Domänenmodell: Fadenkommunikation und -synchronisation

Alexander Humphreys, Reinhard Rösch, Fabian Scheler

15. Mai 2003

Inhaltsverzeichnis

1 Domänendefinition

Gegenstand der Domäne ist die Kommunikation verschiedener Fäden, d.h. der Austausch von Nachrichten aller Art, sowie die Synchronisierung der Abläufe verschiedener Fäden und deren Zugriffe auf gemeinsam benutzte Betriebsmittel/Datenbereiche in einem System mit einem Adressraum. Hier sollen keine Aspekte der Fadenverwaltung betrachtet werden, auch die Verwaltung der Warteschlangen der bei Synchronisation blockierten Fäden soll hier nicht behandelt werden.

2 Domänenlexikon

and/or/not Die jeweiligen Verknüpfungen einer oder mehrerer Bedingungen.

Avoidance Vermeidung von Deadlocks.

Banker's Algorithm Ein Verfahren zur Vermeidung von Deadlocks.

Betriebsmittelgraphen Eine Möglichkeit zur Erkennung von Deadlocks.

blocking/non blocking Ist ein angefordertes Betriebsmittel nicht vefügbar, wartet der anfordernde Faden / wartet der anfordernde Faden nicht.

binary/multiValue Eine Semaphore kann nur zwei oder auch mehrere Werte annehmen.

CompareSwap Hierbei werden zwei Werte verglichen, sind Beide gleich, wird einer der beiden durch einen dritten Wert ausgetauscht. Ergebnis der Operation ist eine Auskunft darüber, ob der Wert erfolgreich gegen einen anderen ausgetauscht werden konnte, oder nicht. Der komplette Vorgang erfolgt ununterbrechbar, oft ist eine entsprechende Prozessor-Anweisung vorhanden.

Critical Section Ein Bereich des Kontrollflusses, der zu einem bestimmten Zeitpunkt nur von einem Faden durchlaufen werden darf.

Deadlock Mehrere Fäden warten zyklisch auf nicht verfügbare Betriebsmittel.

Detection Erkennung von Deadlocks.

dynQueueSize/statQueueSize Die Größe der Warteschlange der auf ein belegtes Betriebsmittel wartender Fäden wird statisch festgelegt, oder dynamisch ermittelt.

dynSize/statSize Die Größe des Puffers z.B. einer Mailbox wird dynamisch bzw. statisch festgelegt.

Event Eine Bedingung, deren Erfüllung zu bestimmten Aktionen im System führen kann (z.B. zur Aktivierung eines Fadens).

Kill Ein Deadlock wird beseitigt, indem einem Faden, der an diesem Deadlock beteiligt ist, die belegten Betriebsmittel entzogen werden und der Faden selbst beendet wird.

Kommunikation Austausch von Nachrichten aller Art zwischen verschiedenen Fäden

Mailbox Ein Nachrichtenpuffer, in den ein Sender seine Nachrichten ablegen kann aus dem und ein Empfänger die Nachrichten entnehmen kann.

Message Passing System Ein Subsystem unterhalb der Anwendungsschicht, das Nachrichten vom Empfänger entgegennimmt und für die Zustellung zum Empfänger sorgt.

Monitor Ein Monitor kapselt ein gemeinsames und nur exklusiv nutzbares Betriebsmittel oder eine Critical Section, und sichert zu, dass zu einem bestimmten Zeitpunkt nur ein Faden Zugriff auf dieses Betriebsmittel erhält, bzw. nur ein Faden die Critical Section durchlaufen kann.

single/multi Ein Faden kann nur auf die Erfüllung einer bzw. mehrerer Bedingungen warten.

Port Ein Kommunikationsendpunkt, ähnlich einem Socket.

Prevention Die Verhinderung von Deadlocks, indem eine der notwendigen Bedingungen für Deadlocks ausgeschlossen wird.

Queue Eine verkettete Liste, an deren Ende Daten eingekettet werden und an deren Kopf Daten entnommen und ausgekettet werden, kann als Nachrichtenpuffer verwendet werden.

Resolution Auflösung von Deadlocks.

resPrio Die Betriebsmittel werden mit Prioritäten versehen, das Belegen und Freigeben von Betriebsmitteln ist nur fallender bzw. steigender Ordnung der Prioritäten möglich.

Ringpuffer Speicherbereich fester Länge, der mit einem Schreib- und einem Lesezeiger zyklisch durchlaufen wird. Ist die vorhandene Anzahl von Feldern befüllt, wird das älteste Datum im Puffer überschrieben.

Semaphore Ein ,,künstliches Betriebsmittel", das verwendet wird, um Criticial Sections zu schützen, die Übeprüfung, ob dises ,,Betriebsmittel" verfügbar ist, und das evtl. anschließende belegen dieses ,,Betriebmittels" erfolgen atomar, d.h. ununterbrechbar.

Synchronized Memory : Ein Speicherabschnitt, den mehrere Fäden gemeinsam nutzen können um Daten auszutauschen. Wie der Name bereits deutlich macht, muss der Zugriff auf diesen Speicherbereich synchronisiert werden.

Signal Eine leere Nachricht, deren Bedeutungs das Signal selbst ist. Ein Signal wird von einem bestimmten Prozedur behandelt oder ignoriert, falls z.B. keine behandelnde Prozedur vorhanden ist.

Socket Ein Kommunikationsendpunkt beim Sender oder Empfänger. Eine Schnittstelle zu einem Messagepassing-System.

Stack Ein Stapel, eingehende Daten/Nachrichten werden oben auf den Stapel abgelegt, und ebenso werden Daten/Nachrichten auch wieder oben vom Stapel entnommen. Kann z.B. als Array oder verkettete Liste implementiert werden.

synchronous/asynchronous Der Sender wartet/wartet nicht bis die Gegenseite oder die Kommunikationsschicht den Empfang einer Nachricht bestätigt hat.

Synchronisation Sequentialisierung der Zugriffe auf gemeinsame und nur exklusiv nutzbare Betriebsmittel, die Sicherung von Critical Sections und die Koordinierung der Abläufe verschiedener Fäden.

TestSet Es wird geprüft, ob eine bestimmte Variable bereits gesetzt ist und falls dies nicht der Fall ist, wird sie anschließend gesetzt. Entscheidend ist hier, dass die Überprüfung und das Setzen ununterbrechbar erfolgen. Zusätzlich wird irgendwie zurückgegeben, ob der Wert gesetzt werden konnte. Diese Operation ist oft als Prozessor-Anweisung verfügbar.

3 Konzeptmodelle

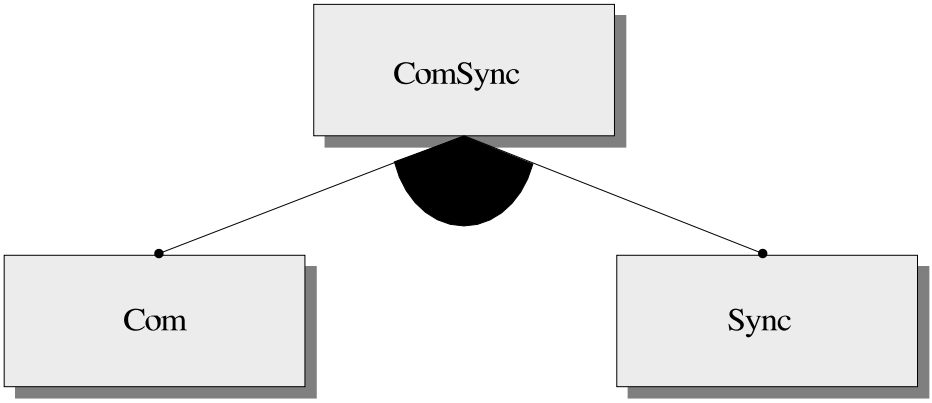
Synchronisation: Der Zugriff auf gemeinsam genutzte Betriebsmittel, der Eintritt in kritische Sektionen muss sequentialisiert werden, Abläufe von verschiedenen Fäden müssen aufeinander abgestimmt werden, z.B. wenn ein Faden auf ein Berechnungsergebnis eines anderen Fadens wartet, hierfür ist die Synchronisation zuständig, die für die Einhaltung gewisser Regeln sorgt, eben z.B. dass sich zu einem gewissen Zeitpunkt nur ein Faden in einem kritischen Abschnitt aufhält. Hierbei ist zu unterscheiden, ob in einem System nur ein Faden oder aber mehrere Fäden vorhanden sein können, oder ob deren Anzahl statisch festgelegt ist, oder sich dynamisch entwickeln kann. Synchronisation kann blockierend, oder nicht blockierend sein.

Kommunikation: Oft ist es notwendig, dass verschiedene Fäden auf irgendwelche Art einander Nachrichten schicken, um z.B. Ergebnisse einer Berechnung oder Ähnlichem mitzuteilen, das Überbringen der Nachrichten geschieht mit Hilfe der Mechanismen, die die Fadenkommunikation zur Verfügung stellt. Hier sind synchrone und asynchrone Kommunikation zu unterscheiden.

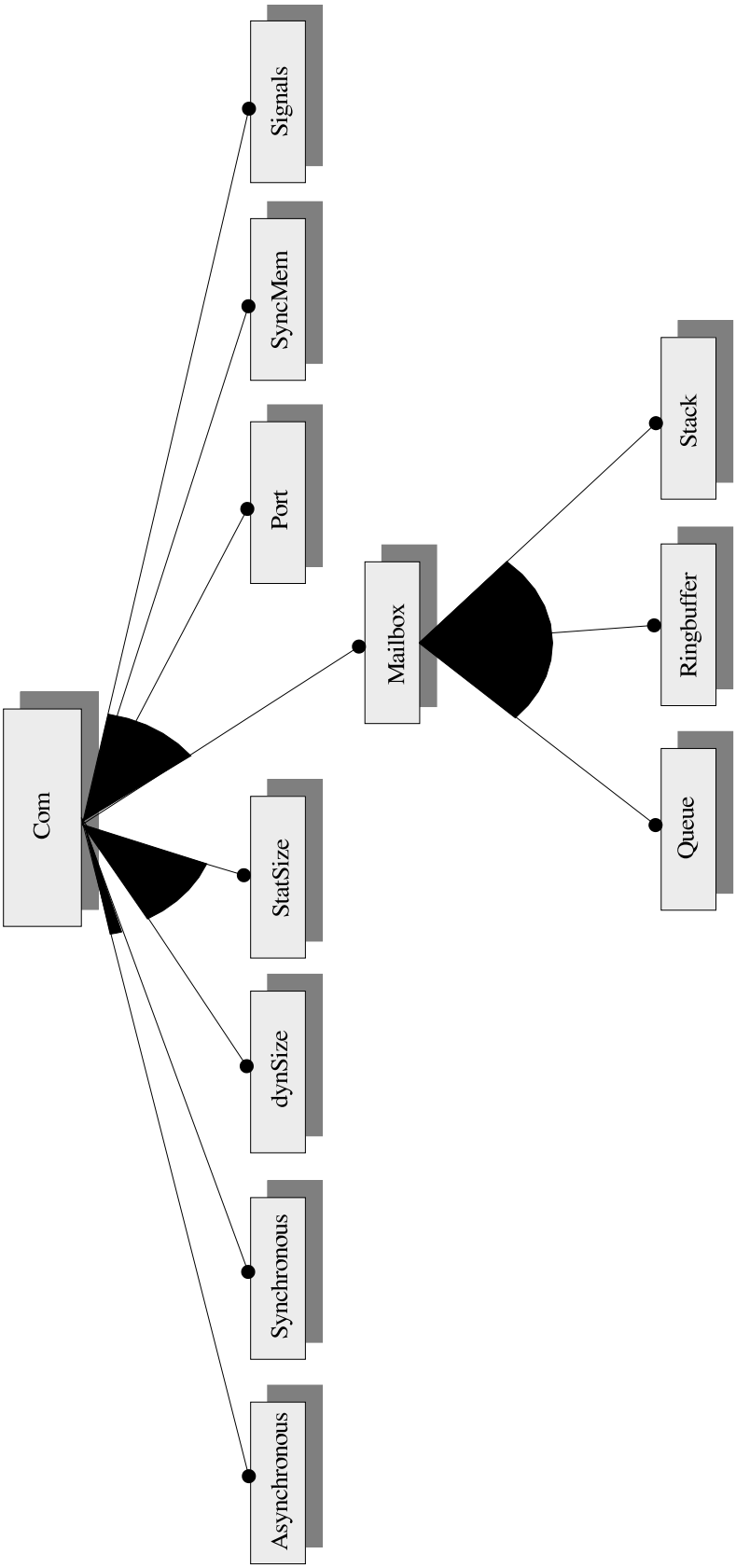
Zusammenhänge: Oft ist es möglich bestimmte Abläufe durch das Verwenden von Nachrichten (z.B. Signale) zu synchronisieren, auf der anderen Seite muss der Zugriff auf die Datenstrukturen, die zur Kommunikation verwendet werden, meistens sequentialisiert also synchronisiert werden, um einen konsistenten Zustand bewahren zu können.

4 Featuremodell

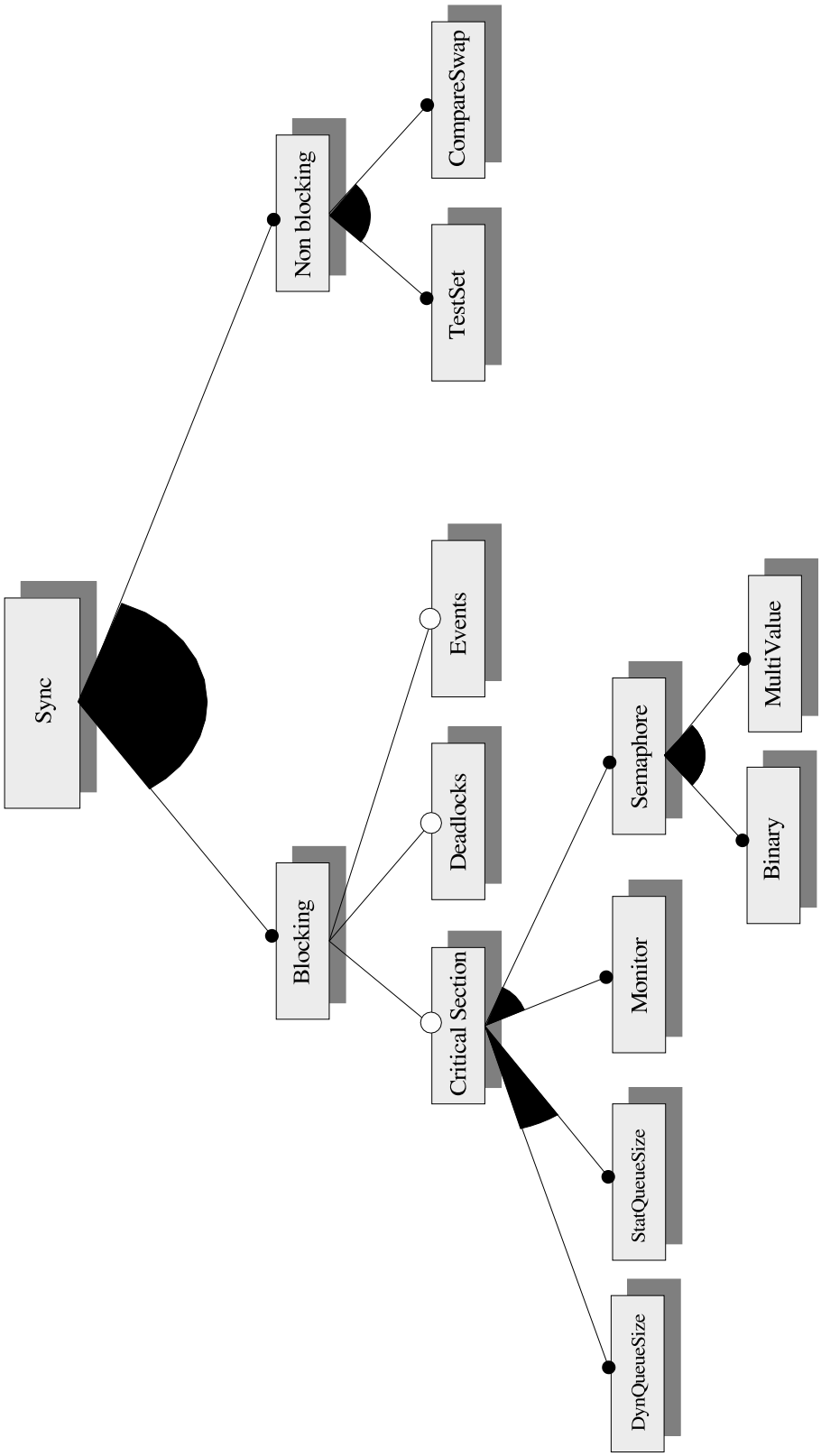
4.1 Kommunikation und Synchronisation



4.2 Kommunikation



4.3 Synchronisation



4.4 Deadlocks und Events