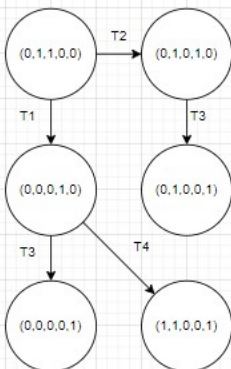


a)



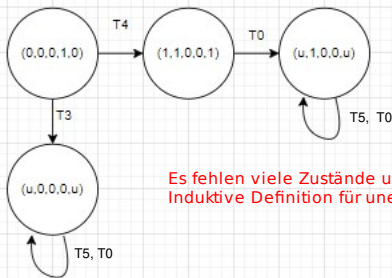
Das Petri-Netz ist tot, da er immer noch einigen Zügen sich keine Marke mehr bewegen kann. Entweder ist die Stelle schon belegt, braucht eine 2 um zu schalten oder es fließt keine Marke zurück.

Einzelne Transitionen fehlen -2

Deadlockfrei? -2

2/6

b)



Es fehlen viele Zustände und Transitionen -6.5

Induktive Definition für unendlich nicht angewandt -2

Deadlockfrei? -0.5

T0 und T5 sind lebendig, da diese immer weiter die Möglichkeit haben werden zu schalten.

T1 kann nicht schalten, da zu s2 nie eine weitere Marke kommen wird und eine Marke bei s1 nicht ausreicht. Dementsprechend ist T2 auch tot.

Der Rest ist tot, da keine Marke zu den Stellen zurückfließt.

0/9

2/15

## Blatt 4

### Aufgabe 12)

a)

#### **Webapplikation <-> Desktop-Applikation <-> App**

- Web-Applikation, da Blatt 3 aufgabe sieben: da „mitarbeiter software über den browser nutzen“

#### **Ein Benutzer <-> mehrere Benutzer**

- Mehrere Benutzer (da mehrere Rechnungswesenmitarbeiter), keine verschiedenen Rollen und Zugriffsrechte (da alle ReWeMitarbeiter die gleichen haben)

#### **Schichten**

- Alle (Präsentationsschicht, Webzugangsschicht, Geschäftslogik, Datenhaltung und Datenbank)

#### **Zuordnung der Schichten zu Client bzw. Server**

- Client: Präsentationsschicht
- Server: Webzugangsschicht, Geschäftslogik, Datenhaltung

#### **Ob und welche DBMS (relational, NoSQL,...)?**

- Relationales DBMS, da fest strukturierte Daten und Transaktionsverwaltung wichtig (ACID)

#### **Eine gemeinsame DB oder ein Datenbestand pro Service?**

- Eine gemeinsame DB um Doppelbuchungen zu vermeiden

#### **Sequentiell <-> parallel/verteilt/nebenläufig?**

- Parallel: Client-Server, da Webserver = Server mit Webschnittstelle und Mitarbeiter = Clients Hier reicht sequentiell da Leistungsanforderungen überschaubar sind. -0.5

#### **Unterschiedliche Plattformen nötig?**

- Nein (Annahme: Alle Mitarbeiter der Firma haben eine einheitliche Plattform)

#### **Wiederverwendung vorhandener Bausteine?**

- Nein, nicht nötig und wäre nur Mehraufwand

#### **Virtualisierung**

- Da keine unterschiedlichen Plattformen auch keine Virtualisierung nötig

#### **Ausgelegt für hohe Anfragerate und Resilienz?**

- Nein, es werden nicht sehr viele Autos verkauft und ungefähr gleichbleibende Rate, deshalb ist Verkaufsanmeldung, Kundendatenprüfung, ... nicht extrem oft nötig → Enterprise-Architektur

#### **On Premises <-> in Cloud?**

- On Premises, da eher keine Schwankungen in Nutzung und Datensicherheit wichtig

#### **Gestaltung der Benutzerschnittstelle?**

- GUI (da Website)

#### **Ob und welche Hilfesysteme?**

- Ja, um den Mitarbeitern das Arbeiten damit so leicht wie möglich zu machen

#### **Einbinden von sonstigen Dienstleistungen?**

- Nein, nicht nötig  
Führt APIs sind nötig. -0.5

b)

#### **Webapplikation <-> Desktop-Applikation <-> App**

- Webapplikation, da so keine Installation nötig und ist plattformübergreifend

### **Ein Benutzer <-> mehrere Benutzer**

- Mehrere Benutzer: Nutzer, Chat-Support-Mitarbeiter und Ranghöhere
- Somit auch unterschiedliche Rollen und Zugriffsrechte

### **Schichten**

- Alle (Präsentationsschicht, Webzugangsschicht, Geschäftslogik, Datenhaltung und Datenbank)

### **Zuordnung der Schichten zu Client bzw. Server**

- Client: Präsentationsschicht
- Server: Webzugangsschicht, Geschäftslogik, Datenhaltung

### **Ob und welche DBMS (relational, NoSQL,...)?**

- Relationales DBMS, da fest strukturierte Daten

### **Eine gemeinsame DB oder ein Datenbestand pro Service?**

- Eine gemeinsame, da nur die Chats speichert (also nur eine hat)

### **Sequentiell <-> parallel/verteilt/nebenläufig?**

- Parallel: Client-Server

### **Unterschiedliche Plattformen nötig?**

- Ja, da nicht weiß was für Plattformen Nutzer haben

### **Wiederverwendung vorhandener Bausteine?**

- Nein, nicht nötig und wäre nur Mehraufwand

### **Virtualisierung**

- Ja -> Container-Virtualisierung für hohe Anfragerate (Docker)

### **Ausgelegt für hohe Anfragerate und Resilienz?**

- Ja, da es „zu hunderten bis zu tausenden Chat-Sitzungen gleichzeitig“ kommt

### **On Premises <-> in Cloud?**

- In Cloud, da Schwankungen in Nutzung wahrscheinlich (falls Fehler bei monatlicher gleichzeitiger Lieferung) und somit kostengünstiger

### **Gestaltung der Benutzerschnittstelle?**

- GUI

### **Ob und welche Hilfesysteme?**

- Ja, um den Nutzern den Chat so leicht wie möglich zu machen

### **Einbinden von sonstigen Dienstleistungen?**

- Ja, vorgefertigtes Chatbot System

Frameworks für high-level Websockets. -0.5

c) Annahme: Design nur vom Server

### **Webapplikation <-> Desktop-Applikation <-> App**

- Desktop-Applikation mit Serverschnittstelle zur Übertragung der zu augmentierenden Dateien

### **Ein Benutzer <-> mehrere Benutzer** Nur ein Benutzer nutzt Applikation. -0.5

- Mehrere Benutzer, verschiedene Rollen und Zugriffsrechte (Annahme: es gibt Unterschiede zwischen kostenloser Version und Upgrades etc.)

### **Schichten**

- Alle (Präsentationsschicht, Webzugangsschicht, Geschäftslogik, Datenhaltung und Datenbank)

### **Zuordnung der Schichten zu Client bzw. Server**

- Client: Präsentationsschicht (Zeichenprogramm)
- Server: Webzugangsschicht, Geschäftslogik, Datenhaltung

### **Ob und welche DBMS (relational, NoSQL,...)?**

- Relationales DBMS, da fest strukturierte Daten und Transaktionsverwaltung wichtig (ACID) Daten nur solange behalten wie nötig. -0.5

#### **Eine gemeinsame DB oder ein Datenbestand pro Service?**

- Ein Datenbestand pro Service (da man nicht auf Bilder von anderen zugreifen kann)

#### **Sequentiell <-> parallel/verteilt/nebenläufig?**

- verteilt: Client-Server

#### **Unterschiedliche Plattformen nötig?**

- Ja, da Nutzer unterschiedliche haben

#### **Wiederverwendung vorhandener Bausteine?**

- Nein, nicht nötig und wäre nur Mehraufwand

#### **Virtualisierung**

- Ja Warum? -0.5

#### **Ausgelegt für hohe Anfragerate und Resilienz?**

- Mittlere Anfragerate, da viele Nutzer hat (Annahme)

#### **On Premises <-> in Cloud?**

- On Premises, da eher keine Schwankungen in Nutzung und Datensicherheit sehr wichtig Variable Anfragemenge dafür Cloud besser. -0.5

#### **Gestaltung der Benutzerschnittstelle?**

- GUI

#### **Ob und welche Hilfesysteme?**

- Ja, um den Nutzern das Arbeiten damit so leicht wie möglich zu machen

#### **Einbinden von sonstigen Dienstleistungen?**

- Nein, nicht nötig erprobte Modelle zur Augmentierung. -0.5