Praktikum

Entwicklung von Kommunikationssystemen (SoSe 2000)

Prof. Reinhard Gotzhein, Philipp Schaible

Aufgabe 2 (2 Wochen)

In der ersten Aufgabe haben wir die grundlegende Struktur des zu entwickelnden Kommunikationssystems und ein Teil der Protokollfunktionalität spezifiziert. In dieser Aufgabe soll die Protokollfunktionalität erweitert werden. Anschließend wird die erstellte Entwurfsspezifikation mit unterschiedlichen Verfahren analysiert. Zum Gegenstand dieser Analyse machen wir solche Systemeigenschaften wie Freiheit von Verklemmungen, unspezifiziertem Empfang, Zustandsmehrdeutigkeiten und nicht ausführbaren Transitionen[1]. Auch die Konformität zu den vorgegebenen Kommunikationsanforderungen wird untersucht.

Die Problemstellung aus dem Schritt 1 werden folgendermaßen ergänzt.

- Anmelden und abmelden bei dem Server.
- Auflösung der Situation mit mehrmals anmelden.
- Anfragen, wer alles angemeldet ist.
- Die Kommunikation zwischen den Protokollinstanzen wird als zuverlässig angesehen.
- Die Größe der Basisdienst-Empfangspuffer des Servers wird als begrenzt angesehen. Das bedeutet, daß nur eine bestimmte Anzahl der Signale von dem Server gespeichert werden kann.

Bemerkung1: Es ist sinnvoll, die Zwischenergebnisse (Spezifikationen aus einzelnen Entwicklungsschritten) zwecks Dokumentation und Wiederverwendung aufbewahrt werden. Damit die Ergebnisse aus dem Schritt1 nicht verloren gehen, sollte man eine Kopie der Spezifikation erstellen, welche dann zu einer Schritt2-Spezifikation erweitert wird. Am einfachsten ist es, das ganze Spezifikation-Verzeichnis zu kopieren. Achtung! Die Datei-Pfade sollen dabei relativ zu Dokument-Directory sein und die Target- und Source-Directories sollen auf System-File-Directory eingestellt sein (s. <File><SetDirectories> Menü). Alternativ kann man auch SDL-Import Mechanismus (s. Organizer) benutzen, um eine Kopie der Spezifikation anzulegen.

Bemerkung2: Bei der Erstellung der Spezifikation sollen (Pattern-Gruppe) wenn möglich, die vorhandenen SDL-Muster benutzt werden. Nur wenn für ein Entwurfsproblem kein passendes Muster existiert, überlegt euch eine eigene Lösung.

a) Erstellt die Entwurfsspezifikation. Richtet Euch nach dem folgenden Entwicklungsprozeß.

i) Erstellt zuerst ein Analysemodell (beteiligte Objekte, deren Beziehungen, typische Kommunikationsszenarien). Bedient euch dabei entsprechender Spezifikationstechniken (UML-Objektdiagramme und MSC-Diagramme).

ii) Mit Hilfe des Analysemodells findet passende SDL Muster(Selektion), adaptiert diese (Adaptation) und integriert sie eines nach dem anderen in die Spezifikation (Komposition).

1. Kernvorlesung “Rechnernetze“

b) Wendet die szenarienbasierte Analyse an. Geht dabei folgendermaßen vor.

i) Laßt die Spezifikation in dem Simulator ablaufen. Generiert dabei MSC Diagramme und vergleicht diese mit den Diagrammen aus dem Analysemodell. (Korrigiert die Spezifikation falls nötig.)

***Bemerkung**: Die durch die Analyse (MSC Basiert, Erreichbarkeitsanalyse, reading, usw.) gefundenen Fehler sollen sofort in die dafür vorgesehenen Fragebögen eingetragen werden.*

ii) Beantwortet die Fragen (stichpunktartig, **schriftlich**): Was kann man über diese Art Analyse erreichen? Was kann man nicht erreichen?

c) Wendet die Erreichbarkeitsanalyse an. Geht dabei folgendermaßen vor.

iii) Erstellt einen **Erreichbarkeitsgraphen.** Laßt Euch bei der Erforschung des Zustandsraumes durch Validator unterstützen[1]. Führt eine **Erreichbarkeitsanalyse** durch. Was ist bei der Datenübertragungsphase auffallend? Was kann man tun, um das System doch noch zu analysieren? Führt jetzt die Bit-State Analyse durch (Eine in SDT integrierte Variante der Erreichbarkeitsanalyse). Bestätigt die Ergebnisse der per Hand durchgeführten Erreichbarkeitsanalyse.

Bemerkung: Für die Erreichbarkeitsanalyse geht nur von einem Client aus. Dazu betrachtet man im Server nicht allgemeine sondern die Client-spezifischen Zustände (z.B. *ClientIdle*, *ClientLoggedIn*, *ClientChatOpened*).

iv) Betrachtet die Ergebnisse der Analyse im Bezug auf die Kommunikationsanforderungen für diesen zweiten Schritt. Wie kann man das Protokoll erweitern, damit die Vorgabe (eingeschränktes Pufferplatz beim Server) erfüllt wird?

v) Erweitert das Protokoll um einen einfachen Flußkontrollmechanismus, welcher pro Client die maximale Anzahl der sich unterwegs befindenden Nachrichten hin zum Server auf eine Nachricht begrenzt. Wie wird sich dadurch der Erreichbarkeitsgraph verändern?

d) Führt eine musterbasierte Analyse durch. (Man argumentiert über die Konformität der Spezifikation bezüglich der Anforderungen und die Abwesenheit solcher Fehler wie z.B. Verklemmung oder Unspezifizierter Empfang anhand der Musterstruktur der Spezifikation) An den Stellen wo keine Muster angewendet werden konnten, versucht durch ähnliche Argumentation die Lücke zu schließen. Nutzt dabei die für das Muster definierte Eigenschaften. Welche Fragen (s. o.) lassen sich mit dieser Analyse beantworten? Welche nicht?

1. Achtet auf die Einstellungen des Validators (Set-Specification-All)