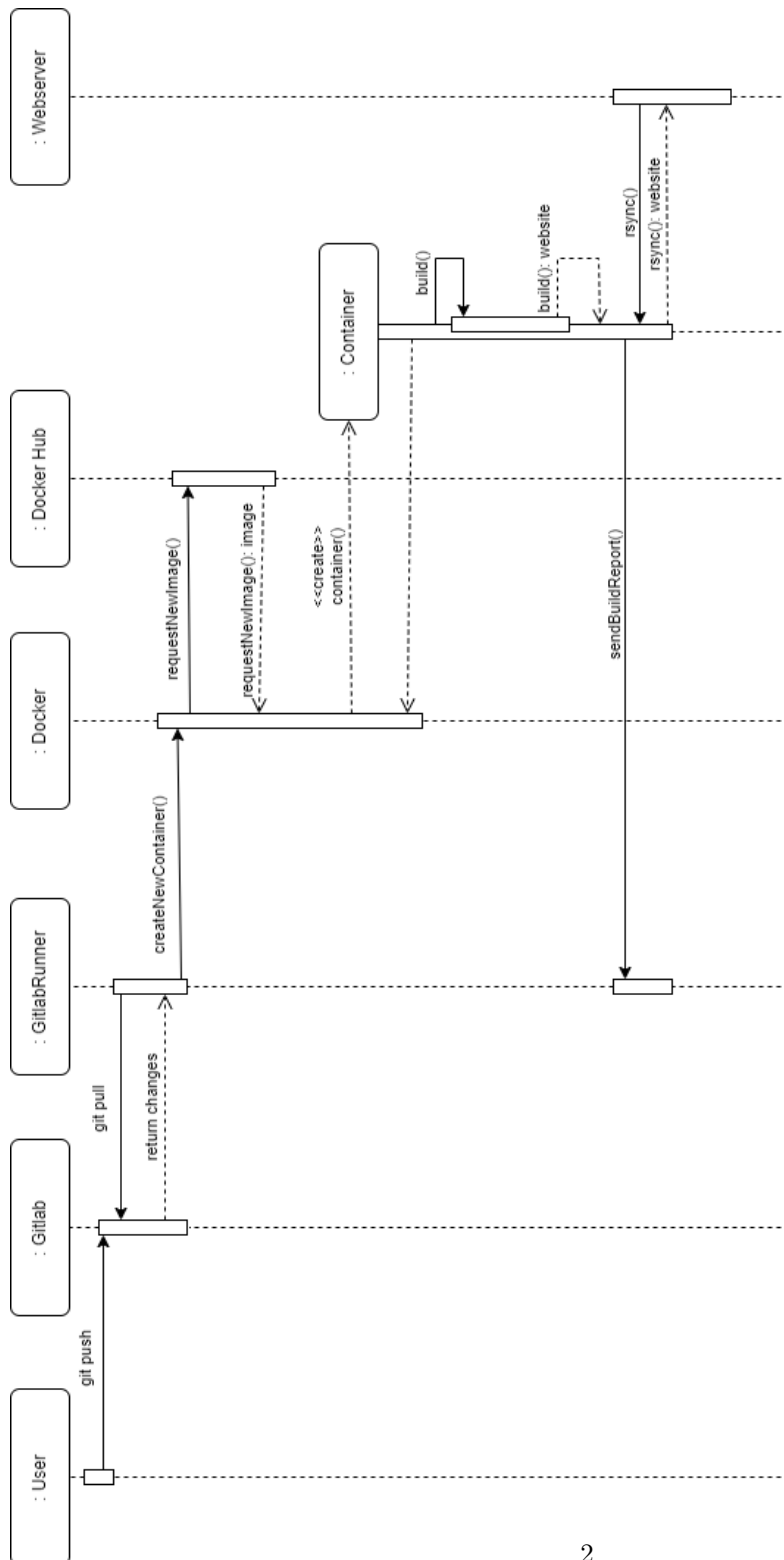


Software Engineering Hausaufgabe 11

Youran Wang (719511, RAS), Yannick Fuchs (723866, ITS)

Januar 2022

1 Sequenzdiagramme komplexer Prozesse



2 Analyse einer algebraischen Spezifikation

2.1

Es handelt sich hierbei um eine algebraische Spezifikation von Polynomen. Mittels dem Befehl $make(poly, nat)$ können wir ein Polynom erstellen und mittels $terms(poly)$ den Wert eines Terms ermitteln. Wenn wir jetzt also beispielsweise $make(r, n)$ ausrechnen, erhalten wir ein Polynom der Form $rx + n$. Sollte man dann den Term auflösen wollen, so kann man dieses Polynom in unsere $terms$ -Funktion einsetzen und angenommen $n = 3$, dann würde die Auflösung wie folgt aussehen: $terms(make(r, n)) = succ(succ(succ(terms(r))))$.

2.2

Dieses Axiom setzt n in das Polynom $make(p, a)$ ein und berechnet den Wert von diesem. Wir lösen also die Gleichung: $p \cdot n + a$.

Dabei berechnen wir erst einmal rekursiv alle Bestandteile des inneren Polynoms. Wir können schließlich auch welche von höheren Graden als 1 haben. Diese werden mittels dem rekursiven Aufruf $evaluate(p, n)$ berechnet. Im nächsten Schritt multiplizieren wir diesen Wert mit n , da für unser "inneres" Polynom p schließlich gilt: $(p)x + a$ und x in diesem Fall unser n darstellt. Zu guter Letzt addieren wir nochmal den Wert von a auf unser Ergebnis und erhalten das Ergebnis des Polynoms an der Stelle $x = n$.

2.3

$$\begin{aligned}
 empty^{\mathcal{M}} &= \langle \rangle \\
 poly^{\mathcal{M}} &= \langle \rangle \mid \langle \omega_0, \dots, \omega_{n-1} \rangle \\
 make^{\mathcal{M}} &= \langle \rangle, n^{\mathcal{N}} \rightarrow \langle n^{\mathcal{N}} \rangle \\
 &\quad \langle \omega_0, \dots, \omega_{n-1} \rangle, n^{\mathcal{N}} \rightarrow \langle \omega_0, \dots, \omega_{n-1}, n^{\mathcal{N}} \rangle \\
 terms^{\mathcal{M}} &= \langle \rangle \rightarrow 0^{\mathcal{N}} \\
 &\quad \langle \omega_0, \dots, \omega_{n-1}, 0^{\mathcal{N}} \rangle \rightarrow terms(\langle \omega_0, \dots, \omega_{n-1} \rangle) \\
 &\quad \langle \omega_0, \dots, \omega_{n-1}, succ(0^{\mathcal{N}}) \rangle \rightarrow succ(terms(\langle \omega_0, \dots, \omega_{n-1} \rangle)) \\
 addPoly^{\mathcal{M}} &= \omega, \langle \rangle \rightarrow \omega \\
 &\quad \omega_0, \omega_1 \rightarrow make^{\mathcal{M}}(addPoly^{\mathcal{M}}(\omega_0, \omega_1), add^{\mathcal{N}}(a^{\mathcal{N}}, b^{\mathcal{N}})) \\
 evaluate^{\mathcal{M}} &= \langle \rangle, n^{\mathcal{N}} \rightarrow 0^{\mathcal{N}} \\
 &\quad \omega^{\mathbf{N}^*}, n^{\mathcal{N}} \rightarrow m^{\mathcal{N}}
 \end{aligned}$$

3 Vier-Ohren-Modell

3.1

In dem Vier-Ohren-Modell geht es darum, dass eine Nachricht, welche wir jemandem vermitteln auf vier Arten unterbewusst aufgenommen wird. Erstens wird es auf der Sachebene wahrgenommen. Darüber läuft die reine Information der Nachricht. Als nächstes gibt es die Appellebene. Hier wird aufgefasst, was der Gesprächspartner mit dieser Information anfangen soll, also z.B. dass er nochmal etwas tun soll um den Sachverhalt zu ändern. Danach folgt die Beziehungsseite. Es schwingt ja auch immer ein gewisser Unterton mit und um den geht es hier. Wir als Sender geben in dieser Seite unterbewusst preis, was wir von unserem gegenüber halten. Als Letztes gibt es die Selbstoffenbarungsebene. Hier werden Informationen des Senders mitgeteilt, welche nicht explizit gesagt werden müssen. Haben wir zum Beispiel die Nachricht "In diesen beiden Zeilen habe ich noch Bugs gefunden.", dann wäre die Selbstoffenbarung: "Ich habe mir nochmal die Zeit genommen und deinen Code durchgeschaut."

3.2

Die Kenntnis dieses Modells kann von Vorteil sein, damit eine reibungslose Kommunikation gewährleistet werden kann und man darauf achten kann, Nachrichten so zu formulieren, dass sich niemand in Meetings/Gesprächen angegriffen oder benachteiligt fühlt.