eine sehr geringe Verzögerung, damit beide Partner auch wirklich über das reden, was sie gerade machen, sowie eine exzellente Bildqualität, möglichst ohne Fehler. Ein Übertragungsfehler an einer wichtigen Stelle im MPEG-Strom kann dazu führen, daß für einige Zeit bestimmte Stellen auf dem Bildschirm nicht aktualisert werden können. Dies ist jedoch bei Operationen, bei denen es auf jedes Detail ankommt, nicht akzeptabel.

Wegen des geringen Delays kommt eine Retransmission falsch oder gar nicht angekommener Pakete nicht in Frage. Um die Zuverlässigkeit zu erhöhen, wird daher *Forward Error Correction* (FEC) eingesetzt. Dies erhöht die Zahl der zu sendenden Pakete und damit den Durchsatz. Geringere Verzögerung erfordert evtl. die Verwendung eines anderen netzes bzw. die Reservierung einer größeren Menge von Ressourcen. Damit werden sich auch die Kosten erhöhen, die der Benutzer bezahlen muß.

Aufgabe 5: Token Ring für Multimedia

(a) Mit welchem Mechanismus können maximale Verzögerungen im Token Ring garantiert werden?

Lösung:

MAC-Level Prioritäten in Verbindung mit einem Bandbreitenmanager

(b) Nennen Sie die verschiedenen Bearbeitungsschritte, die notwendig sind, um einen Videostrom von einem Benutzer eines Token Ring zu einem anderen Benutzer (Ende-zu-Ende) zu übertragen.

Lösung:

- Datenkompression und Paketierung
- Wartezeit, bis das Token verfügbar ist (Access Delay)
- Sendezeit (Transmission Delay)
- Entpaketierung, Dekompression

(c) Zu welcher zusätzlichen Paketierungsverzögerung führt eine Paketgröße von 4 KByte in einem Videostrom von 1,4 MBit/s?

Lösung:

Paktierungsverzögerung tritt auf, weil eine bestimmte Paktegröße gegeben ist. Erst wenn ein Paket vollständig mit Daten gefüllt ist, kann es verschickt werden. Die ersten Daten im Paket müssen daher entsprechend lange warten.

Bei 1,4 MBit/s dauert es $\frac{4 \text{ Kbyte}}{1{,}4 \text{ MBit/s}} = 22{,}32 \text{ ms}$, um ein 4KByte Paket zu füllen.

Daher sollten bestimmte maximale Paketgrößen nicht überschritten werden, um ein zu hohes Delay zu vermeiden.

(d) Wie lange dauert es, ein solches Paket auf einem 16 MBit/s-Token Ring zum Empfänger zu übertragen (unter Vernachlässigung der Ausbreitungsdauer auf dem Medium)?

Lösung:

$$\frac{4 \text{ KByte}}{16 \text{ MBit/s}} = 1{,}95 \text{ ms}$$

(e) Wie lange muß eine Station maximal auf das Senderecht warten, wenn insgesamt Bandbreite für 8 Ströme der Größe 1,4 MBit/s allokiert wurde, die jeweils 4 KByte Pakete versenden?

Lösung:

Worst-Case-Situation: eine low-priority-Station sendet gerade ein Paket, alle anderen acht Stationen warten auf eine Übertragungsmöglichkeit.
Senden eines Paketes dauert 1,95 ms, dann muß eine Station maximal $8 \cdot 1{,}95\ \text{ms} = 15{,}6\ \text{ms}$ warten.

Aufgabe 6: Cooperative QoS Management

(a) Beschreiben Sie den QoS-Verhandlungsvorgang bei einem traditionellen Ende-zu-Ende - QoS-Management-Verfahren kurz vor Beginn des Fußballweltmeisterschafts-Endspiel!

Lösung:

Alle Empfänger verhandeln kurz vor Spielbeginn mit dem Sender, bei dem es schnell zu einer Überlastungssituation kommt.

(b) Beschreiben Sie nun denselben Vorgang bei Cooperative QoS in einem realistischen Netzwerk-Szenario.

Lösung:

Im Coop. QoS kommunizieren nur benachbarte Knoten des Multicast-Baumes miteinander, und ein Empfänger mu"s niemals bis zur Quelle gehen, um seinen QoS zu verhandeln. Dadurch ergibt sich kein Bottleneck ab irgendeiner Stelle des Systems.