



Prof. Dr.-Ing. habil. Lutz Winkler, Hochschule Mittweida (FH) – University of Applied Sciences, FB Informationstechnik & Elektrotechnik



<http://telecom.htwm.de>

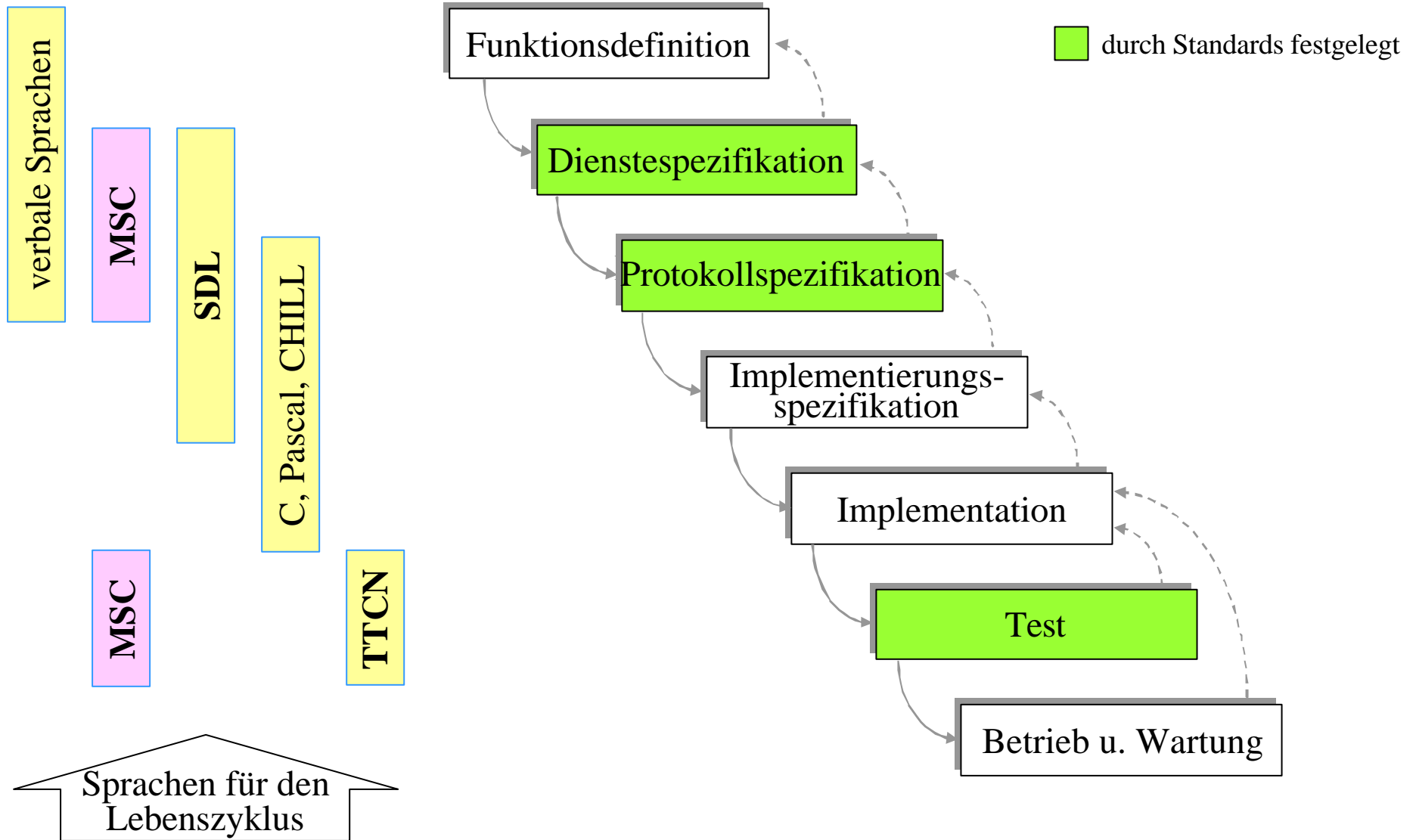
- ITU-T-Recommendation Z.120
- MSC soll in Verbindung mit den Sprachen SDL (Specification and Description Language) und TTCN (Tree and Tabular Combined Notation) den Entwurf, die Simulation, den Test und die Dokumentation von Kommunikationssoftware unterstützen.
- Mittels MSC kann man:
 - das beabsichtigte (Teil-) Verhalten eines verteilten Systems beschreiben,
 - das reale Verhalten bei der Simulation oder dem Test darstellen.
- MSC basiert auf der Darstellung des Nachrichtenaustausches zwischen verteilten Systemen bzw. zwischen Systemen und der Umgebung.
- Zwei Darstellungsformen:
 - MSC.GR, graphical representation,
 - MSC.PR, phrase representation.



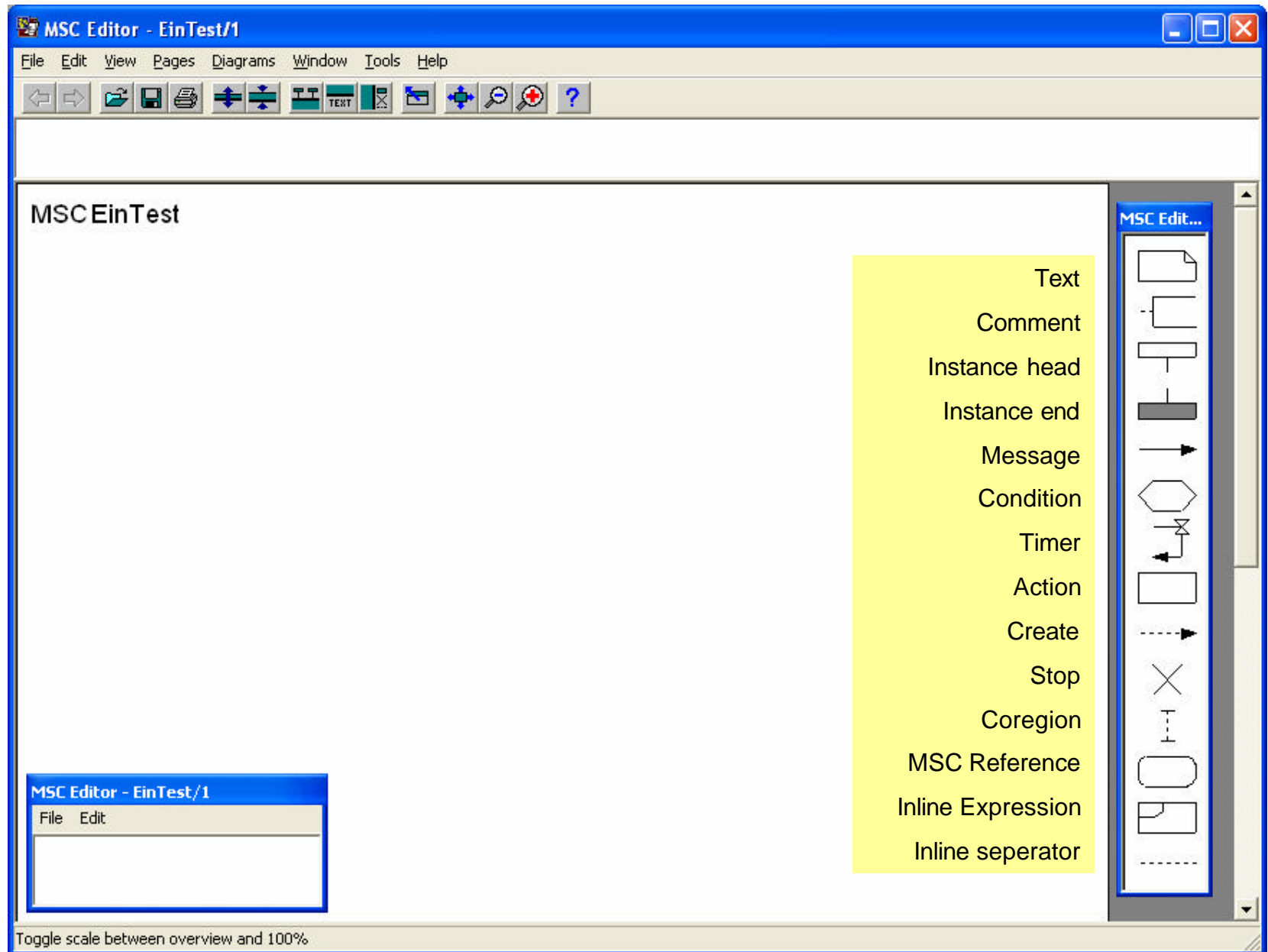
- Erste Ideen (Oktober 1989) ,
- Auf CCITT-Meeting in Helsinki wurde Entwicklung beschlossen (Juni 1990).
- Der erste Standard, Z.120, beschlossen auf CCITT-Meeting in Genf (Mai 1992).
- Überarbeitung, verabschiedet auf ITU-T-Meeting in Genf (April 1996).

- Während in der Z.120'92 nur lineare Abläufe beschrieben werden konnten, wurden in der Z.120'96 auch Ausdrucksmittel für Alternativen hinzugefügt.
- Des weiteren wurden einige Verbesserungen vorgenommen und eine HMSC (High level MSC) wurde kreiert.
- HMSC soll Beschreibungen auf sehr hohem Abstraktionsgrad unterstützen, so zum Beispiel Dienstabläufe. Die einzelnen Phasen der Dienste werden dann detaillierter durch MSC-Grundkonzepte untersetzt.

Wo erfolgt der Einsatz von MSC im Protokolllebenszyklus?



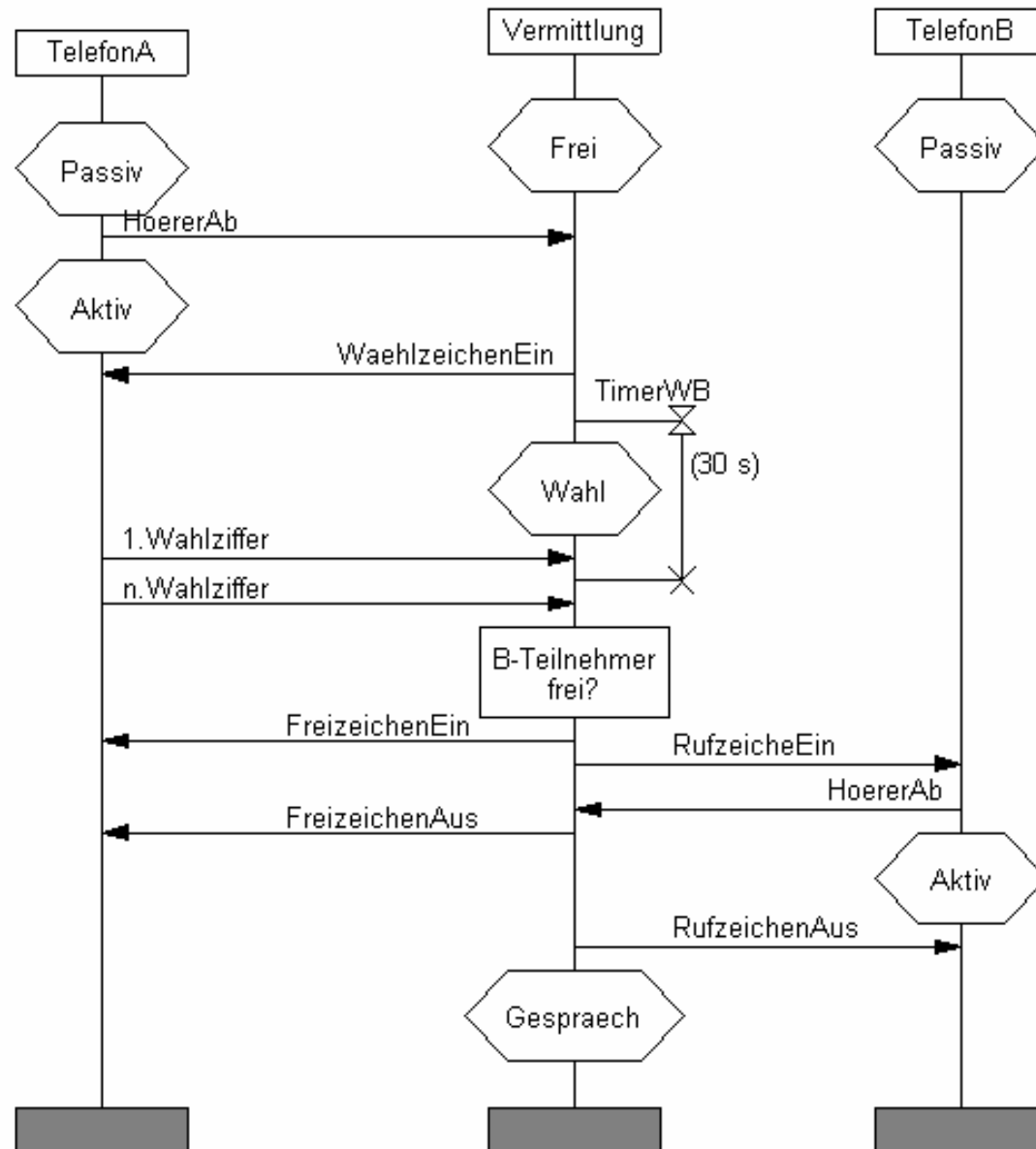
Ein MSC-Editor als Teil von SDT



Beispiel in MSC-GR: EinTelefonat



MSC EinTelefonat


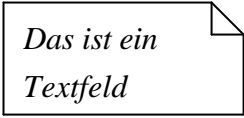
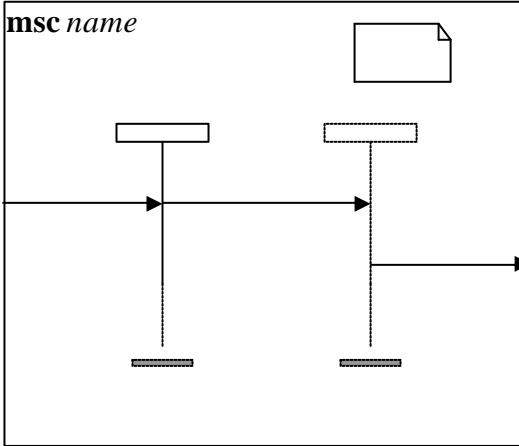






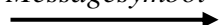

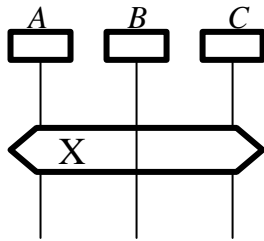
Beispiel in MSC-PR: EinTelefonat












```
mscdocument EinTelefonat;  
msc EinTelefonat;  
TelefonA: instance;  
Vermittlung: instance;  
TelefonB: instance;  
Vermittlung: condition Frei;  
TelefonB: condition Passiv;  
TelefonA: condition Passiv;  
TelefonA: out HoererAb,1 to Vermittlung;  
Vermittlung: in HoererAb,1 from TelefonA;  
TelefonA: condition Aktiv;  
Vermittlung: out WaehlzeichenEin,2 to TelefonA;  
TelefonA: in WaehlzeichenEin,2 from Vermittlung;  
Vermittlung: set TimerWB,Tmr1/* #SDTMSCPAR(30 s) */;  
Vermittlung: condition Wahl;  
TelefonA: out 1.Wahlziffer,3 to Vermittlung;  
Vermittlung: in 1.Wahlziffer,3 from TelefonA;  
Vermittlung: reset TimerWB,Tmr1;
```

```
TelefonA: out n.Wahlziffer,4 to Vermittlung;  
Vermittlung: in n.Wahlziffer,4 from TelefonA;  
Vermittlung: action 'B-Teilnehmer frei?';  
Vermittlung: out FreizeichenEin,5 to TelefonA;  
TelefonA: in FreizeichenEin,5 from Vermittlung;  
Vermittlung: out RufzeicheEin,6 to TelefonB;  
TelefonB: in RufzeicheEin,6 from Vermittlung;  
TelefonB: out HoererAb,7 to Vermittlung;  
Vermittlung: in HoererAb,7 from TelefonB;  
Vermittlung: out FreizeichenAus,8 to TelefonA;  
TelefonA: in FreizeichenAus,8 from Vermittlung;  
TelefonB: condition Aktiv;  
Vermittlung: out RufzeichenAus,9 to TelefonB;  
TelefonB: in RufzeichenAus,9 from Vermittlung;  
Vermittlung: condition Gesprach;  
TelefonA: endinstance;  
Vermittlung: endinstance;  
TelefonB: endinstance;  
endmsc;
```

Beschreibung	textuelle Darstellung	grafische Darstellung
Kommentare	<code><comment> ::=</code> comment <character string>	
Text	<code><text definition> ::=</code> text <character string>;	
Paging		
MSC Dokument	<code><msc document> ::=</code> <msc document head> <msc document body> <code><mscdocument head> ::=</code> mscdocument <msc document name>; <code><msc document body> ::=</code> { <message sequence chart> <msc diagramm> }*	
Message Sequence Chart	<code><message sequence chart> ::=</code> msc <msc head><msc body> endmsc ; <code><msc head> ::=</code> <msc name>[<msc interface>] <code><msc interface> ::=</code> inst <instance list>; <code><instance list> ::=</code> <instance name>[:<instance kind>][,<instance list>] <code><instance kind> ::=</code> [<kind denominator>]<identifier> <code><kind denominator> ::=</code> system block process service <code><msc body> ::=</code> { <instance definition> <text definition> }*	<p>Ein Rahmen grenzt ein MSC ab. Außerhalb des Rahmens ist Environment env.</p> 

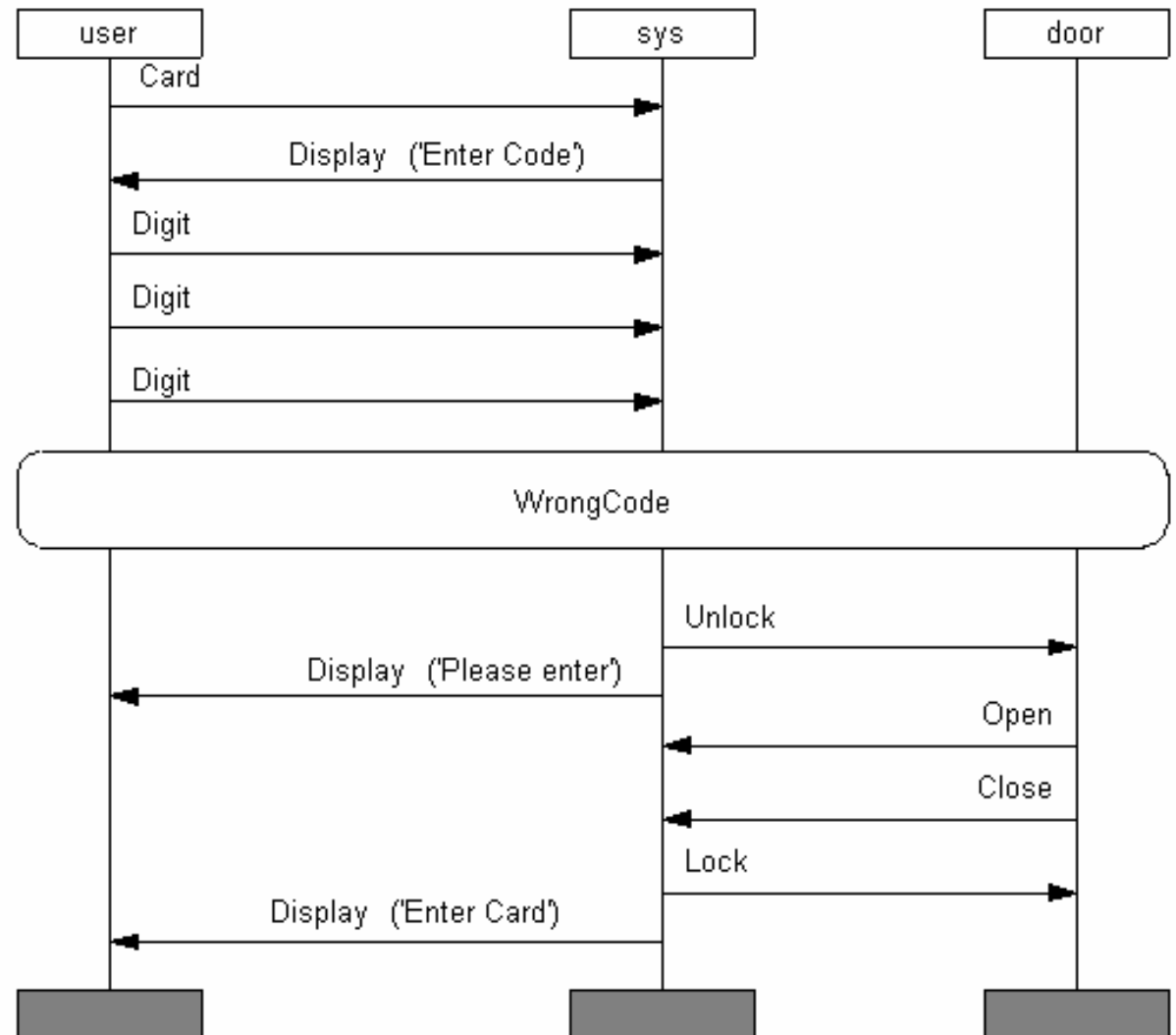
Beschreibung	textuelle Darstellung	grafische Darstellung
Instance	<p> <code><instance definition> ::=</code> instance <instance head> <instance body> endinstance; </p> <p> <code><instance head> ::=</code> <instance name> [[:]<instance kind>] [decomposed]; </p> <p> <code><instance body> ::=</code> <instance event list> [<stop>] </p> <p> <code><instance event list> ::=</code> {<message input> <message output> <create> <timer statement> <coregion> <action> <condition>}* </p>	<p> <i>Instanzkopf-symbol</i> -----  </p> <p> <i>Ende der Beschreibung</i> -----  </p> <p> <i>Instanzkopf-symbol</i> -----  </p> <p> <i>Ende der Beschreibung</i> -----  </p>
Message	<p> <code><message input> ::=</code> in<msg identification> from <address>; </p> <p> <code><message output> ::=</code> out<msg identification> to <address>; </p> <p> <code><msg identification> ::=</code> <message name> </p> <p> <code><address> ::=</code> <instance name> env </p>	<p> <i>Messagesymbol</i>  </p> <p> <i>Conditionsymbol</i>  </p>
Condition	<p> <code><condition> ::=</code> condition <condition name> [shared {<shared instance list> all }]; </p> <p> <code><shared instance list> ::=</code> <instance name> [, <shared instance list>] </p>	<p>  </p> <p> <i>Condition X gilt für A und C, aber nicht für B</i> </p>

Beschreibung	textuelle Darstellung	grafische Darstellung
Timer	$\langle \text{timer statement} \rangle ::= \langle \text{set} \rangle \langle \text{reset} \rangle \langle \text{timeout} \rangle$ $\langle \text{set} \rangle ::= \text{set} \langle \text{timer name} \rangle [, \langle \text{timer instance name} \rangle] [\langle \text{duration name} \rangle] ;$ $\langle \text{reset} \rangle ::= \text{reset} \langle \text{timer name} \rangle [, \langle \text{timer instance name} \rangle] ;$ $\langle \text{timeout} \rangle ::= \text{timeout} \langle \text{timer name} \rangle [, \langle \text{timer instance name} \rangle] ;$	<i>Set Timer Symbol</i>  <i>Reset Timer Symbols</i>  <i>Timeout Symbols</i> 
Action	$\langle \text{action} \rangle ::= \text{action} \langle \text{action text} \rangle ;$	<i>Action Symbol</i> 
Process creation	$\langle \text{create} \rangle ::=$ $\text{create} \langle \text{instance name} \rangle [(\langle \text{parameter list} \rangle)] ;$ $\langle \text{shared instance list} \rangle ::= \langle \text{instance name} \rangle [, \langle \text{shared instance list} \rangle]$	<i>Create Symbol</i> 
Process stop	$\langle \text{stop} \rangle ::= \text{stop} ;$	<i>Stop Symbol</i> 
Coregion	$\langle \text{coregion} \rangle ::= \text{concurrent} \{ \langle \text{coevent} \rangle \} * \text{endconcurrent} ;$ $\langle \text{coevent} \rangle ::= \langle \text{message input} \rangle \langle \text{message output} \rangle$	<i>Coregion Start</i>  <i>Coevent area</i>  <i>Coregion Stop</i> 

MSCEnter_Building

```

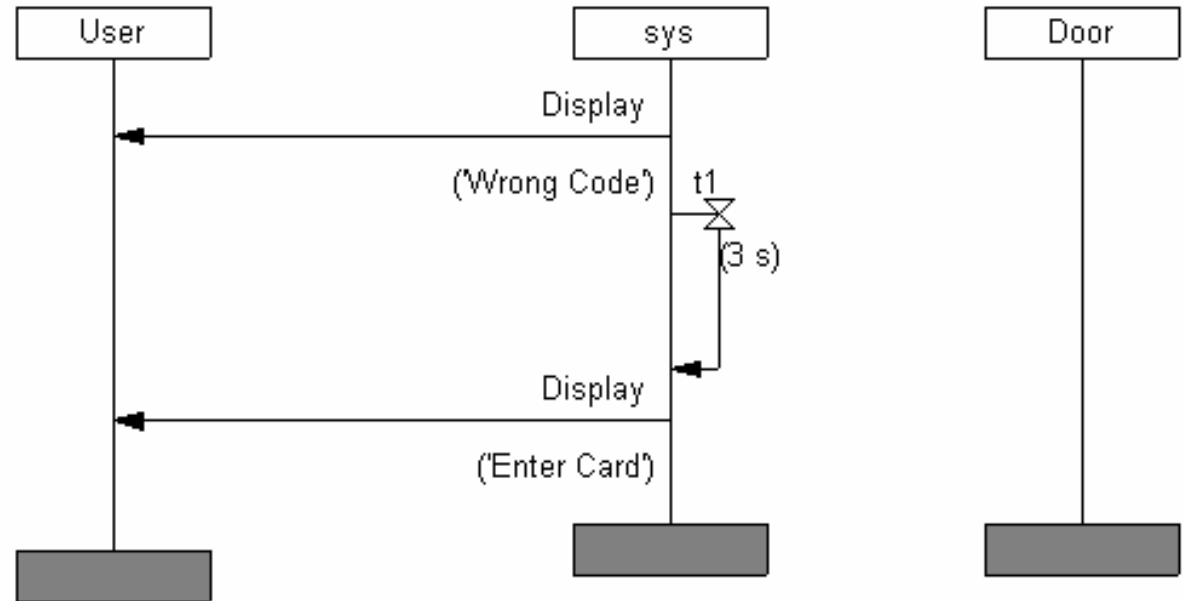
mscdocument Enter_Building;
msc Enter_Building;
user: instance;
sys: instance;
door: instance;
user: out Card,1 to sys;
sys: in Card,1 from user;
sys: out Display,2('Enter Code') to user;
user: in Display,2('Enter Code') from sys;
user: out Digit,3 to sys;
sys: in Digit,3 from user;
user: out Digit,4 to sys;
sys: in Digit,4 from user;
user: out Digit,5 to sys;
sys: in Digit,5 from user;
all: reference WrongCode;
sys: out Unlock,6 to door;
door: in Unlock,6 from sys;
sys: out Display,7('Please enter') to user;
user: in Display,7('Please enter') from sys;
door: out Open,8 to sys;
sys: in Open,8 from door;
door: out Close,9 to sys;
sys: in Close,9 from door;
sys: out Lock,10 to door;
door: in Lock,10 from sys;
sys: out Display,11('Enter Card') to user;
user: in Display,11('Enter Card') from sys;
user: endinstance;
sys: endinstance;
door: endinstance;
endmsc;
    
```





MSC WrongCode

```
mscdocument WrongCode;  
msc WrongCode;  
User: instance;  
sys: instance;  
Door: instance;  
sys: out Display,1('Wrong Code') to User;  
User: in Display,1('Wrong Code') from sys;  
sys: set t1,Tmr1/* #SDTMSCPAR(3 s) */;  
sys: timeout t1,Tmr1;  
sys: out Display,2('Enter Card') to User;  
User: in Display,2('Enter Card') from sys;  
sys: endinstance;  
Door: endinstance;  
User: endinstance;  
endmsc;
```

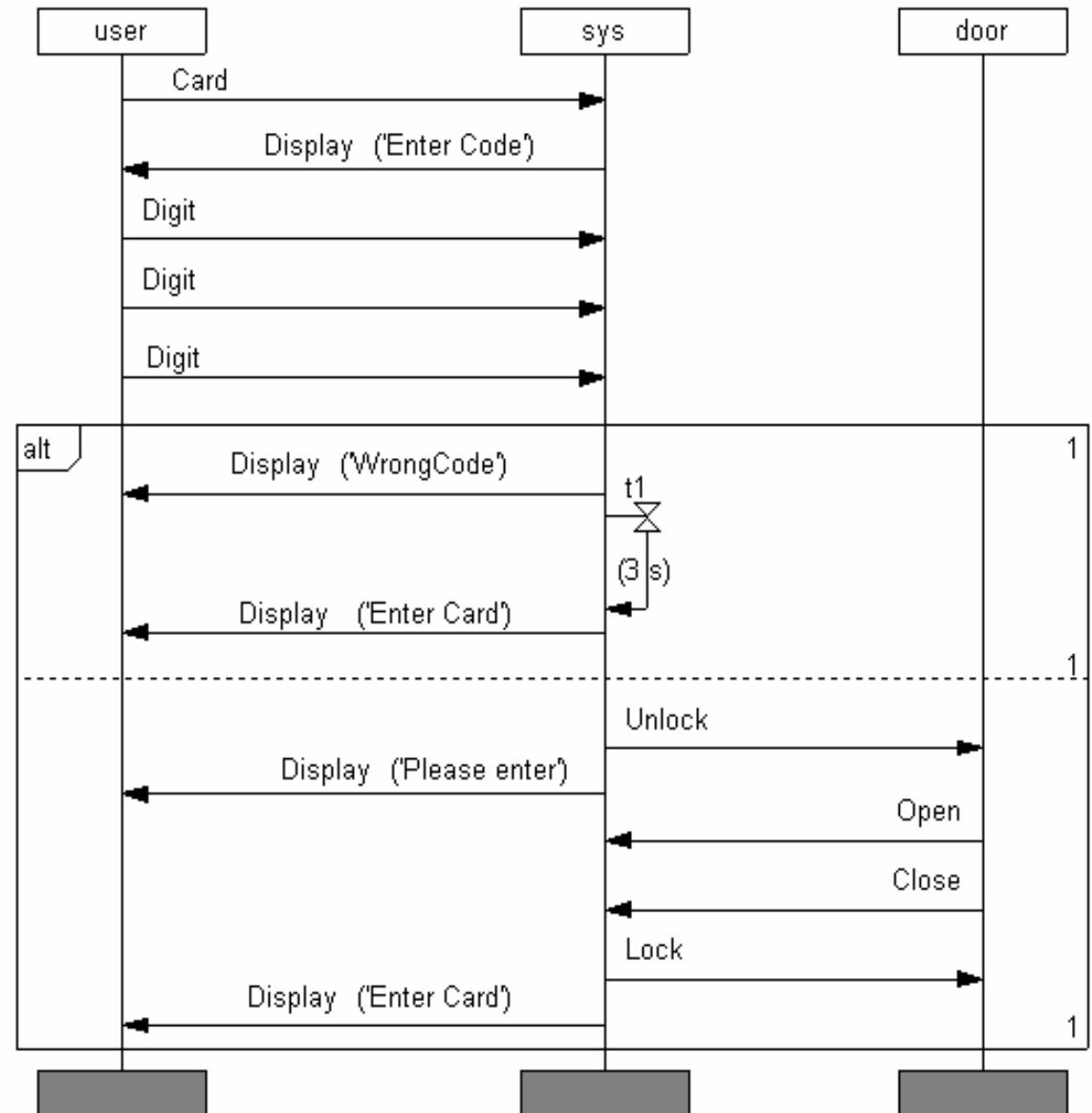




MSCEnter_Building1

```

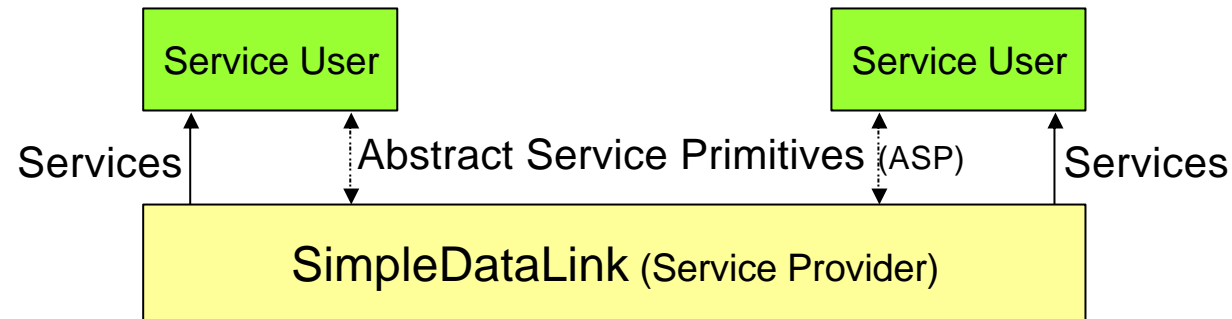
mscdocument Enter_Building1;
mhc Enter_Building1;
user: instance;
sys: instance;
door: instance;
user: out Card,1 to sys;
sys: in Card,1 from user;
sys: out Display,2('Enter Code') to user;
user: in Display,2('Enter Code') from sys;
user: out Digit,3 to sys;
sys: in Digit,3 from user;
user: out Digit,4 to sys;
sys: in Digit,4 from user;
user: out Digit,5 to sys;
sys: in Digit,5 from user;
all: alt begin;
sys: out Display,6('WrongCode') to user;
user: in Display,6('WrongCode') from sys;
sys: set t1,Tmr1/* #SDTMSCPAR(3 s) */;
sys: timeout t1,Tmr1;
sys: out Display,7('Enter Card') to user;
user: in Display,7('Enter Card') from sys;
alt;
sys: out Unlock,8 to door;
door: in Unlock,8 from sys;
sys: out Display,9('Please enter') to user;
user: in Display,9('Please enter') from sys;
door: out Open,10 to sys;
sys: in Open,10 from door;
door: out Close,11 to sys;
sys: in Close,11 from door;
sys: out Lock,12 to door;
door: in Lock,12 from sys;
sys: out Display,13('Enter Card') to user;
user: in Display,13('Enter Card') from sys;
alt end;
user: endinstance;
sys: endinstance;
door: endinstance;
endmhc;
    
```





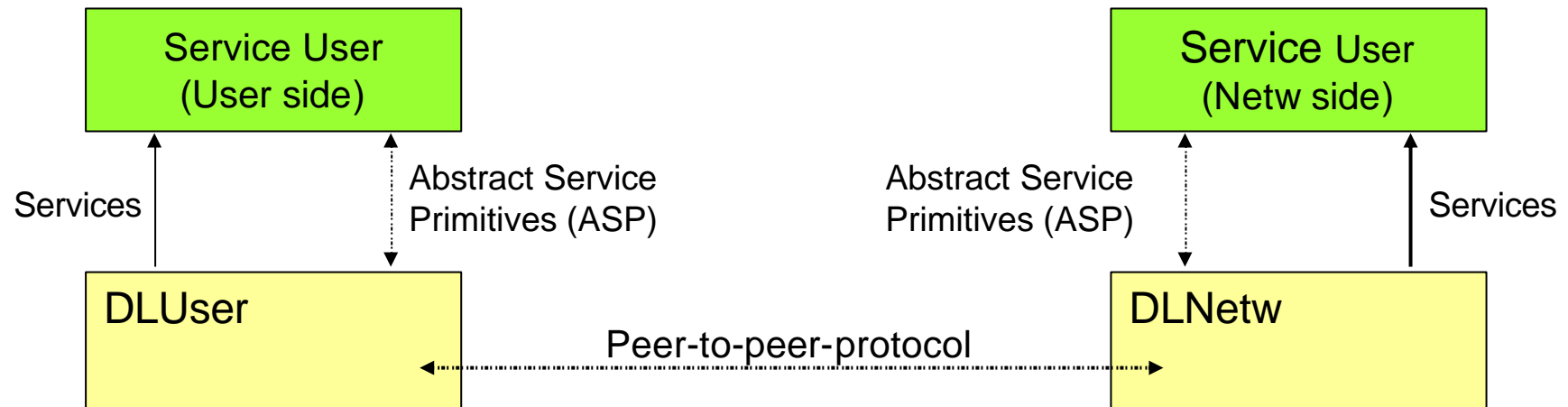
- Anwendung anhand eine einfachen Schicht-2-Dienstes.
- Bei der Definition dieser einfachen Sicherungsschicht (SimpleDataLink) sollen die Grundprinzipien einer ISDN-Schicht-2 benutzt werden. Dies gilt für die Bezeichnung der Dienste, der abstrakten Primitive und der benutzten Protokoll-Daten-Einheiten, nicht aber für die Funktionalität.
- SimpleDataLink soll den Dienstnutzern folgende Funktionen bereit- bzw. nicht bereitstellen:
 - einen bestätigten Auf- und Abbau einer Verbindung,
 - quittierte Datenübertragung,
 - unquitierte Datenübertragung,
 - Adressierungs- und Fehlererkennungsmechanismen werden nicht unterstützt.
- In einem nächsten Schritt muß man jetzt für jeden Dienst abstrakte Dienstprimitive (Abstract Service Primitives) und Parameter festlegen.

Anwendung von MSC'93 anhand eines einfachen Protokolls: Dienstspezifikation für eine SimpleDataLink



Service (Dienst)	ASP's	Parameter
Anforderung zum Aufbau einer Verbindung	dlEstablishRq	keine
Anzeige, Verbindung, initiiert vom Partner, aufgebaut	dlEstablishIn	keine
Bestätigung, Verbindung wurde aufgebaut	dlEstablishCf	keine
Anforderung zur quittierten Datenübertragung	dlDataRq	Daten
Anzeige von quittiert empfangenen Daten	dlDataIn	Daten
Anforderung zur unquitierten Datenübertragung	dlUnitDataRq	Daten
Anzeige von unquitiert empfangenen Daten	dlUnitDataIn	Daten
Anforderung zum Abbau einer Verbindung	dlReleaseRq	keine
Anzeige, Verbindung, initiiert vom Partner, abgebaut	dlReleaseIn	keine
Bestätigung, Verbindung wurde abgebaut	dlReleaseCf	keine

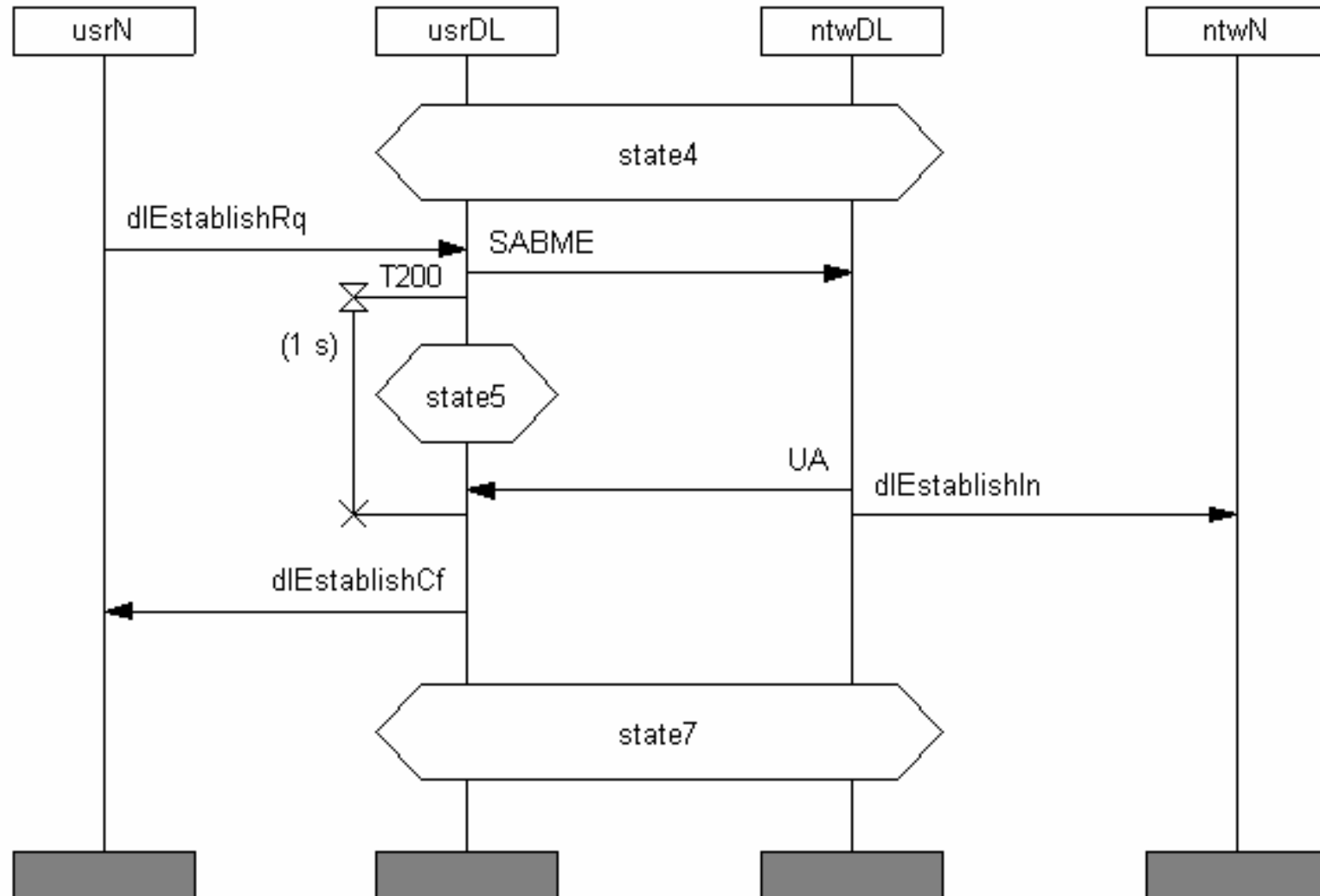
Anwendung von MSC'93 anhand eines einfachen Protokolls: Protokollspezifikation



*Zwischen den paaren Instanzen findet ein virtueller Austausch von Protocol data units (PDU's) nach feststehenden Regeln statt. Die Definition von PDU's und der Regeln des Austausches selbiger bezeichnet man als **Protokoll***

PDU	Bedeutung	Parameter
SABME	Gehe in den Zustand 7, damit quittierte Datenübertragung möglich ist	keine
DISC	Gehe in den Zustand 4 (nur noch unquitierte Datenübertragung möglich)	keine
UA	Positive Quittung als Antwort auf SABME oder DISC	keine
DM	Negative Quittung als Antwort auf SABME	keine
I	Informationsrahmen zur quittierten Datenübertragung	Daten
UI	Unnumerierter Informationsrahmen zur unquitierten Datenübertragung	Daten
RR	Quittierung eines empfangenen Informationsrahmens	keine

MSC SimpleDataLinkEstablish



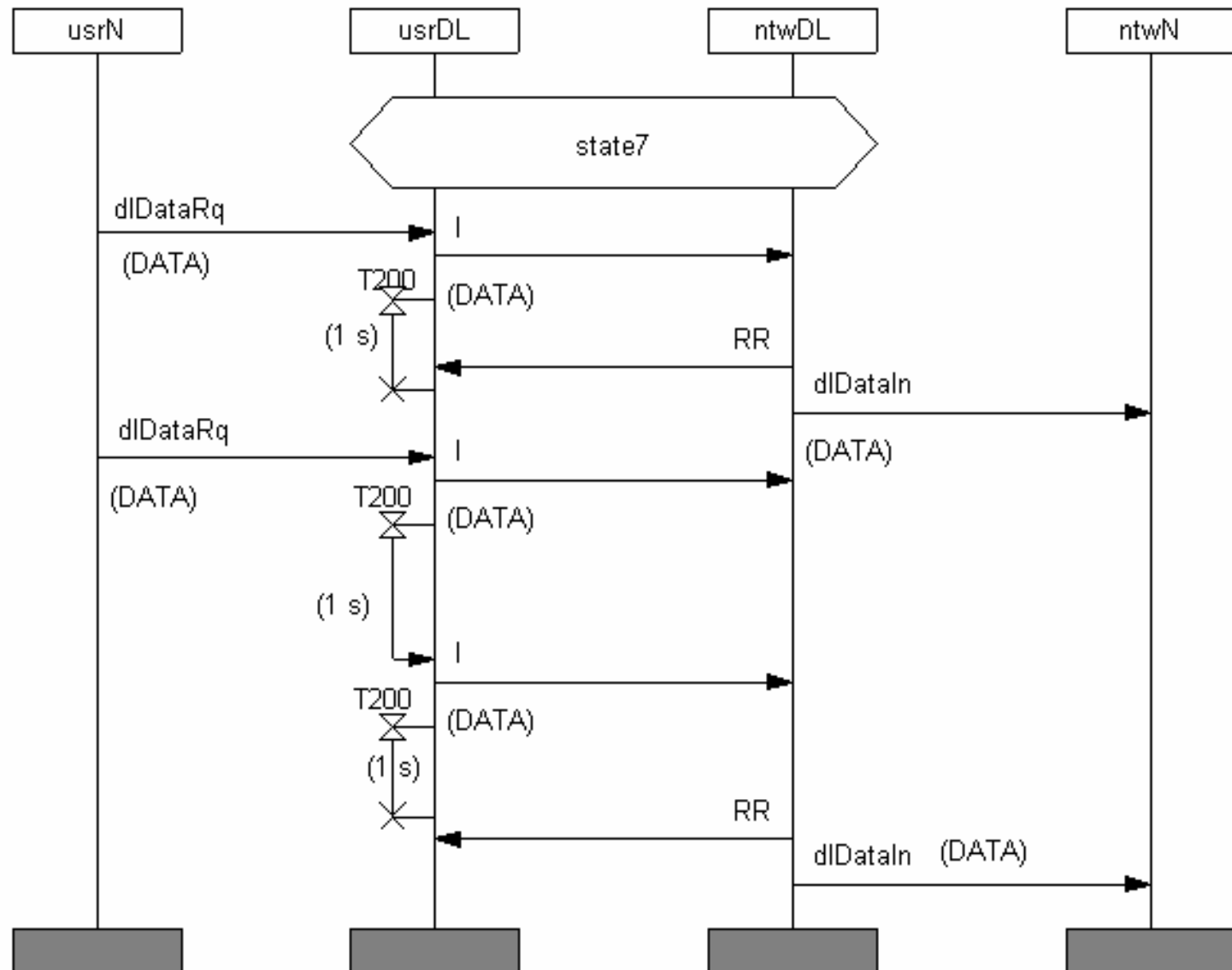
SimpleDataLink: erfolgreicher Verbindungsaufbau (MSC-PR)



```
mscdocument SimpleDataLinkEstablish;
msc SimpleDataLinkEstablish;
usrN: instance;
usrDL: instance;
ntwDL: instance;
ntwN: instance;
usrDL,ntwDL: condition state4;
usrN: out dlEstablishRq,1 to usrDL;
usrDL: in dlEstablishRq,1 from usrN;
usrDL: out SABME,2 to ntwDL;
ntwDL: in SABME,2 from usrDL;
usrDL: set T200,Tmr1/* #SDTMSCPAR(1 s) */;
usrDL: condition state5;
ntwDL: out UA,3 to usrDL;
usrDL: in UA,3 from ntwDL;
usrDL: reset T200,Tmr1;
ntwDL: out dlEstablishIn,4 to ntwN;
ntwN: in dlEstablishIn,4 from ntwDL;
usrDL: out dlEstablishCf,5 to usrN;
usrN: in dlEstablishCf,5 from usrDL;
usrDL,ntwDL: condition state7;
usrN: endinstance;
usrDL: endinstance;
ntwDL: endinstance;
ntwN: endinstance;
endmsc;
```



MSC SimpleDataLinkDataTransfer



SimpleDataLink: Datenübertragung (MSC-PR)



```
mscdocument SimpleDataLinkDataTransfer;
msc SimpleDataLinkDataTransfer;
usrN: instance;
ntwN: instance;
ntwDL: instance;
usrDL: instance;
usrDL,
ntwDL: condition state7;
usrN: out dlDataRq,1(DATA) to usrDL;
usrDL: in dlDataRq,1(DATA) from usrN;
usrDL: out I,2(DATA) to ntwDL;
ntwDL: in I,2(DATA) from usrDL;
usrDL: set T200,Tmr1/* #SDTMSCPAR(1 s) */;
ntwDL: out RR,3 to usrDL;
usrDL: in RR,3 from ntwDL;
usrDL: reset T200,Tmr1;
ntwDL: out dlDataIn,4(DATA) to ntwN;
ntwN: in dlDataIn,4(DATA) from ntwDL;
usrN: out dlDataRq,5(DATA) to usrDL;
usrDL: in dlDataRq,5(DATA) from usrN;
usrDL: out I,6(DATA) to ntwDL;
ntwDL: in I,6(DATA) from usrDL;
usrDL: set T200,Tmr2/* #SDTMSCPAR(1 s) */;
usrDL: timeout T200,Tmr2;
usrDL: out I,7(DATA) to ntwDL;
ntwDL: in I,7(DATA) from usrDL;
usrDL: set T200,Tmr3/* #SDTMSCPAR(1 s) */;
usrDL: reset T200,Tmr3;
ntwDL: out RR,8 to usrDL;
usrDL: in RR,8 from ntwDL;
ntwDL: out dlDataIn,9(DATA) to ntwN;
ntwN: in dlDataIn,9(DATA) from ntwDL;
usrN: endinstance;
usrDL: endinstance;
ntwDL: endinstance;
ntwN: endinstance;
endmsc;
```



- Editieren Sie folgende MSC's unter Nutzung des Standards Q.921
 - Unnummerierte Datenübertragung (S. 88)
 - Verbindungsaufbau wird vom Partner abgelehnt (S. 63 state 4, S. 67 state 5)
 - Verbindungsabbau (S. 70 state 7)