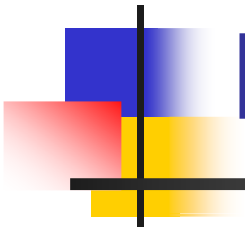
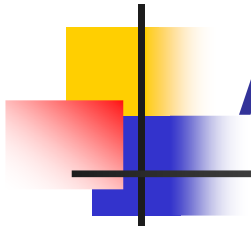


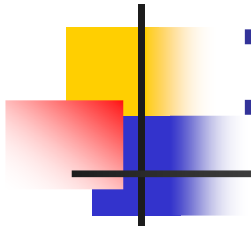
Grammar Directed TTCN-3 CD Program Synthesis

2009-11-15



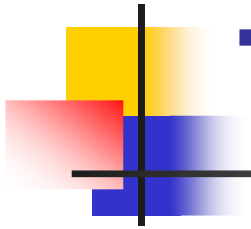
Agenda

- **Introduction**
- **The Challenges**
- **The Common Framework of Generation CD**
- **The Method of automatic CD Generation**
- **Example**



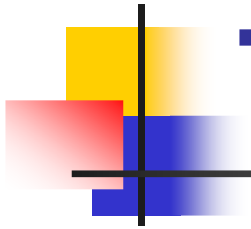
Introduction

- CD is an important component for TTCN-3 testing
- The CD entity is optionally responsible for the external encoding and decoding data associated with message based or procedure based communication within the TE.
- There may be a different CD for different SUT
- CD development sometimes waste of time



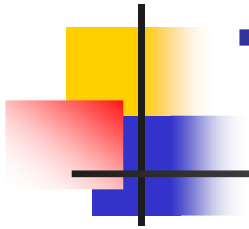
The Challenges

- For large protocol stack, the effort involved in CD development is huge
- The data type of the SUT may change, leading to difficulties in maintaining CD



The Challenges

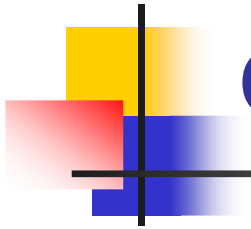
- For example: the CD (Testing Technologies IST GmbH version 1.0.0) of the SIP testing
  - #data types : about 120
  - #Sip Message header type :about 40
- The size of SIP CD in C++
  - LOC : about 10000
  - The amount of functions : about 200
- The size of SIP CD by OSIP
  - LOC : about 5000
  - The amount of functions : about 150



The Challenges

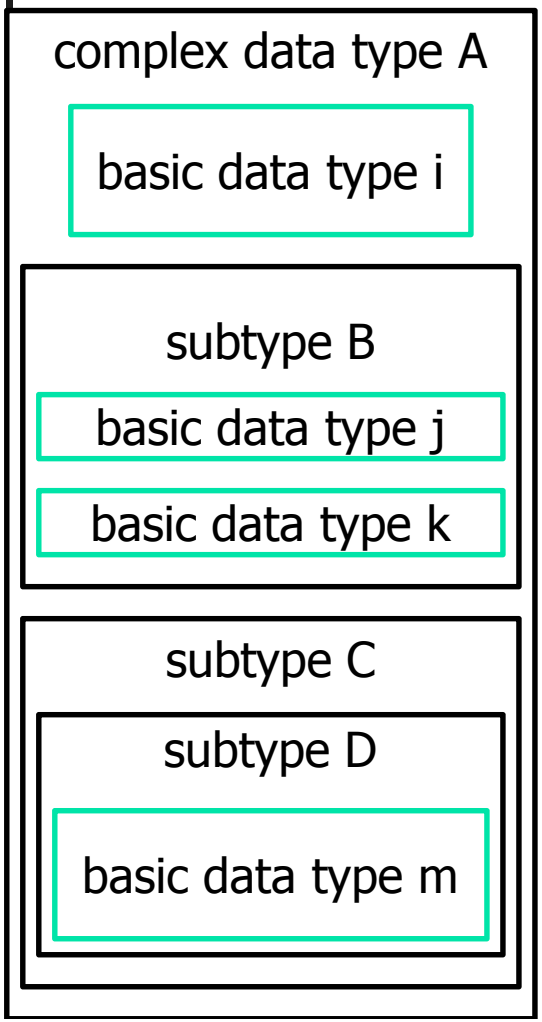
- Problem: there is any cost-effective method?

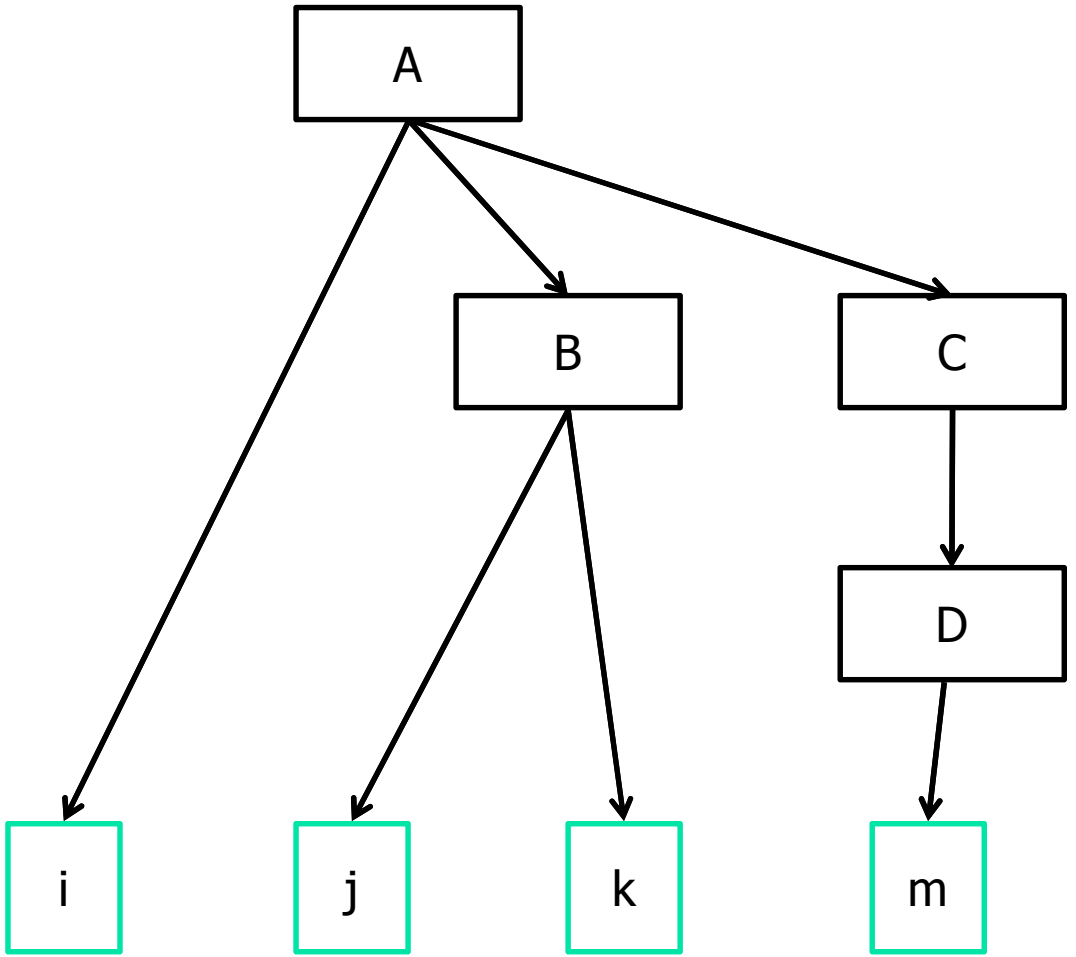
The Common Framework of Generation CD



- The idea of the CD development
  - analyze data flow
  - data flow decomposition
  - CD generation

The Common Framework of Generation CD





The Common Framework of Generation CD

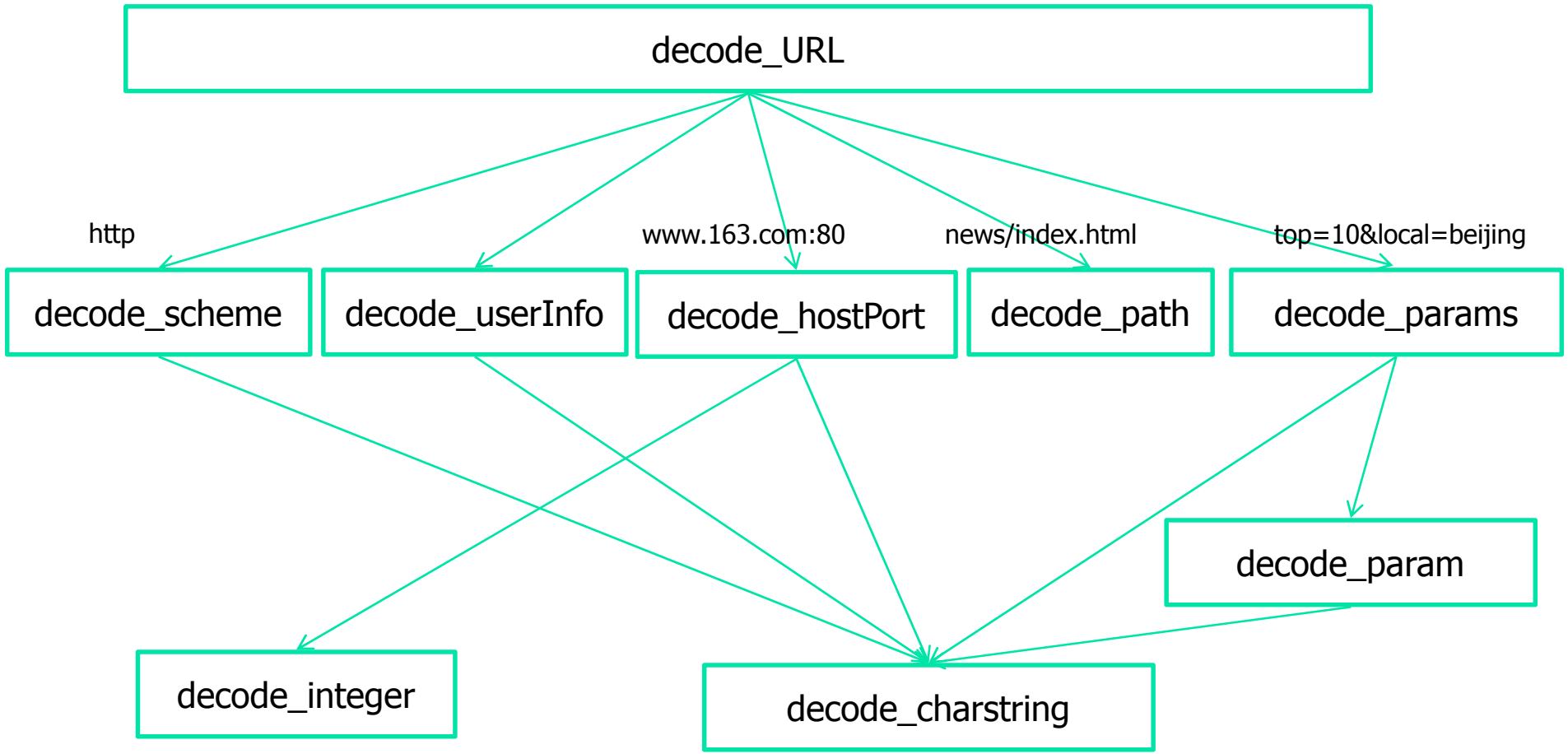
- For example: a URL decode

```
TTCN-3 data type :
  type record Url {
        charstring scheme optional,
        UserInfo userInfo optional,
        HostPort hostPort optional,
        Param_List urlParameters optional
}
```

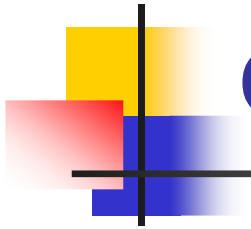
```
Decode function:

decode_scheme;
decode_userInfo;
decode_hostPort;
decode_path;
decode_params;
decode_param;
decode_integer;
decode_charstring;
```

The Common Framework of Generation CD

http://www.163.com:80/news/index.html?top=10&local=beijing



The Common Framework of Generation CD



- The shortage of the common framework of generation CD
  - A different CD corresponds with a different SUT
  - complex CD development is difficult

The method of automatic CD Generation

- The idea of the automatic CD generation
  - data type of TTCN-3 definition

```
For example:
type record URL {
  charstring scheme optional,
  UserInfo userInfo optional,
  HostPort hostPort optional,
 Param_List urlParameters optional
}
```

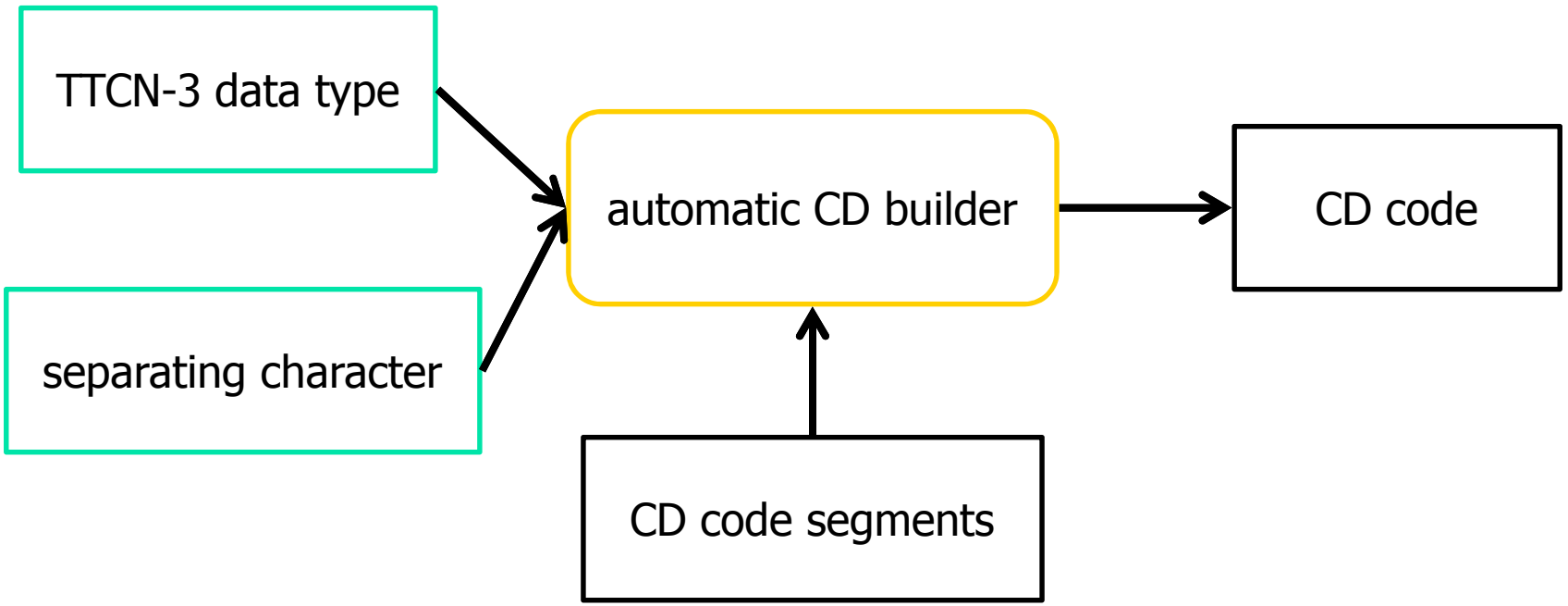
  - separating character

    separating character is useful to analyze the data

  - data type + separating character +syntax➔ CD

The method of automatic CD Generation

- The framework of the automatic CD generation

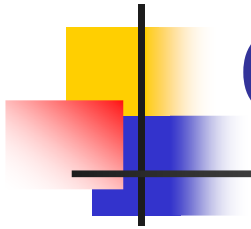


The method of automatic CD Generation

- separating character
  - For example:

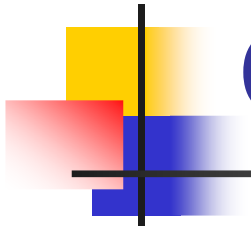
{Url}:='<'|charstring/scheme|':'{UserInfo/userInfo}!'@'{HostPort/hostPort}'>'{Params/urlParameters}!

The method of automatic CD Generation



- Step1:generate syntax of the data type by TTCN-3
- Step2:complete the sentence manually (add the necessary token)
- Step3:generate CD by full syntax

The method of automatic CD Generation



- Step1:generate syntax of the data type by TTCN-3
- Step2:complete the sentence manually (add the necessary token)
- Step3:generate CD by full syntax

Example (SIP CD)

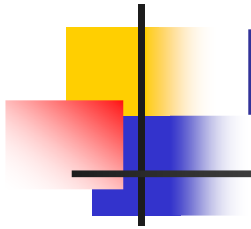
- Step1:generate syntax of the data type by TTCN-3

```
{MessageHeader}:={To/toField}!
{To}:={FieldName/fieldName}{Addr_Union/addressField}{SemicolonParam_Li
    st/toParams}!
{FieldName}:==|enumerated|
{Addr_Union}:={NameAddr/nameAddr}
                {SipUrl/addrSpecUnion}
{NameAddr}:=|charstring/displayName|!{SipUrl/addrSpec}
{SipUrl}:=|charstring/scheme|{UserInfo/userInfo}!{HostPort/hostPort}{Semic
    olonParam_List/urlParameters}!{AmpersandParam_List/headers}!
{UserInfo}:=|charstring/userOrTelephoneSubscriber|!|charstring/password|!
{HostPort}:=|charstring/host|!|integer/portField|!
{SemicolonParam_List}:={GenericParam/genericParam}..
{GenericParam}:=|charstring/id|!|charstring/paramValue|!
{AmpersandParam_List}:={GenericParam/genericParam}..
```

Example (SIP CD)

- Step2:fill the separation tokens in grammar rules manually

```
{MessageHeader}:={To/toField}!
{To}:={FieldName/fieldName}{Addr_Union/addressField}{SemicolonParam_List/toP
    arams}!
{FieldName}:==|enumerated|':'
{Addr_Union}:={NameAddr/nameAddr}
                {SipUrl/addrSpecUnion}
{NameAddr}:=|charstring/displayName|!{SipUrl/addrSpec}
{SipUrl}:='<'|charstring/scheme|':'{UserInfo/userInfo}!'@'{HostPort/hostPort}'>'{
    SemicolonParam_List/urlParameters}!{AmpersandParam_List/headers}!
{UserInfo}:=|charstring/userOrTelephoneSubscriber|!|'='charstring/password|!
{HostPort}:=|charstring/host|!|':'integer/portField|!
{SemicolonParam_List}:=';'{GenericParam/genericParam}..';'
{GenericParam}:=|charstring/id|!|'='charstring/paramValue|!
{AmpersandParam_List}:='&'{GenericParam/genericParam}..'&'
```



Example (SIP CD)

- Step3:generate CD by full syntax

The amount of code : about 1200 lines

Thank you!