

Protocol Technology



SDL and INRES protocol

Gusztáv Adamis

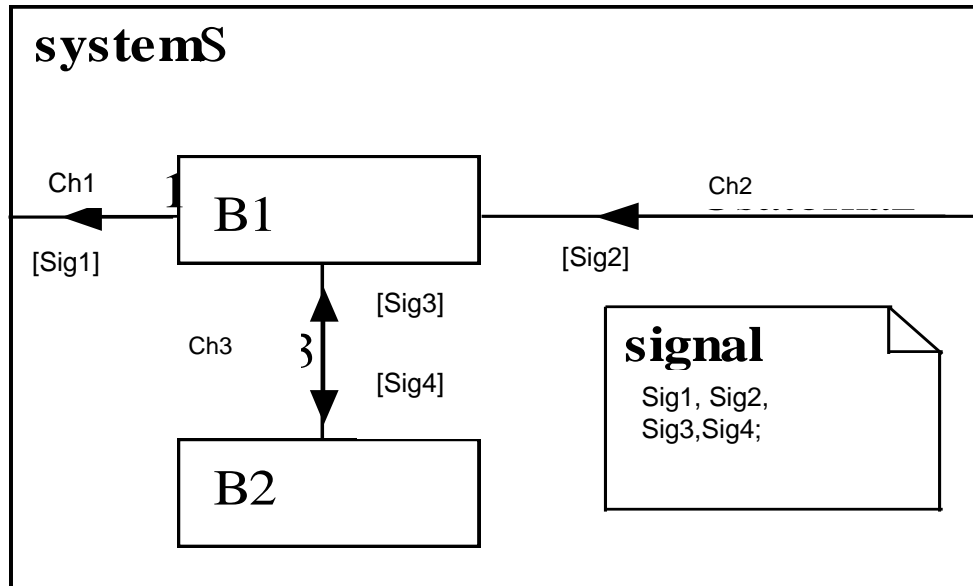
BME TMIT

2016

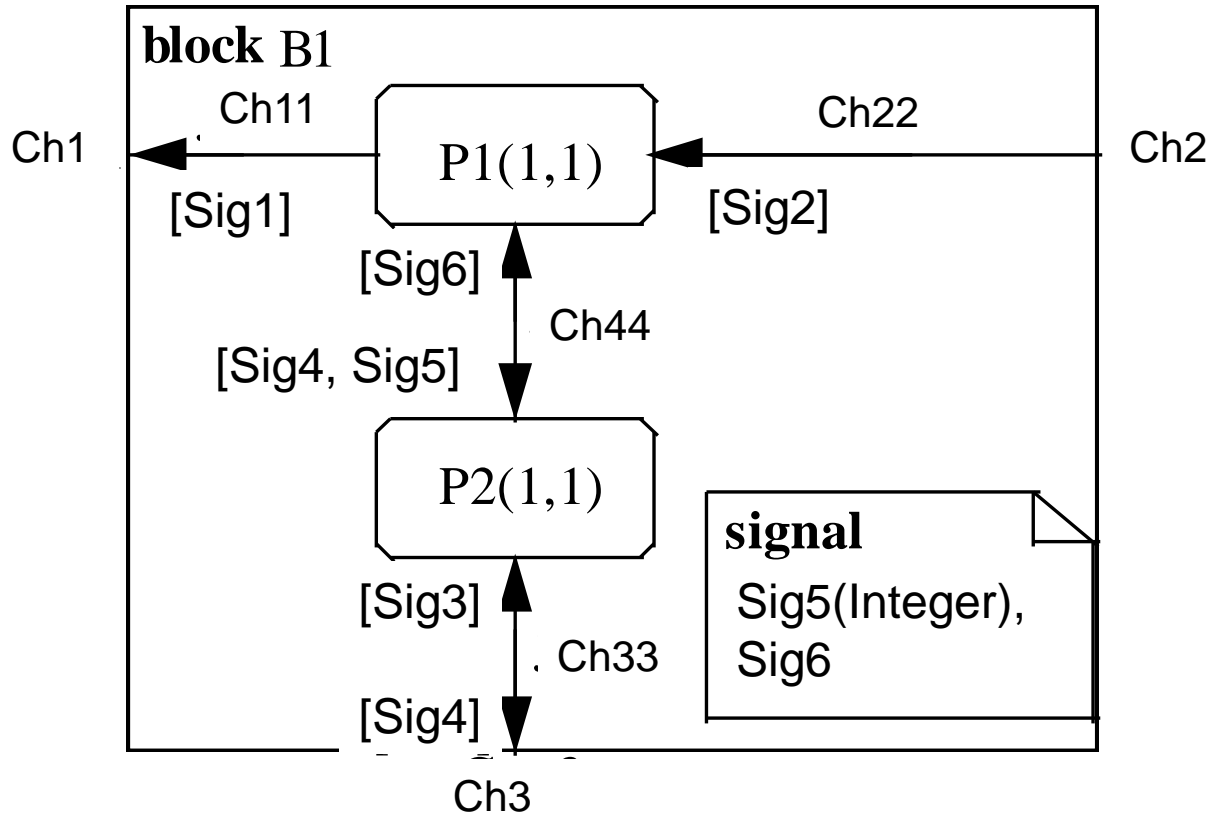
SDL

- Specification and Description Language
 - Standardised by ITU
 - SDL/PR and SDL/GR
 - Background: Communicating EFSM
 - Hierarchical description
 - system
 - everything else is the environment
 - block
 - Communication:
 - channels between system and environment
 - channels among blocks
 - signals
 - process
 - EFSM

system

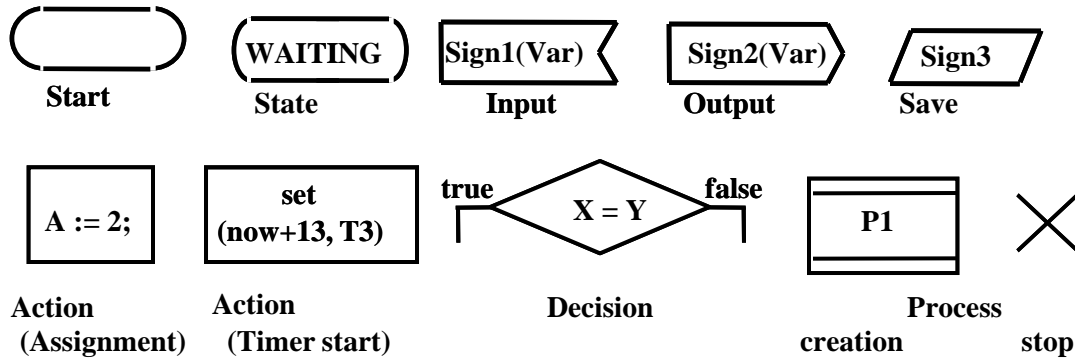


block



- Connecting channels at border
- Signal with parameter
- Process (x,y):
 - x: max. number
 - y: number at init

process

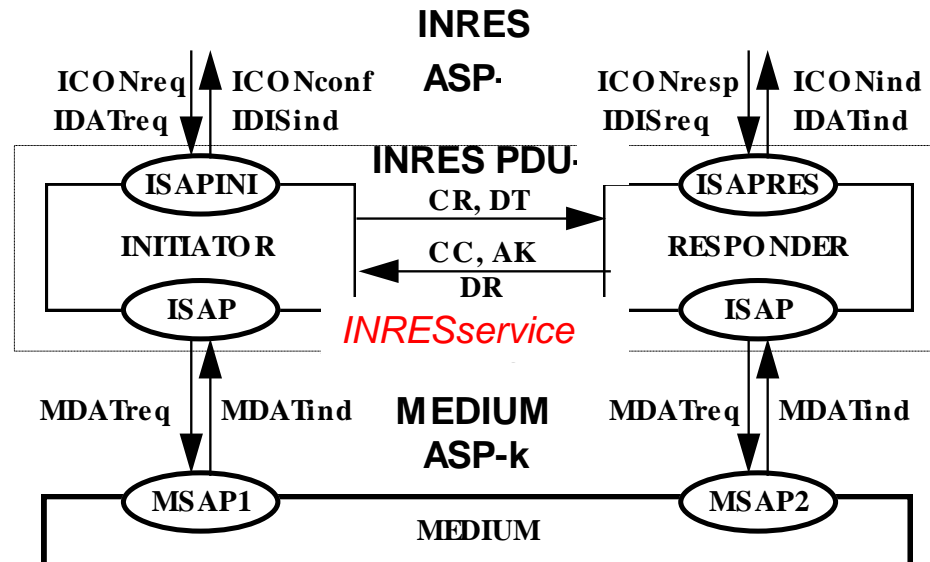


Processes

- Static/Dynamic
- Has ONE input queue (FIFO)
- State transition/Implicit state transition
- Timer operations
 - start – now +
 - reset
 - timeout: timer name as a message to the end of the queue of the starting process
 - message sending: if the destination is unambiguous: TO process/VIA channel
 - Process variables: SELF, PARENT, OFFSPRING, SENDER

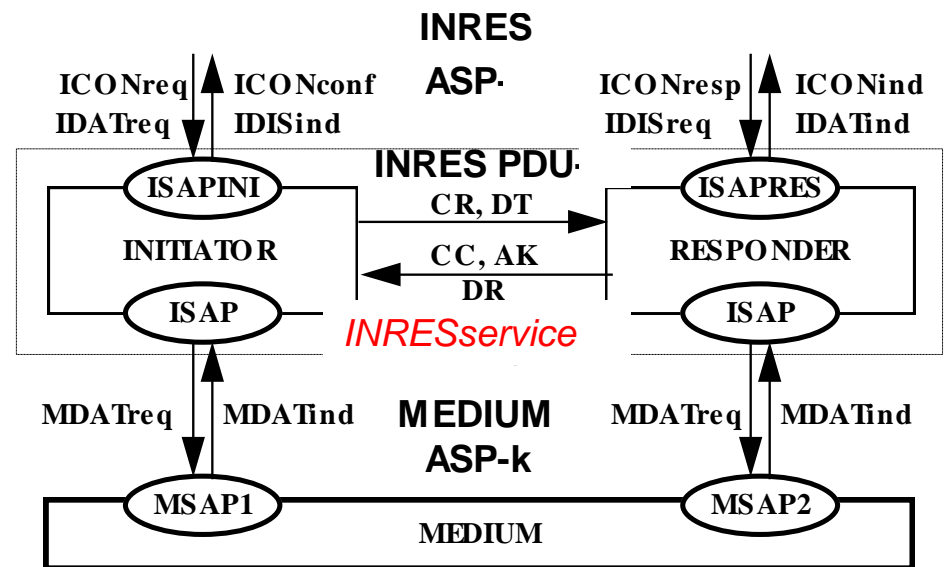
INRES

- Initiator/Responder
- Not real protocol, but contains typical protocol mechanisms
- For testing purposes
- Asymmetric

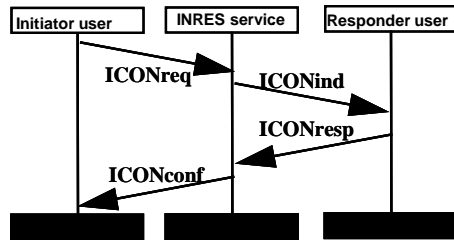


Primitives

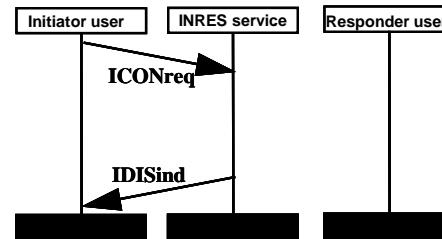
- Request, Indication
- Response, Confirmation
- In INRES [Primitive(parameter)]
 - ICONreq
 - ICONind
 - ICONresp
 - ICONconf
 - IDATreq(ISDU)
 - IDATind(ISDU)
 - IDISreq
 - IDISind



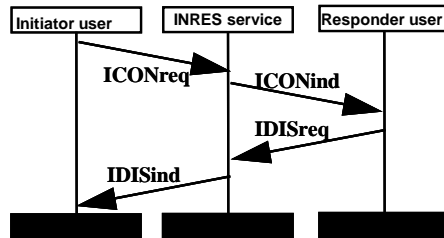
INRES – connection establishment



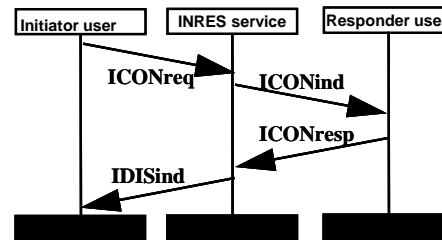
A.) Successful conn. establishment



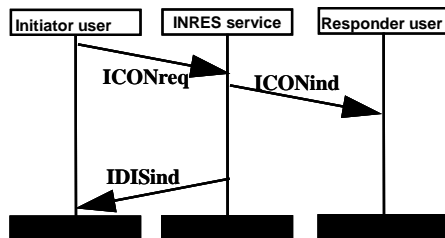
C.) Non-Successful conn. establishment
(Error in transmission of Request)



B.) Non-Successful conn. establishment
(Reject)

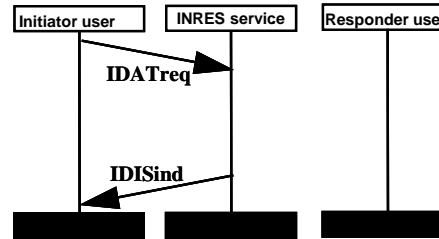
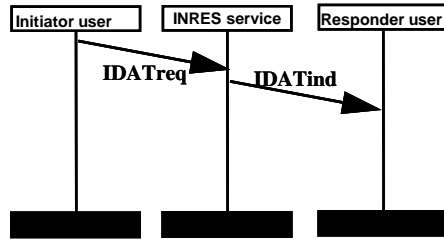


D.) Non-Successful conn. establishment
(Error in transmission of Response)

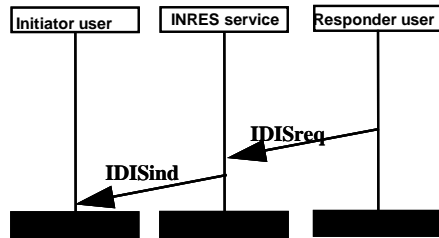


E.) Non-Successful conn. establishment
(Responder user does not react for Request)

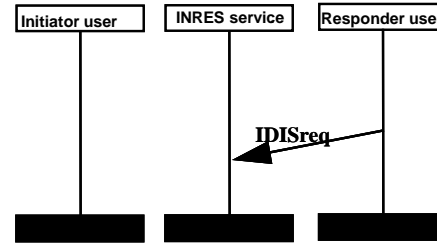
INRES – data transfer



INRES – disconnect



A.) Successful disconnection

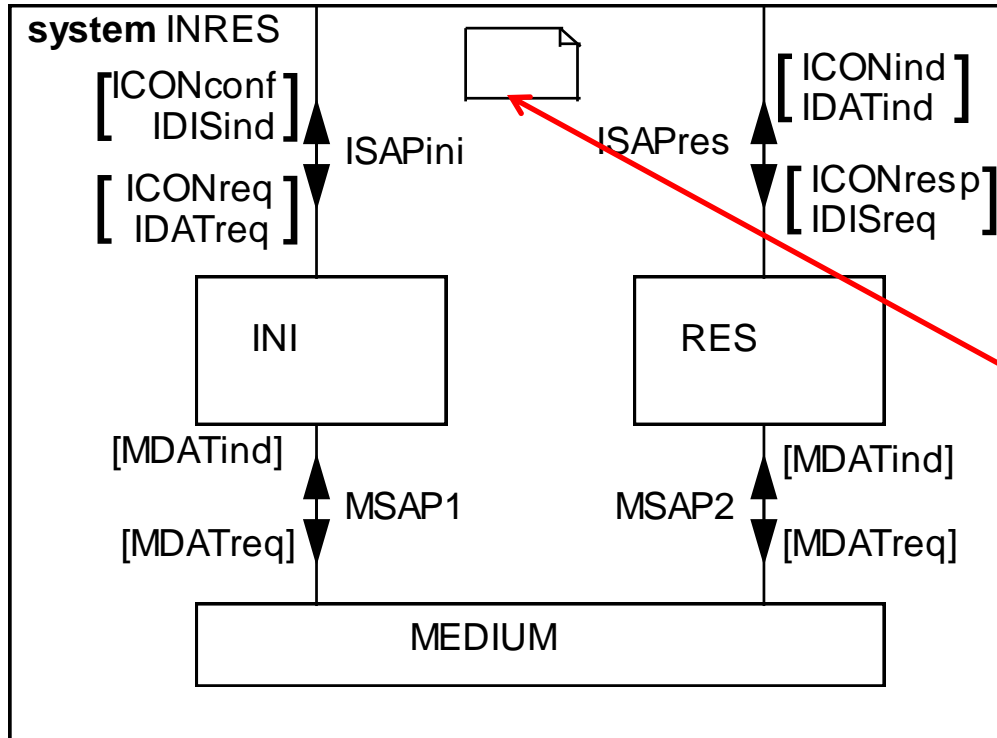


B.) Non-successful disconnection
(Error in transmission of Request)

INRES PDUs

| PDU | Meaning | Parameters | Related INRES ASPs |
|-----|-------------------------|----------------------------------|----------------------|
| CR | Connection Request | | ICONreq ICONind |
| CC | Connection Confirmation | | ICONresp ICONconf |
| DT | Data Transfer | Sequence number + Data (ISDU) | IDATreq IDATind |
| AK | Acknowledgement | Sequence number | - |
| DR | Disconnection Request | | IDISreq IDISind |

INRES system



Here: Data definition
(Next slide)

Data type definition

signal

**ICONreq, IDATreq(ISDUType),
ICONconf, ICONind, ICONresp, IDISreq,
IDISind, IDATind(ISDUType),
MDATreq(MSDUType), MDATind(MSDUType);**

newtype Sequencenumber

literals 0, 1;

operators succ: Sequencenumber -> Sequencenumber:

axioms succ (0) == 1;

succ (1) == 0

endnewtype Sequencenumber;

newtype ISDUType

/* itt az ISDU adatszerkezetét kell definiálni */

endnewtype ISDUType;

newtype IPDUType

literals CR, CC, DR, DT, AK;

endnewtype IPDUType;

newtype MSDUType

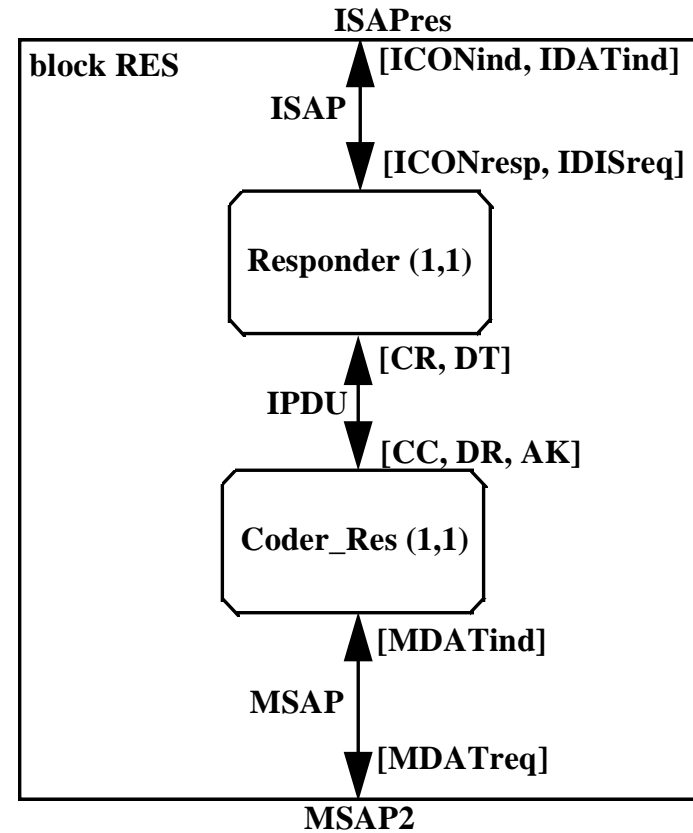
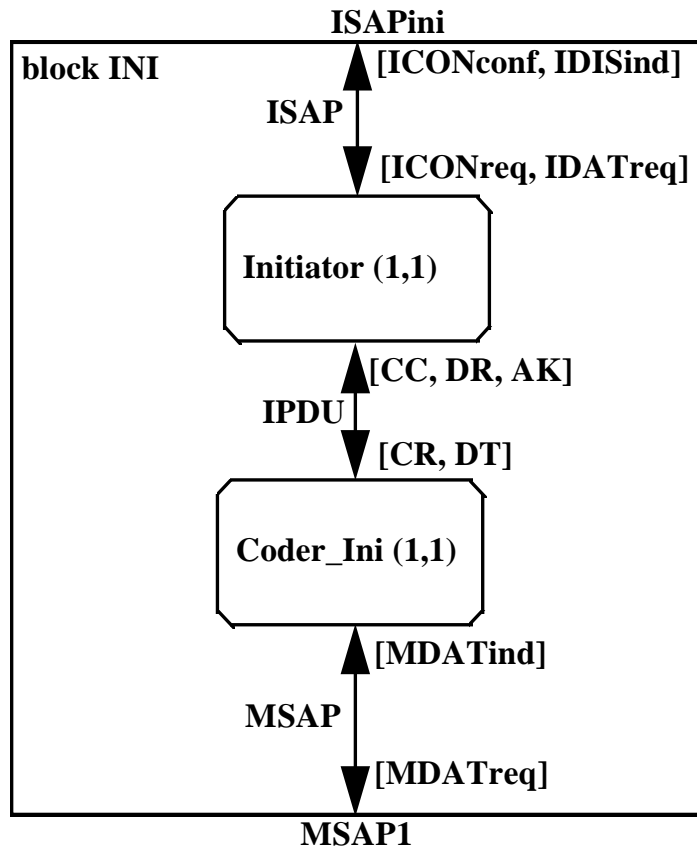
struct id IPDUType;

num Sequencenumber;

data ISDUType;

endnewtype MSDUType;

Block INI and RES

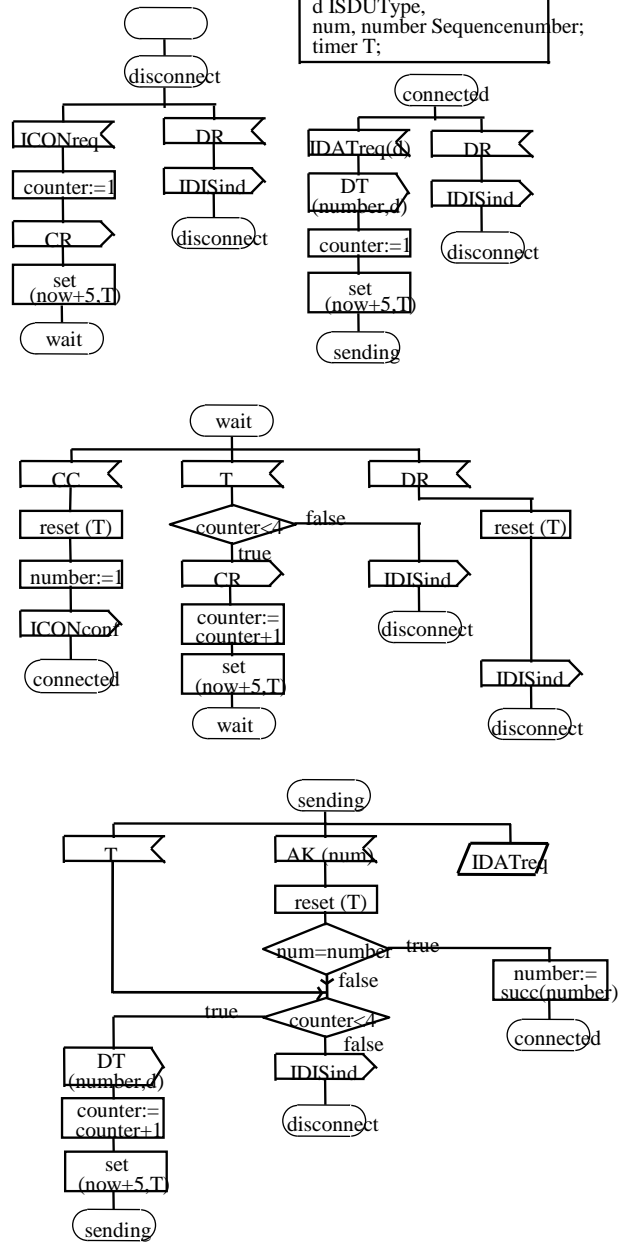


Initiator

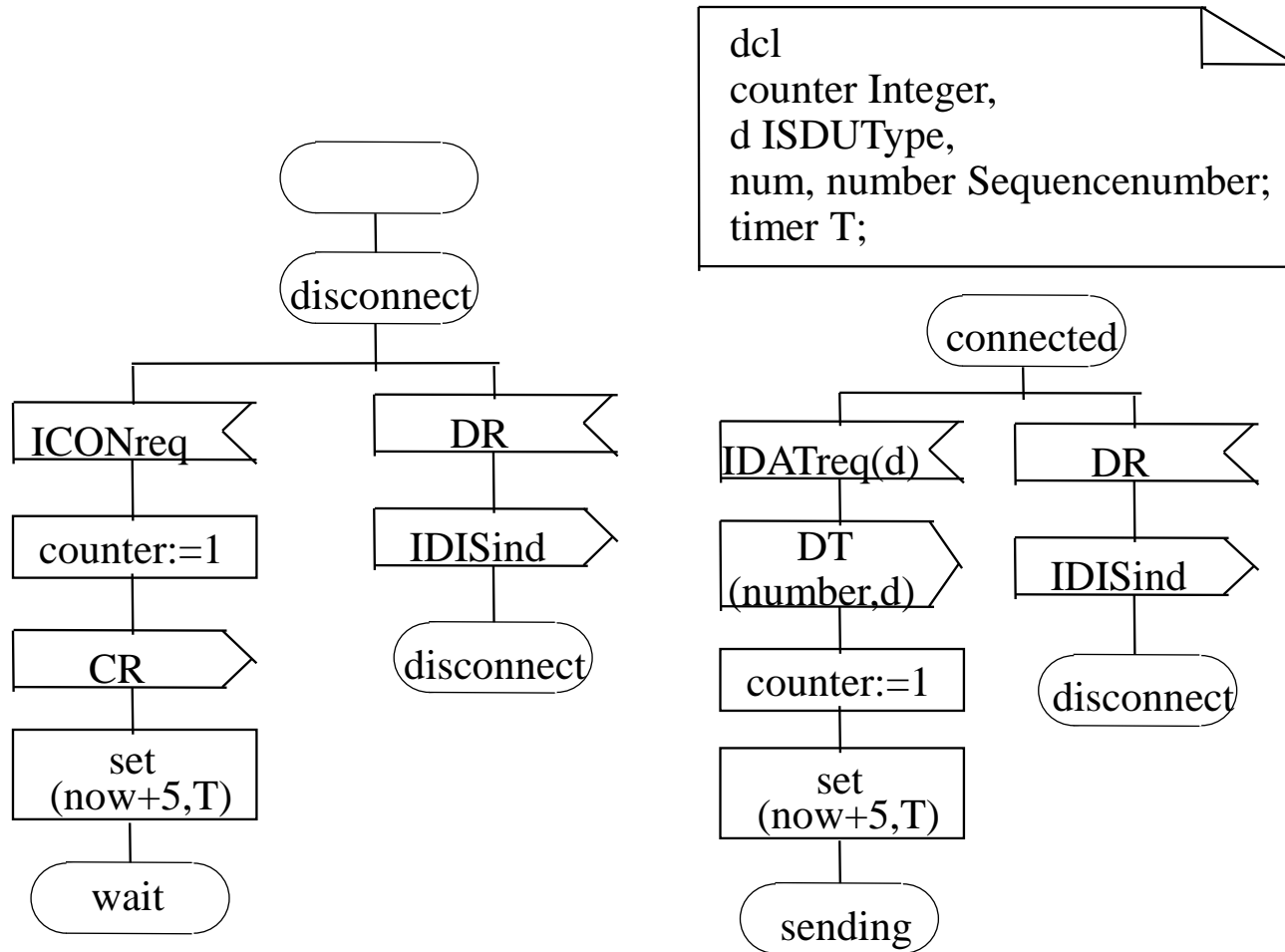
process Initiator

```

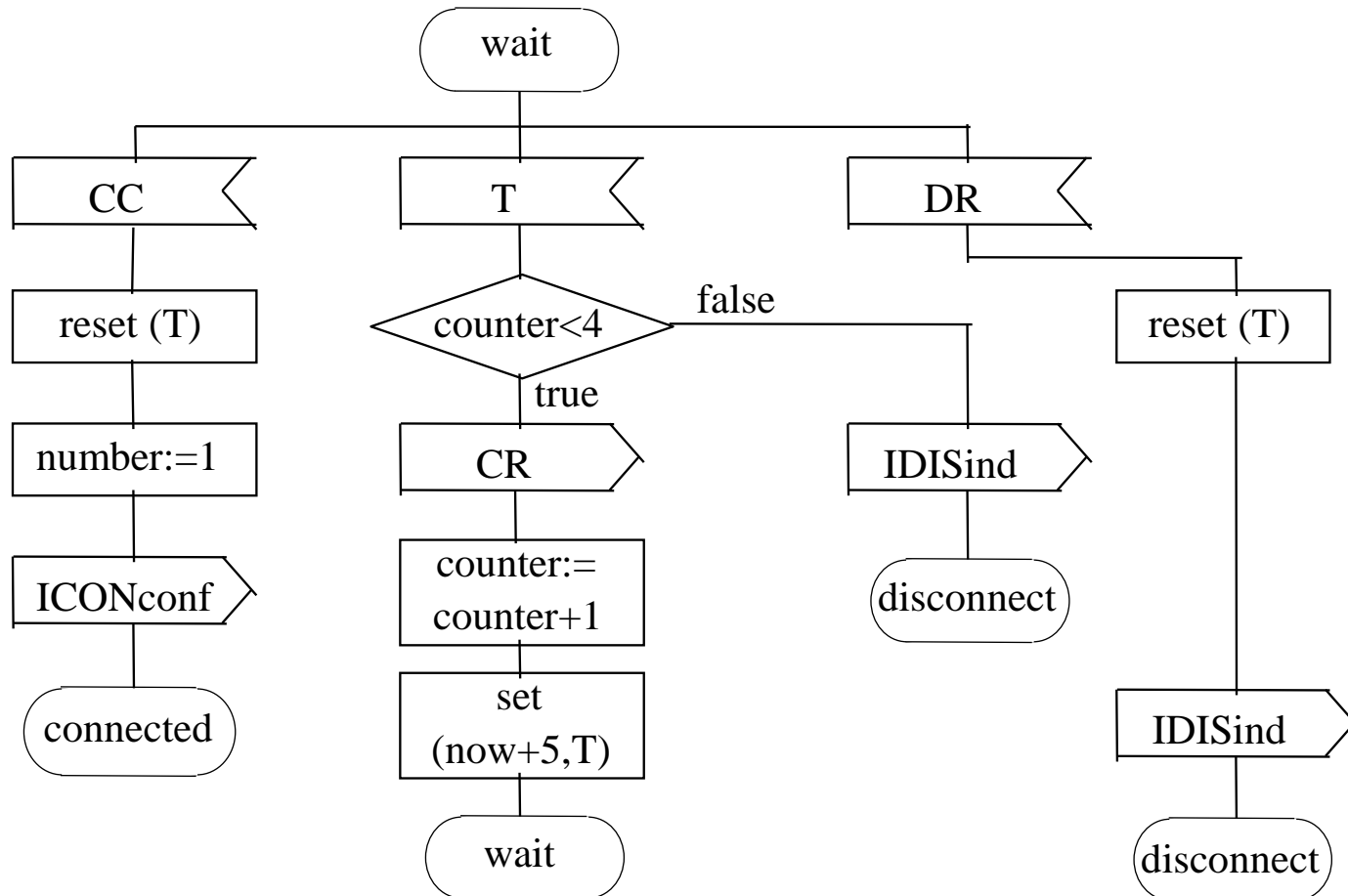
dcl
counter Integer,
d ISDUType,
num, number Sequencenumber,
timer T;
    
```



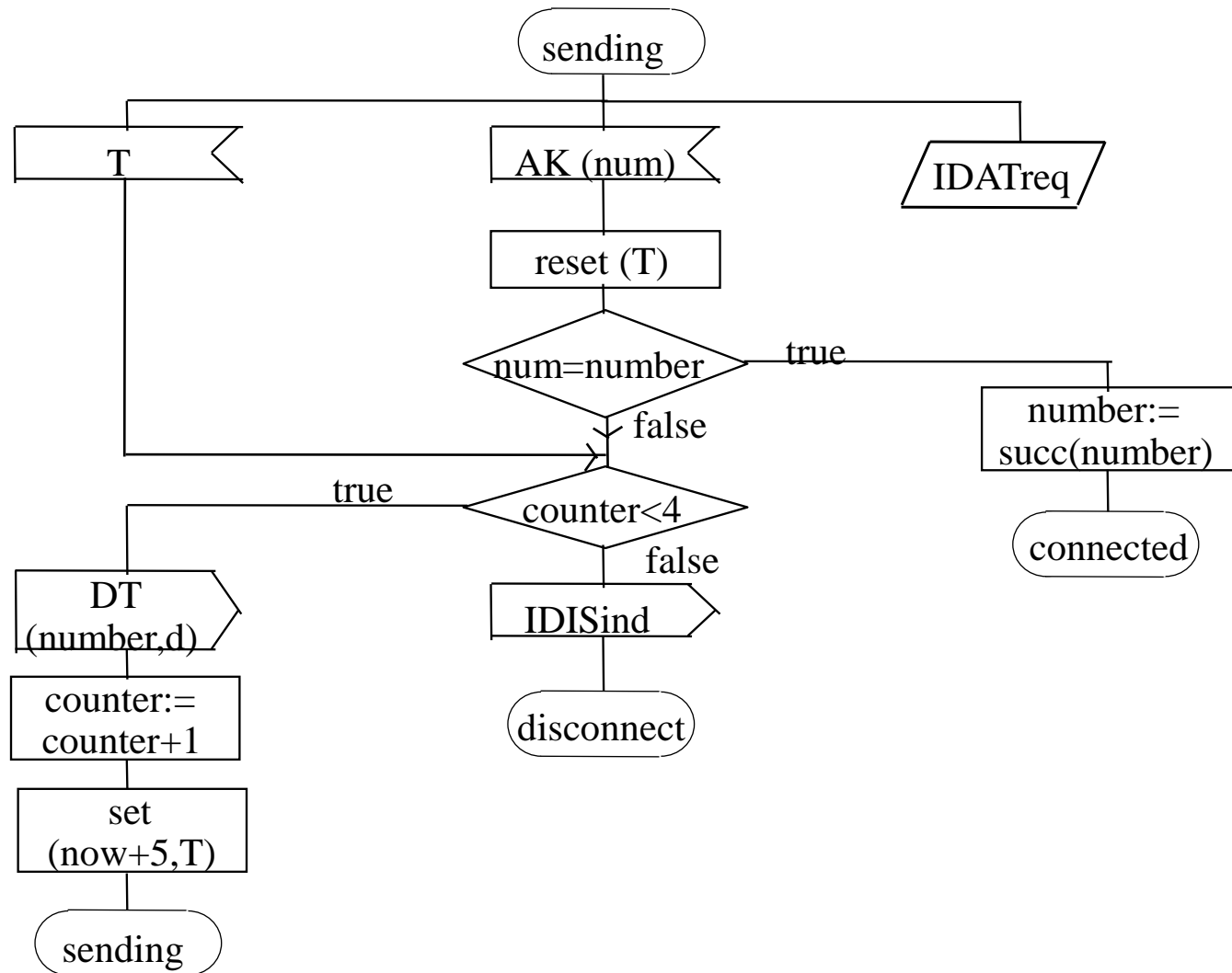
Process Initiator – part I.



Process Initiator – part II.



Process Initiator – part III.

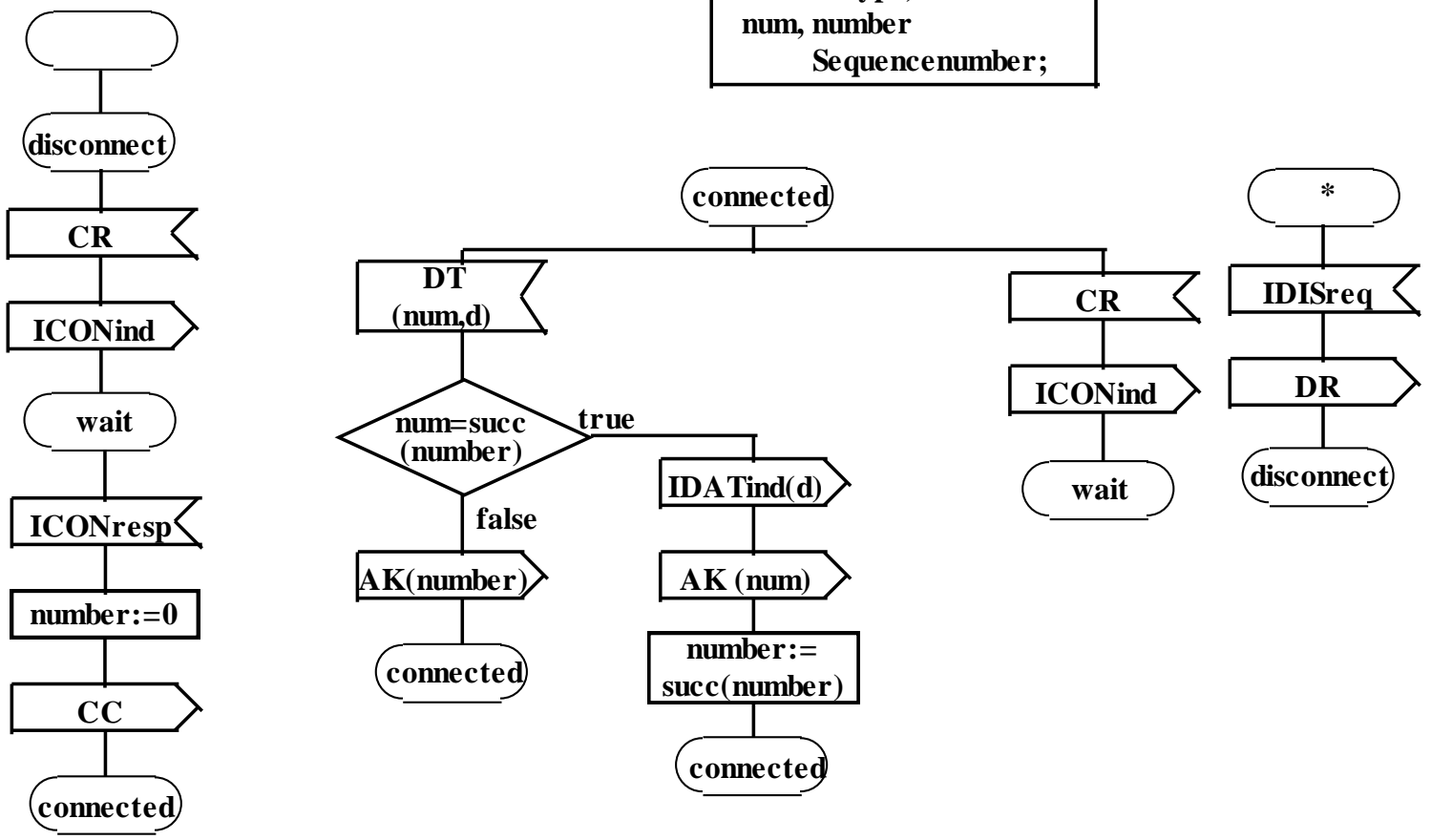


Process Responder

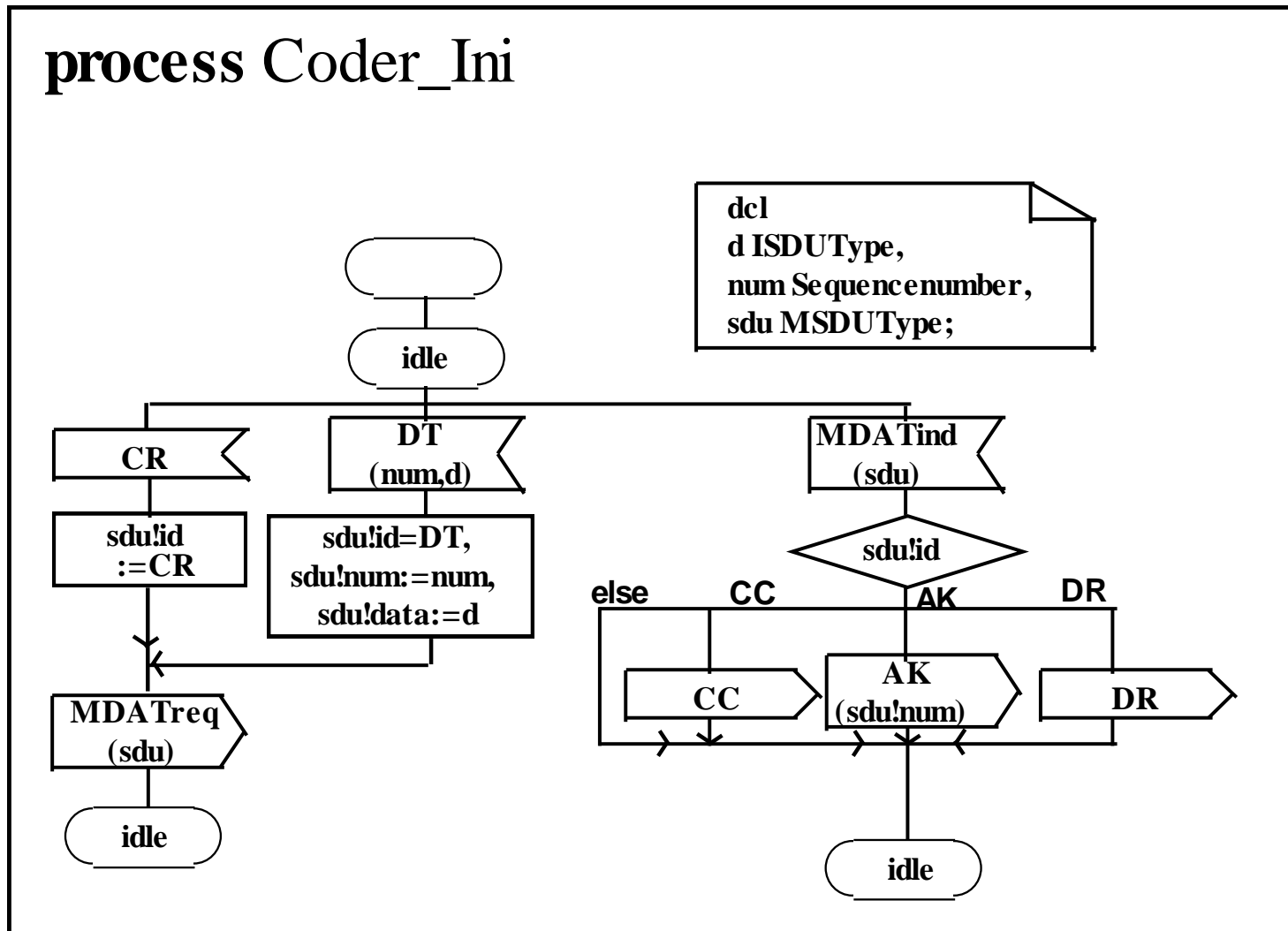
process Responder

```

dcl
d ISDUType,
num, number
Sequencenumber;
    
```



Process Coder_Ini



Process Coder_Res

