

Softwaretechnik 2

Kommunikationsorientierte Middleware

Verfasser*in: Prof. Dr. Sabine Sachweh

Dr. Nhiem Lu

Datum: 31.05.2022

50/

Das letzte Mal
bei SWT2



электронъ къ <09д

Kommunikationsmodelle

Verteiltes System: Zusammenschluss unabhängiger Rechner, die sich als einzelnes System präsentieren

Verteilte Anwendung: Komplexes Anwendungsprogramm, das in einem verteilten System läuft. Mögliche Aufteilung in mehreren Komponenten/Services.

Middleware: Vermittlungsschicht zwischen Anwendung und Betriebssystem. Bereitstellung einer Kommunikationsinfrastruktur mit dem Ziel alle Aspekte der Netzwerkprogrammierung zu abstrahieren

Kommunikationsmodelle

Middleware

Direkte Netzprogrammierung

	Prozedural	Objektorientiert
Synchron	Entfernte Prozeduraufrufe (RPC)	Entfernte Methodenaufrufe (RMI)
Asynchron	Nachrichtenorientiertes Modell (Messaging Queuing)	

Synchrone Kommunikation

Lokales versus verteiltes Objektmodell

Methodenaufrufe zwischen Objekten in entfernten/verschiedenen Prozessen →
entfernte Methodenaufrufe (Remote Method Invocation)

Kommunikationsablauf → **Sender ist zeitweise blockiert**

Asynchrone Kommunikation

Gekennzeichnet durch das zeitlich versetzte Senden und Empfangen von Informationen

→ Keine Blockierung des Senders

Nachrichtenorientiertes Kommunikationsmodell ermöglicht den asynchronen Transport von Daten innerhalb von Messages zwischen Prozessen

→ **Warteschlangenprinzip (FIFO)**: Entkopplung von Sender und Empfänger

→ **Point-to-Point**: zwischen zwei festgelegten Prozessen/Anwendungen

→ **Request-Reply Modell**: Synchroner Kommunikation (fachlich) über eine asynchrone Middleware (technisch)

→ **Publish-Subscribe**: Eine Nachricht wird an alle erreichbaren Prozesse versendet

→ Publisher, Subscriber, Message-Broker

Asynchrone Kommunikation

Vorteile:

- Die kommunizierenden Prozesse können zu unterschiedlichen Zeiten laufen
- Die Prozesse kommunizieren ohne direkten Kontakt unter Verwendung von Queues
- Queues dienen als Puffer - der Empfängerprozess muss zum Versendezeitpunkt nicht zwangsweise laufen
- Einfaches, klares Kommunikationsmodell
- Laufzeitabhängigkeiten werden minimiert bzw. aufgehoben
- Netzwerkverbindungen werden reduziert
- Programme sind weniger anfällig gegenüber Netzwerkfehlern
- Für lose gekoppelte Systeme sehr gut geeignet

Message-oriented Middleware (MoM)

Java Message Services (JMS): JMS Provider, JMS Client, Message, Queues, Topics

IoT-Protokolle

TCP/IP Referenzmodell

Application	HTTP(S), AMQP, MQTT, MQTT-SN, CoAP FTP, SSH, SMTP, DHCP, POP3, SOAP
Transport	TCP, UDP
Internet	IPv4, IPv6
Network	LoRaWAN, Zigbee, Z-Wave, Sigfox, EnOcean, MIOTY, NB-IoT IEEE 802.3 Ethernet IEEE 802.11 WLAN, Bluetooth, LTE

IoT-Protokolle

Application

HTTP (Hypertext Transfer Protocol):

synchrones Protokoll zur Datenübertragung auf der Anwendungsschicht

→ TCP, zustandslos, Request & Response, GET, PUT, POST, DELETE

AMQP (Advanced Message Queuing Protocol):

Binäres Netzwerkprotokoll für nachrichtenorientierte Kommunikation auf der Anwendungsschicht. Oftmals für Kommunikation zwischen Message-Brokern.

Nachrichten = Exchanges, Nachrichtenkopien = Bindings

MQTT (Message Queue Telemetry Transport):

Vernetzten von IoT-Geräte mit geringen Ressourcen zu ermöglichen. Realisiert das Publish-Subscribe Modell mit QoS Eigenschaften

CoAP (Constrained Application Protocol):

Client-Server Protokoll für stark ressourcenbeschränkte Geräte (Nodes) im M2M Kontext auf der Anwendungsschicht. Leichtgewichtige Alternative zu HTTP und verwendet UDP.

IoT-Protokolle

Network

- Low Power Wide Area Networks (**LPWAN**) große Reichweite und niedrigem Energieverbrauch der Endgeräte
- **LoRaWAN** (Long Ranged Wide Area Network) ist ein drahtloses Netzwerkprotokoll für LPWANs
 - LoRaWAN ist energieeffizient und ermöglicht die Datenübertragung über eine große Entfernung
 - Reichweite erstreckt sich von 2-5 km in Stadtgebieten bis zu 40 km in ländlichen Gebieten mit Datenübertragungsraten von 292 Bit/s bis 50 kbit/s
- **Sigfox** ist ebenso wie LoRaWAN ein verbreitetes Protokoll für LPWAN
- Weitere offene Protokolle für LPWANs: **MIOTY, NB-IoT oder LTE-M**
- **EnOcean** ist ein besonders energieeffizientes Funkprotokoll, welches vor allem für die Gebäudeautomation mittels batterieloser Funksensorik (Energy Harvesting) eingesetzt wird
- **Thread, Z-Wave** und **ZigBee** sind weitere verbreitete Protokolle aus dem Bereich Gebäudeautomation / Smart Home

Kommunikation in verteilten Systemen

REST (Representational State Transfer)

- Architekturstil für verteilte Systeme, Client-Server Architektur
- Große Flexibilität → Eingesetzt für die Realisierung von Webservices
- GET, POST, PUT und DELETE für CRUD-Operationen
- Daten können als JSON, YAML oder XML ausgeliefert werden
- CoAP = REST des kleinen Geräts
- **Architekturprinzipien:**
 - Eindeutige Identifikation von Ressourcen
 - Verwendung von Hypermedia
 - Verwendung von HTTP-Standardmethoden
 - Unterschiedliche Repräsentation von Ressourcen
 - Zustandslosigkeit
- JAX-RS und Spring MVC
- Streaming Plattform: Apache Kafka
- Open-Source Framework: Swagger



Diskussionsaufgaben

Kommunikationsmodelle

- Grenzen Sie die Begriffe verteilte System, verteilte Anwendungen und Middleware von einander ab.
- Was für Charakteristiken haben verteilte Anwendungen? Welche Vor- und Nachteile haben sie?
- Was für einen Zweck verfolgt die Middleware? Was stellt sie bereit im Kontext von verteilten Anwendungen?
- Skizzieren Sie den Kontext einer verteilte Anwendung, einer Middleware und Betriebssystemen.
- Grenzen Sie Middleware und direkte Netzprogrammierung von einander ab.
- In welche Kategorien können Kommunikationsmodelle für Middleware unterteilt werden?
- Was verwenden Sie für eine synchrone/asynchrone prozedurale/objektorientierte Kommunikation?

Synchrone Kommunikation

- Wie werden Methodenaufrufe zwischen Objekten in entfernten/verschiedenen Prozessen realisiert?
- Was passiert mit Prozessen beim Senden oder Empfangen von Informationen?
- Skizzieren Sie den Kommunikationsablauf einer synchronen Kommunikation in einem Client-Server Setting.
- Vergleichen Sie die Eigenschaften von einem lokalen Objektmodell zu einem verteilten Objektmodell

Diskussionsaufgaben

Asynchrone Kommunikation

- Erklären Sie den Nachrichtentransfer in einer asynchrone Kommunikation? Was passiert mit dem Sender?
- Wie werden Daten zwischen Prozessen transferiert?
- Auf welcher Grundlage basiert dieses nachrichtenorientierte Kommunikationsmodell?
- Welches Prinzip verfolgt die Warteschlange? Zu welchem Zweck?
- Erklären Sie den Ablauf von Point-to-Point in diesem Kontext.
- Worin unterscheiden sich Point-to-Point und Request-Reply?
- Erklären Sie das Publish-Subscribe Modell. Was macht der Message-Broker?
- Welche Vor- und Nachteile gibt es in der asynchronen Kommunikation?
- Was verstehen Sie unter einer Message-oriented Middleware (MoM)? Welche Mechanismen und Dienste liefert sie?
- Was verstehen Sie unter dem Java Message Service (JMS) und in welchem Zusammenhang steht dieser zu MoM?
- Im Kontext von JMS: was macht der JMS Provider, der JMS Client? Wie sieht die Message aus?
- Welche Nachrichtenkanäle werden von JMS unterstützt?
- Welche Kommunikationsbeziehungen werden durch Queues oder durch Topics realisiert?

Diskussionsaufgaben

IoT-Protokolle

- Erklären Sie das XYZ Protokoll.
- In welcher Schicht wird es angewendet?
- Welche Zustellungsformen werden verwendet?
- Welche Charakteristiken hat das XYZ Protokoll?
- Welche Vor- und Nachteile hat das XYZ Protokoll?
- Grenzen Sie MQTT und CoAP von einander ab.
- Erklären Sie MQTT anhand eines Beispiels.
- Was ist LPWAN? Und welche besonderen Charakteristiken hat es?
- Was können Sie zu LoRaWAN sagen? Welche Alternative zu LoRaWAN kennen Sie?
- Wofür ist EnOcean besonders geeignet?

Kommunikation in verteilten Systemen

- Was verstehen Sie unter REST? Welche Eigenschaften hat es?
- Welche REST Methoden kennen Sie?
- Welche Architekturprinzipien verfolgt REST?
- Grenzen Sie JAX-RS und Spring MVC von einander ab.
- Im Kontext von Middleware, was verstehen Sie unter einer Streaming Plattform? Geben Sie ein Beispiel hierfür an?
- Was ist Swagger und wofür wird es eingesetzt?

Thanks for watching!

10 Persistenz



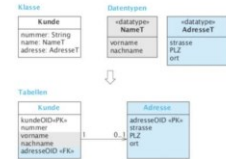
Fachhochschule
Dortmund
University of Applied Sciences and Arts

Objektrelationale Abbildung Komplexer Datentyp



we
focus
on
students

- Optimaler Fall:
Datentypen der UML können auf benutzerdefinierte Datentypen oder Collection-Typen der Datenbank abgebildet werden.
- Alternative
 - Integration des Datentyps NameT in Tabelle Kunde
 - Darstellung des Datentyps AdresseT in eigener Tabelle



10/13 Persistenz

© Prof. Dr. Sabine Schach

SUBSCRIBE

RELATED VIDEO

?



we
focus
on
students

www.fh-dortmund.de

 **FACEBOOK**
facebook.com/fhdortmund

 **TWITTER**
twitter.com/fh_dortmund

 **YOUTUBE**
youtube.com/FachhochschuleDO

 **INSTAGRAM**
instagram.com/fhdortmund

 **XING**
xing.com/companies/fachhochschuledortmund

 **LINKEDIN**
linkedin.com/school/fachhochschule-dortmund

 **KUNUNU**
kununu.com/de/fh-dortmund