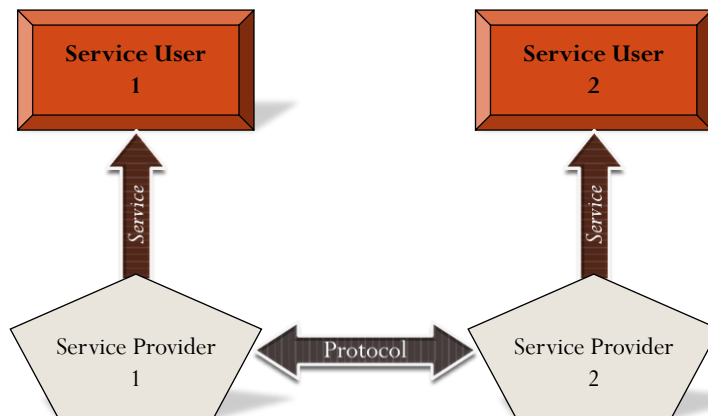


Modeling and Simulation of Communication Systems and Networks

3. Finite State Machines

– Prof. Jochen Seitz –

Service and Protocol (I)



2

Modeling & Simulation - 3. Finite State Machines



Service and Protocol (II)

- **Peer entities** on one layer
 - Utilize the service of the layer below (if there is one)
 - Offer their service to the layer above (if there is one)
 - Only know the interface to the service beneath, but NOT its implementation
 - Communicate with each other according to specific rules
 - Receive input
 - Generate events
- Thus, the service is delivered by **cooperation** of the peer entities in one layer

3

Modeling & Simulation - 3. Finite State Machines

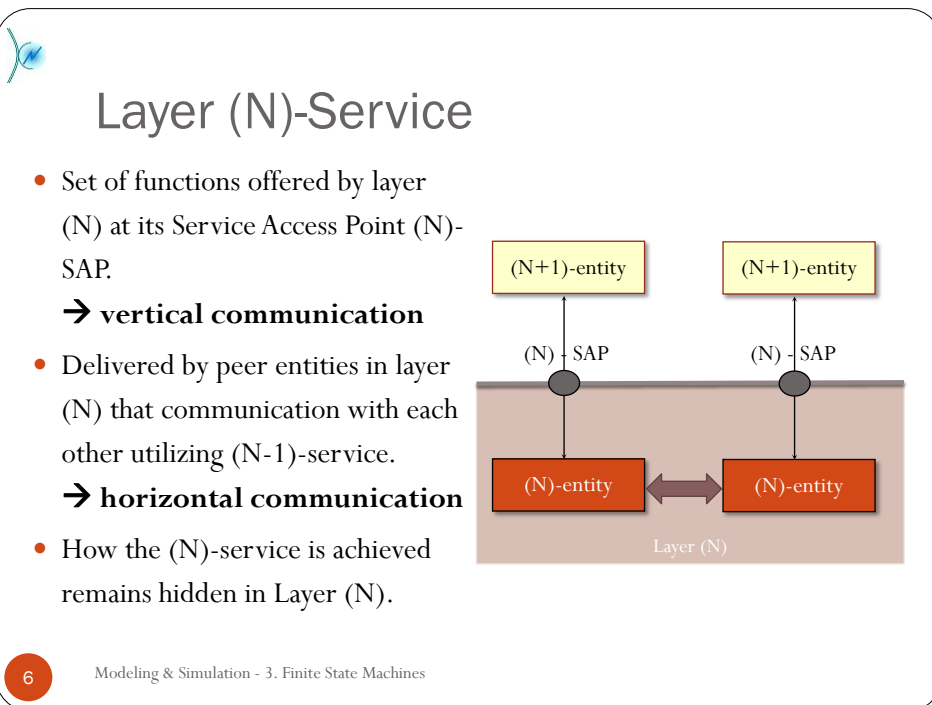
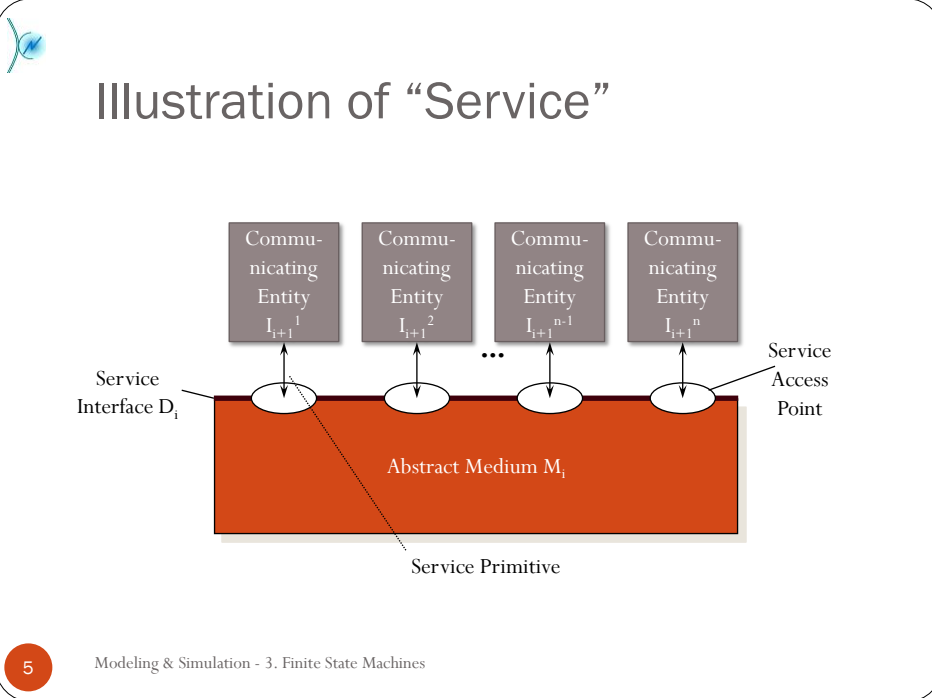


Definition of “Service”

- Service = Set of functions in one layer.
- Delivered by co-operating / communicating entities of the layer, which obey a specific protocol.
- Offered at the service access point of the layer.
- Defined by a set of service primitives and rules for their usage.
- Utilized or indicated by service primitives:
 - *Request*
 - *Indication*
 - *Response*
 - *Confirmation*

4

Modeling & Simulation - 3. Finite State Machines





The Idea Behind Protocols

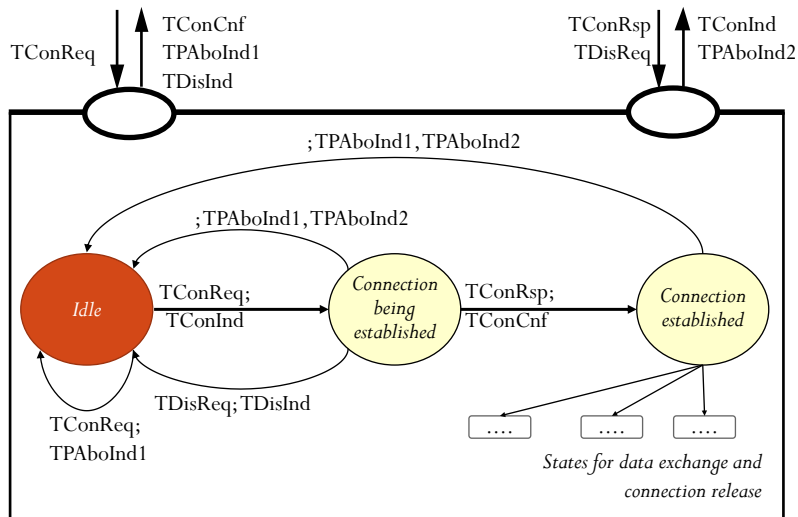
- Rules for concurrently running, but co-operating entities
- Enhancement of the quality and the functions of the layer below
- Distributed algorithm
- Problem:
 - failures and faults must be considered*
- Specification
 - In most cases only for two entities
 - Finite state machines
 - Message-sequence-charts (MSC)

7

Modeling & Simulation - 3. Finite State Machines



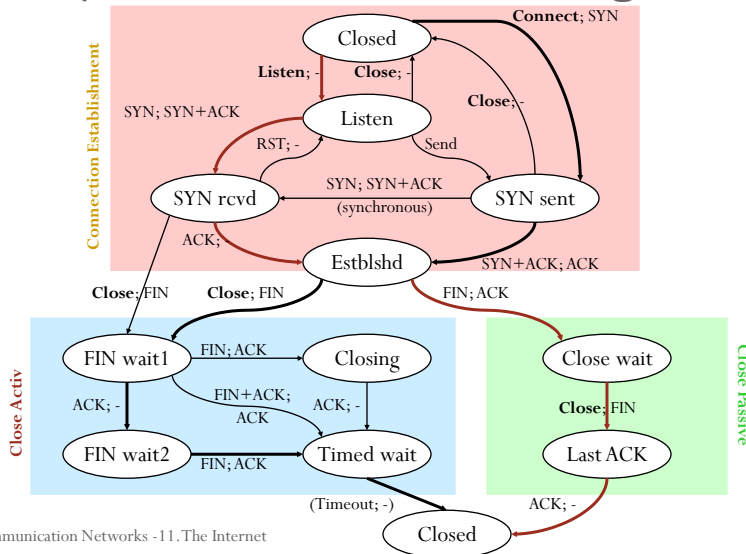
Finite State Machine



8

Modeling & Simulation - 3. Finite State Machines

Example: TCP Connection Management



9

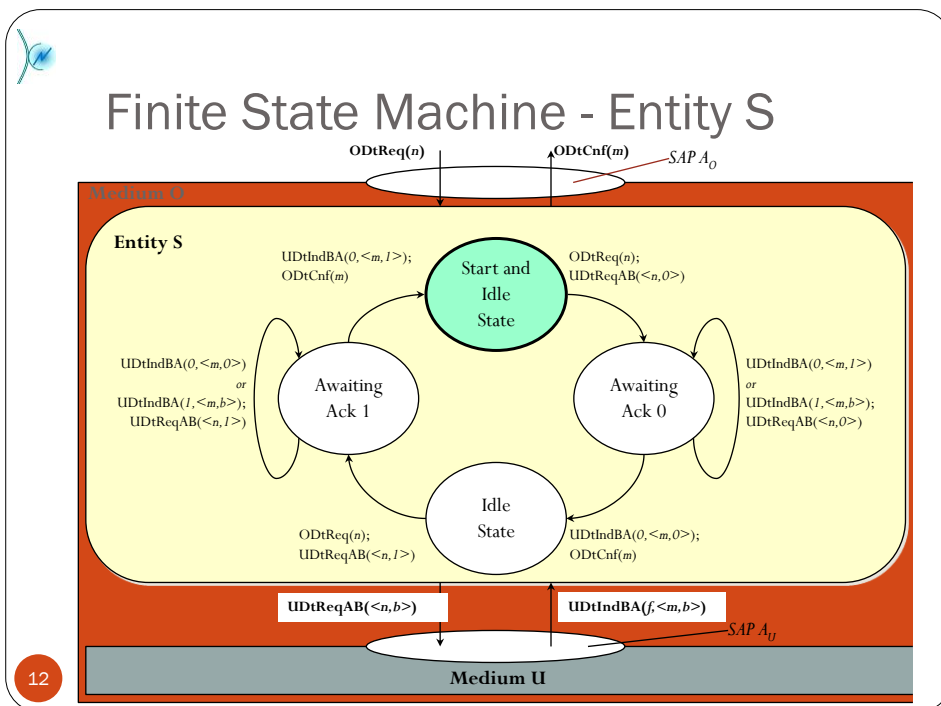
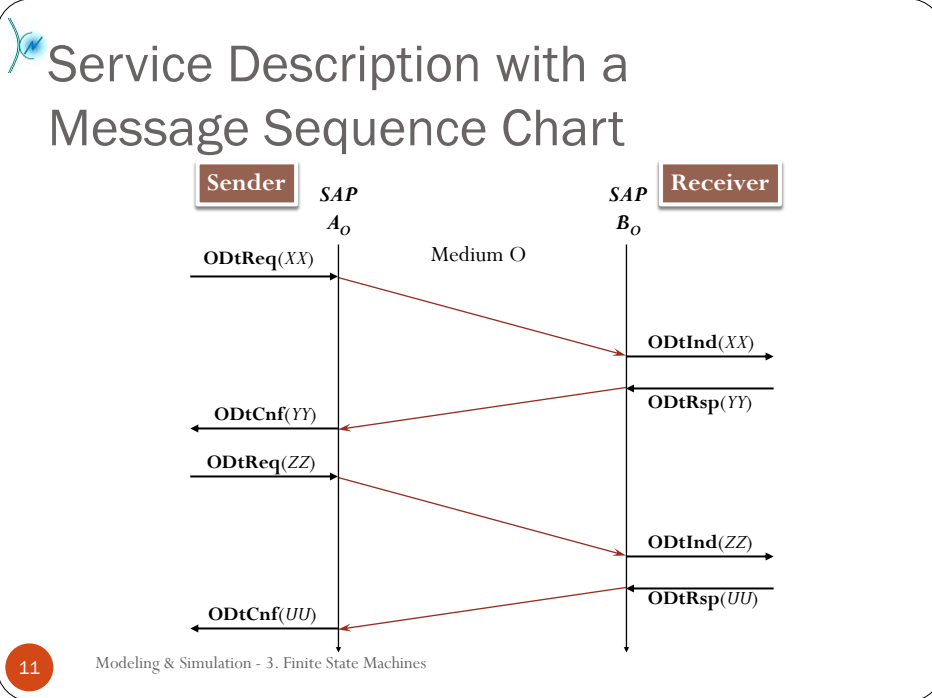
Communication Networks -11. The Internet

Example: Alternating Bit Protocol

- The Alternating Bit Protocol implements a service for confirmed data transfer based on a service for unconfirmed data transfer.
- Service primitives are:
 - *DtReq*;
 - *DtInd*;
 - *DtRsp*;
 - *DtCnf*.
- Acknowledgment is done with alternating acknowledgment numbers 0 and 1. Thus, one can distinguish between message and acknowledgement loss.
- After the confirmed data transfer has been completed, the protocol entity enters the idle state until the next data transfer is requested.

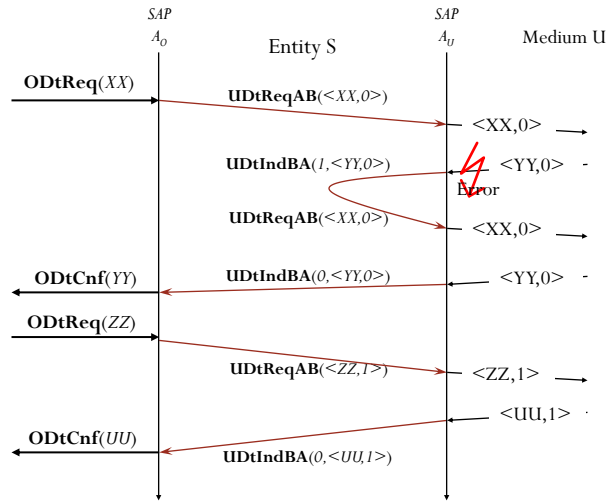
10

Modeling & Simulation - 3. Finite State Machines





Entity S Described in a Message Sequence Chart



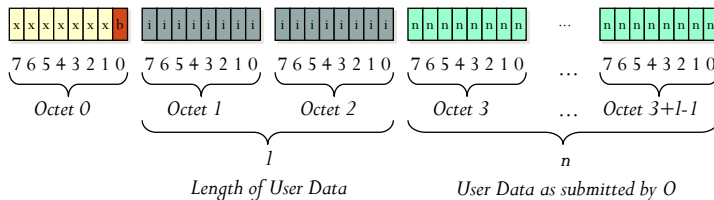
13

Modeling & Simulation - 3. Finite State Machines



Definition of PDUs

→ Example: PDU format $\langle n, b \rangle$



- Octet 0: contains sequence number b in the lowest bit
- Octet 1+2: codes the length of the user data
- Remaining Octets: user data n

14

Modeling & Simulation - 3. Finite State Machines



ASN.1

- ASN.1 (Abstract Syntax Notation 1) is a standard defined by ISO
 - ASN.1 allows
 - definition of data types
 - specification of values
 - Data types are subdivided into 4 classes:
 - **Universal:** Globally defined data types, e. g. Integer
 - **Application:** Data types that are defined in some other (application oriented) standard (e. g. FTAM, MHS)
 - **Private:** Data types which have been defined for some specific non-standard application
 - **Context-Specific:** Some shorter representation of data types that are valid only within a given (application) context

15

Modeling & Simulation - 3. Finite State Machines



ASN.1: Universal Data Types

- Primitive data types:
 - Boolean, Integer, Bit String, Octet String, IA5 String, ...
- Constructed data types:
 - *Sequence:* Ordered list of data types (cf. record in PASCAL)
 - *Set:* Unordered set of data types
 - *Sequence OF:* Ordered list of elements of the same data type (cf. array in PASCAL)
 - *Set OF:* Unordered set of elements of the same data type
 - *Choice:* Unordered set of data types, of which some may be chosen (cf. variant record in PASCAL)

Example:

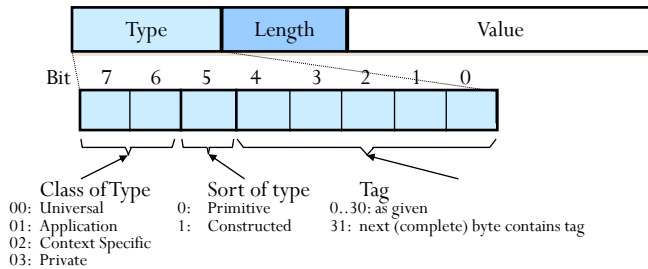
```
Staff_Member ::= Set { name IA5String,  
                      year_of_birth Integer,  
                      personnel_nr Integer }
```

16

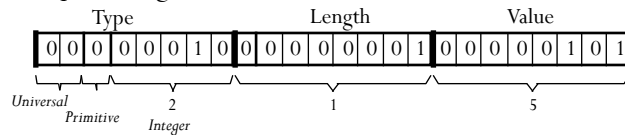
Modeling & Simulation - 3. Finite State Machines

ASN.1: Coding & Decoding

- Basic Encoding Rules, BER:



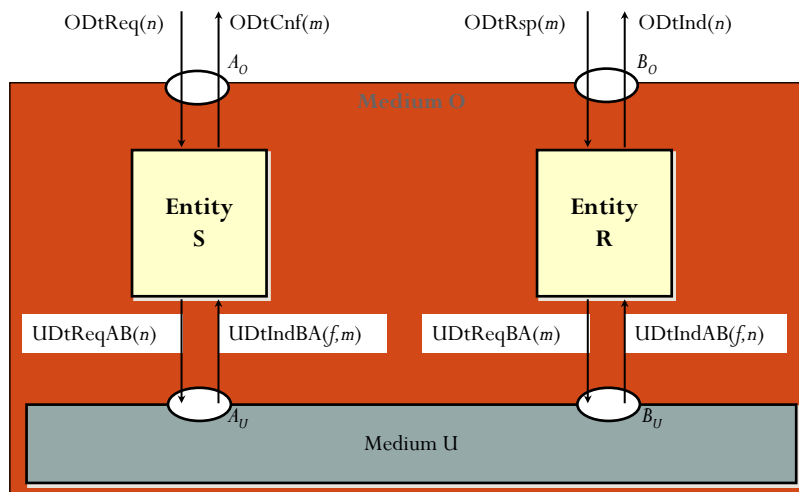
- Example: Integer Value 5



17

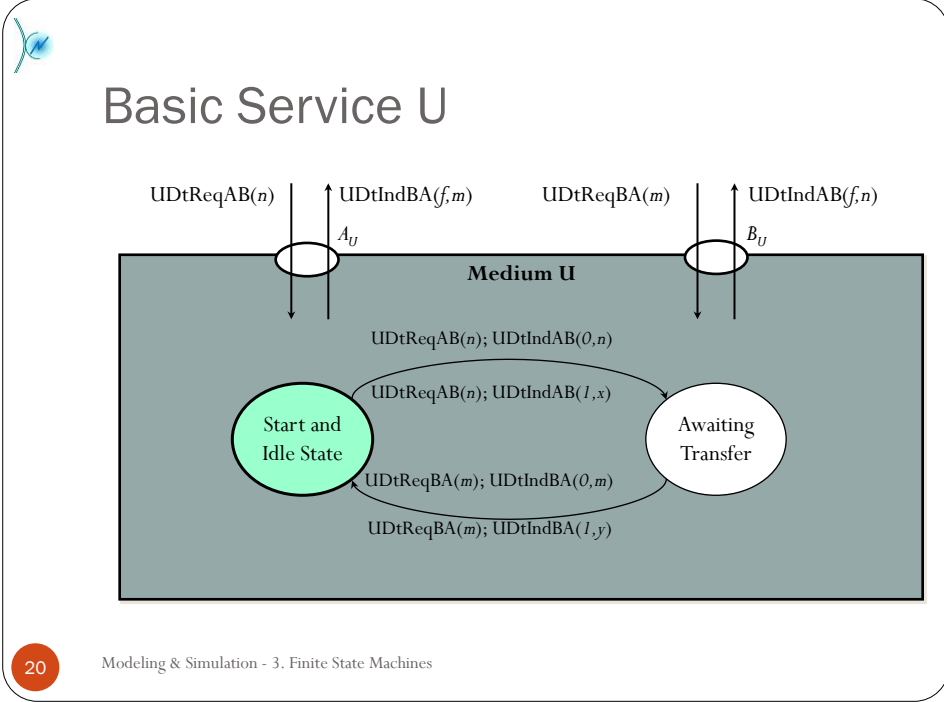
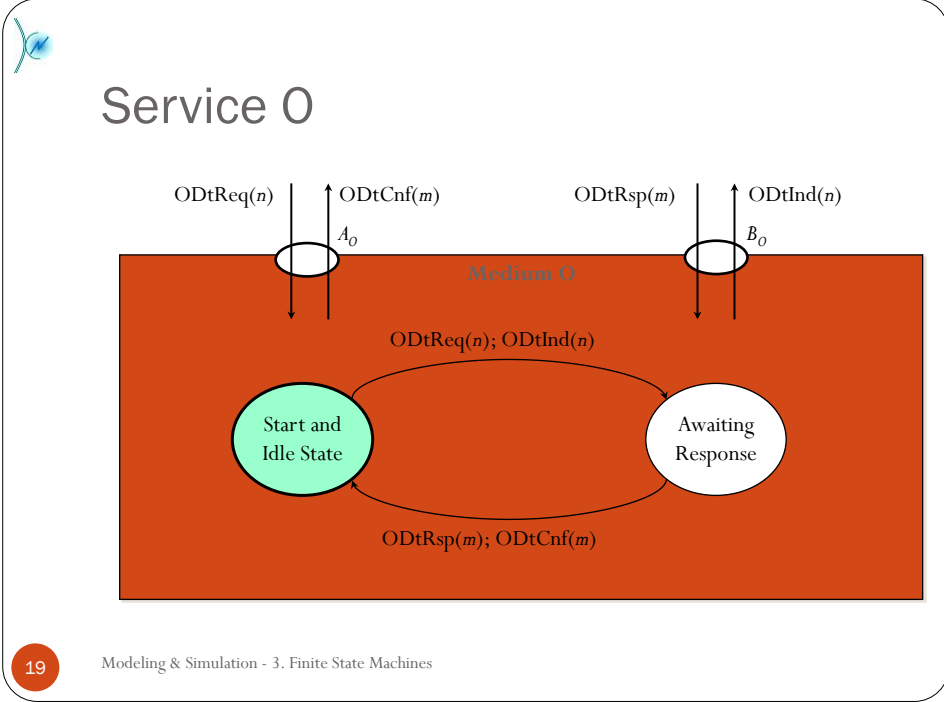
Modeling & Simulation - 3. Finite State Machines

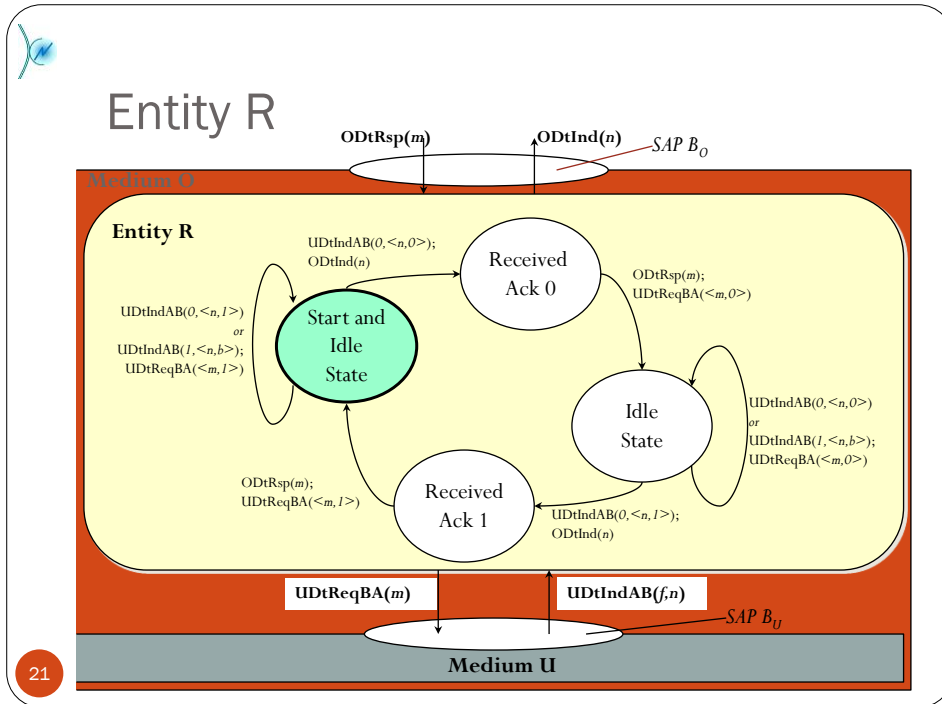
Complete Example



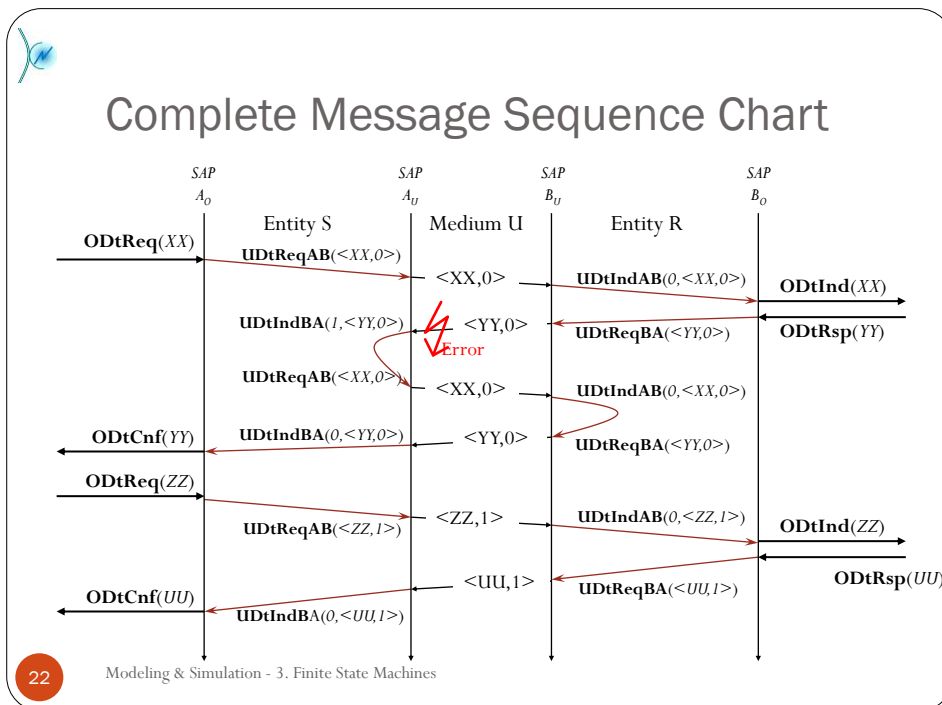
18

Modeling & Simulation - 3. Finite State Machines





21



22

Modeling & Simulation - 3. Finite State Machines



References

- Halsall, F. (1996). *Data Communications, Computer Networks, and Open Systems*. Harlow, England; Reading, Massachusetts; Menlo Park, California, Addison-Wesley.
- Lynch, W. (1968). "Reliable full-duplex file transmission over half-duplex telephone lines". *Communications of the ACM* 11 (6): 407–410.
- Popovic, M. (2006). *Communication Protocol Engineering*. Boca Raton, CRC Press Inc.
- Tanenbaum, A. S. (2002). *Computer Networks*. Upper Saddle River, New Jersey, Pearson Education.
- Venkataram, P. and S.S. Manvi (2004). *Communication Protocol Engineering*. Delhi, Prentice-Hall of India Pvt. Ltd.