A Test-Case Generator using logged Field-Test Traces

R.Kießling

January 22, 2004

THE **TECHNOLOGY** DIFFERENCE

Outline

- **General Idea**
- **Field-Test Logging**
- **Protocol Stack Structures**
- **Test-Case Generation**
- **Test-Case Execution**
- **Current Limitations**
- **Future Plannings**

General Idea … background

- **Testing is still the main possibility to find errors in today's software**

  - … especially if we have to deal with complex architectures as can be found inside protocol stacks of mobile devices

- **Writing test-cases (TC) is a critical procedure often neglected**

  - ... due to a lack of automation

General Idea ... **intentions**

- **Test-Case generators which use, e.g., specifications ...**
  - need detailed behaviour descriptions in a certain format
  - are quite complex to implement

- **Why not use actual data collected anyway during field-tests?**
  - to easily create regression tests
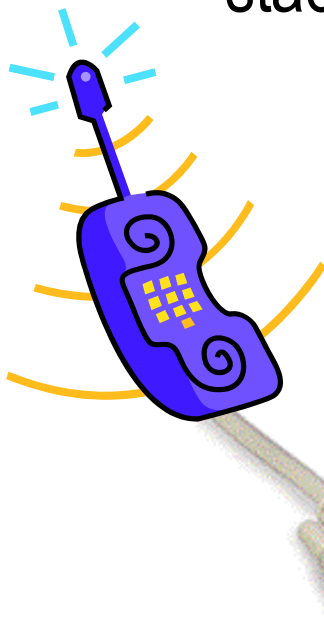 - to conveniently reproduce bugs

General Idea ... data flow

Field-Test

Logfile

100010010100
011001110110

Test-Case Generator

Lab-Test

Development & Bug-Fixing
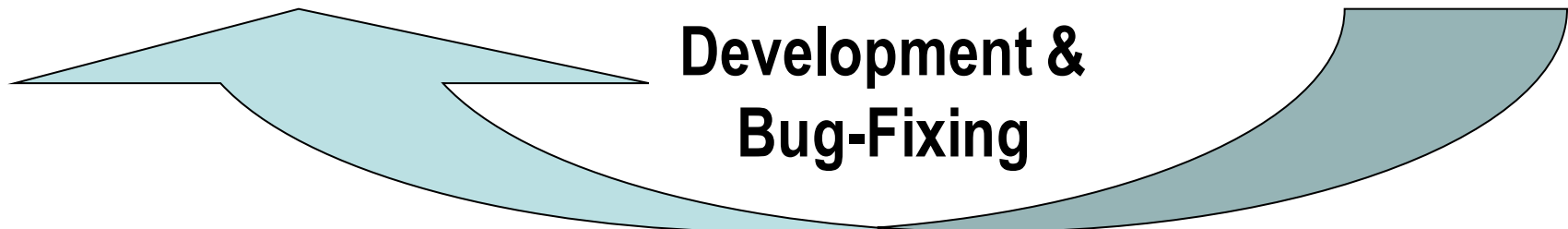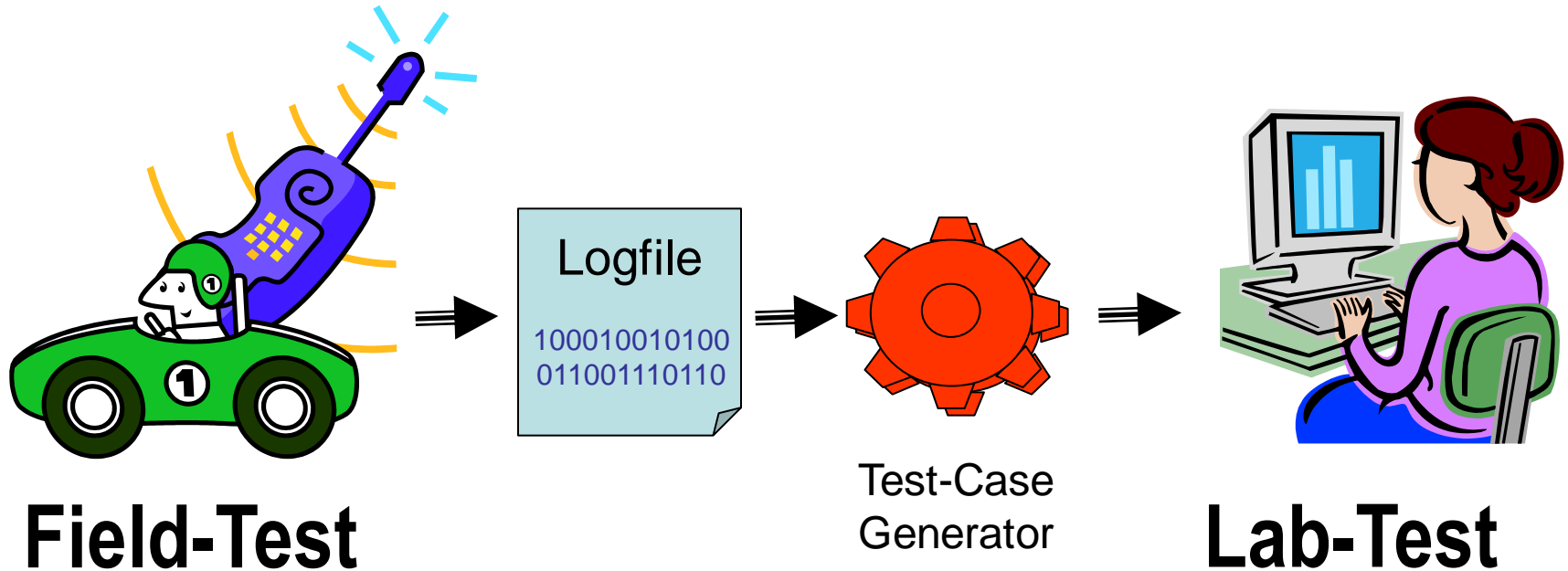
Field-Test Logging ... PCO

- **The tool PCO is used to supervise field-tests**
    - … and record all communications inside the protocol stack (PS) of, e.g., a cellular phone

Logfile

100010010100
011001110110

Field-Test Logging ... GPF-FRAME

- ## A generic framework is used on target and tool side

  - ### …supporting duplication of the communication flow

Dupliciation via the Test Interface
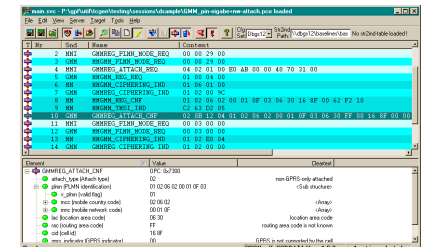
GPF-FRAME

GPF-FRAME

PS Structures ... primitives

- # A protocol stack consists of several entities

  - ... communicating via primitives exchanged between them

- # Primitives contain internal data and air-messages

Entity 1

primitive

| Byte_element |
| Long_element |
| Struct_element |
| Air_message_SDU (encoded) |

Entity 2

air-message (decoded)

| Byte_element |
| Long_element |
| Struct_element |
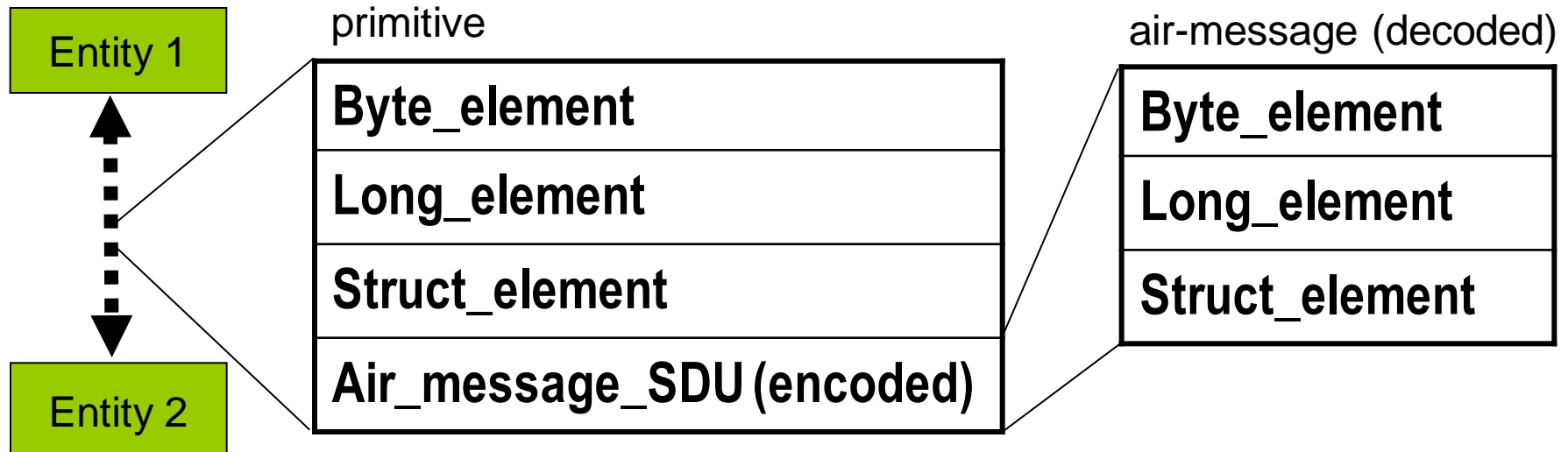
PS Structures ... database

- ## The altering structure definitions are collected in a database

  - ... to keep test tools independent from the actual PS

specifications of primitive and air-message structures (ETSI conform) → table generator → table definitions in C header files → C compiler/ linker → CCD Database (Win32-DLL)
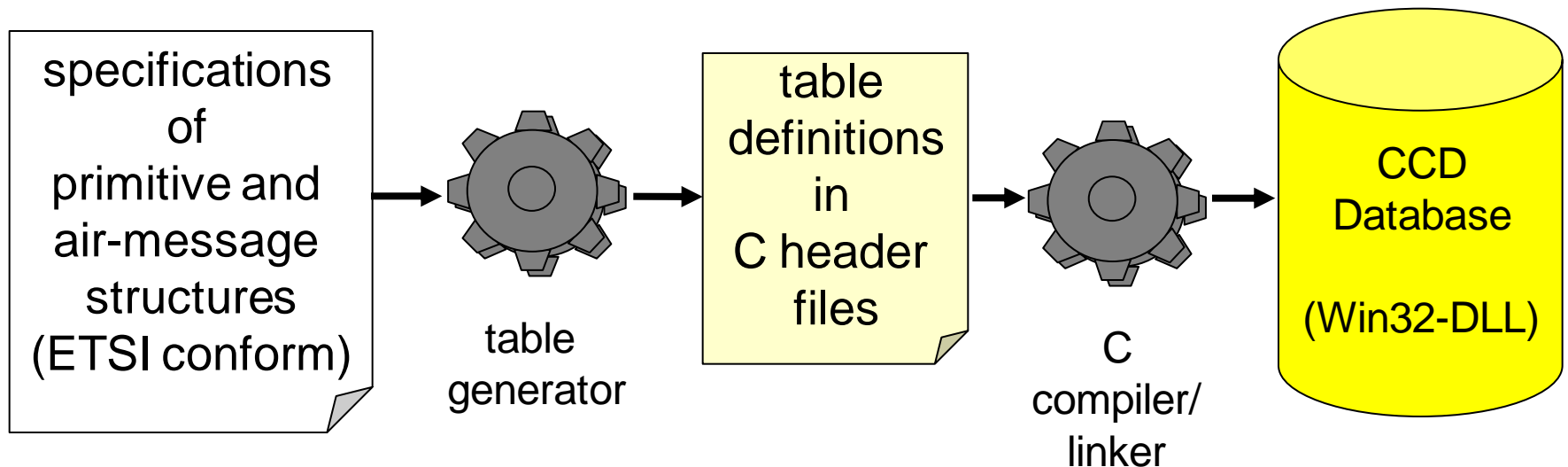
TC Generation ... concept

- **The test-case generator TCGen is implemented as a so-called PCO-Viewer**
  - ... to request logfile data from PCO
- **Primitive interpretation is done using the database**
  - ... which has to match the PS monitored during field-test
- **TDC is a proprietary test script language**
  - ... in which test cases will be generated
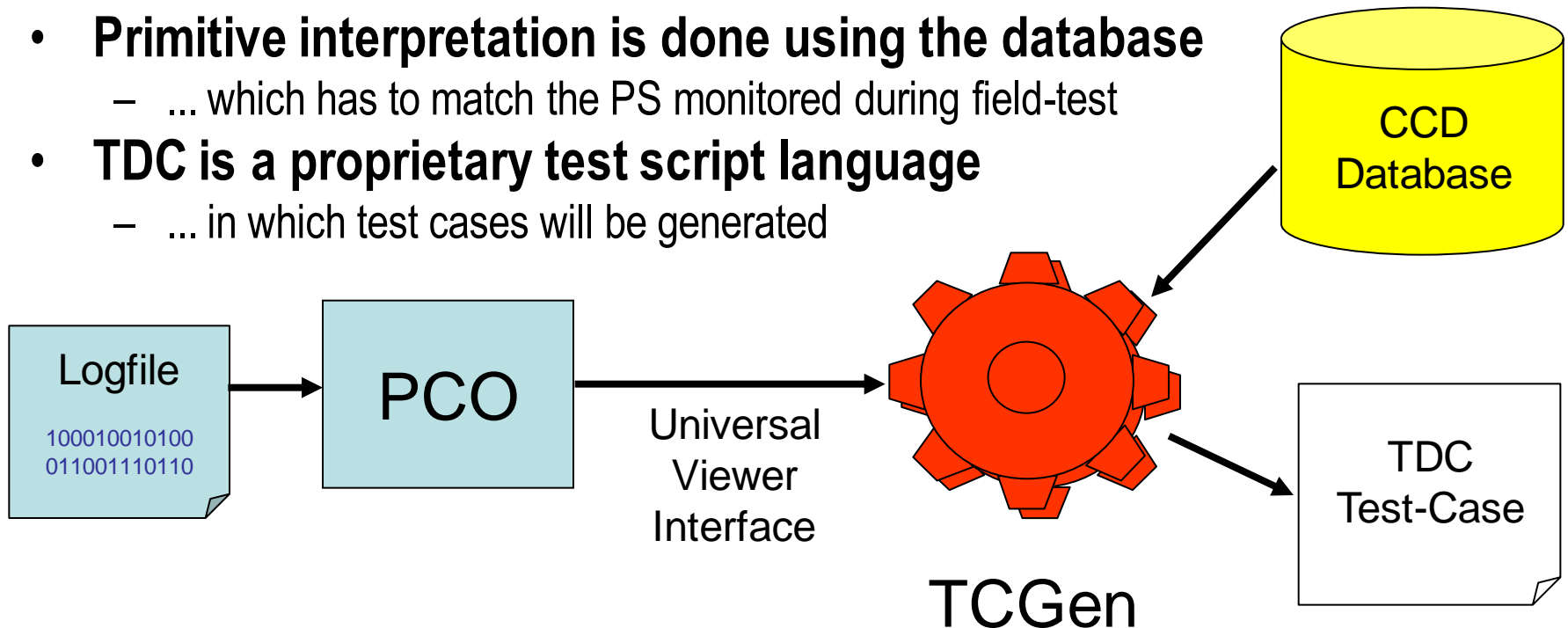
TC Generation ... algorithm

- **The user specifies one or more entities to TCGen**
  - ... which will be considered as the entity block under test (EUT)

- **Recorded primitives are examined in an initial pass to generate ...**
  - SEND-commands for all data received by the EUT
  - AWAIT-commands for all data emitted by the EUT

```cpp
// Test Step Excerpt: gmmsm_steps.cpp

port2GMM.SEND (mmgmm_tmsi_ind_GMMSM028_8());
AWAIT (gmmreg_attach_cnf_GMMSM028_9());

TIMEOUT(300);

port2GMM.SEND (gmmreg_plmn_mode_req_GMMSM028_10());
```

```cpp
// Constraints Excerpt: gmmsm_constraints.cpp

T_PRIMITIVE_UNION gmmreg_attach_cnf_GMMSM028_9()
{
    T_GMMREG_ATTACH_CNF prim;
    prim->attach_type= 0x02;
    prim->plmn= plmn_GMMSM028_6();
    // ...
T_plmn plmn_GMMSM028_6()
{
    T_plmn pstruct;
    pstruct->v_plmn= 0x01;
    pstruct->mcc= mcc_array_GMMSM028_7();
    // ...
T_ARRAY<U8> mcc_array_GMMSM028_7()
{
    const U8 array_elements[3]=
    {
        0x02,
        0x06,
        0x02
    };
    return T_ARRAY<U8>(array_elements);
} // mcc_array_GMMSM028_7;
```

- **By further passes definitions of the primitive content are created**
  - ... according to the values actually used during the field-test

TC Generation ... rules

- ## TCGen supports rule files in XML format
  - – to automatically apply needed modifications of the data recorded

generation fragment

rule file example

...
find_primitive_element()
apply_rules()
write_element()
...

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Rules for TCGenerator -->
<tcgen>
  <options max_timegap="0" timeouts="0"></options>
  <skip primitive="MMGMM_TRIGGER_*"></skip>
  <change primitive="*" param="split_pg_cycle">0xD6</change>
  <change primitive="*" param="new_tlli">
    ((%v & 0xC0000000)==0x40000000 ? 0x7E081997 : %v)
  </change>
</tcgen>
```

- ## Test cases can also be adapted manually
  - – ... because they are generated in a human readable format

TC Execution ... TAP

- **The tool TAP can be used to execute generated test-cases**
  - ... together with a real target or a simulation of the protocol stack
- **Such test runs can be regression tests**
  - ... if, e.g., a succesful phone call has been recorded
- **... or reproductions of erroneous behaviour**

Generated TDC Test-Case

PS Simulation (on PC)

Shared Memory Interface
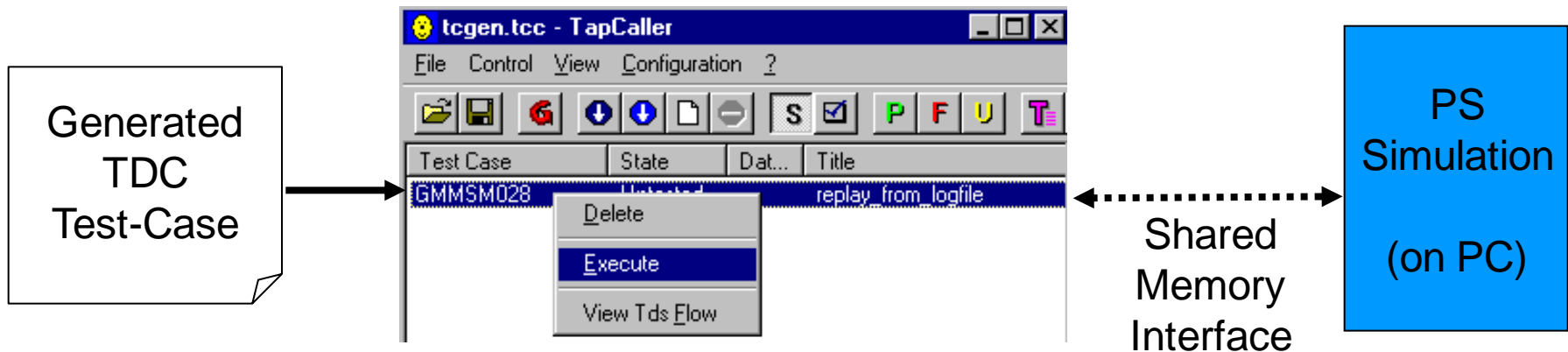
TC Execution ... entity isolation

- **To avoid disturbances from other entities the EUT have to be isolated inside the tested PS**
  - ... this is supported by the routing mechanisms of the generic GPF-FRAME

Entity A

EUT

Entity 1

Entity 2

Entity B

TAP
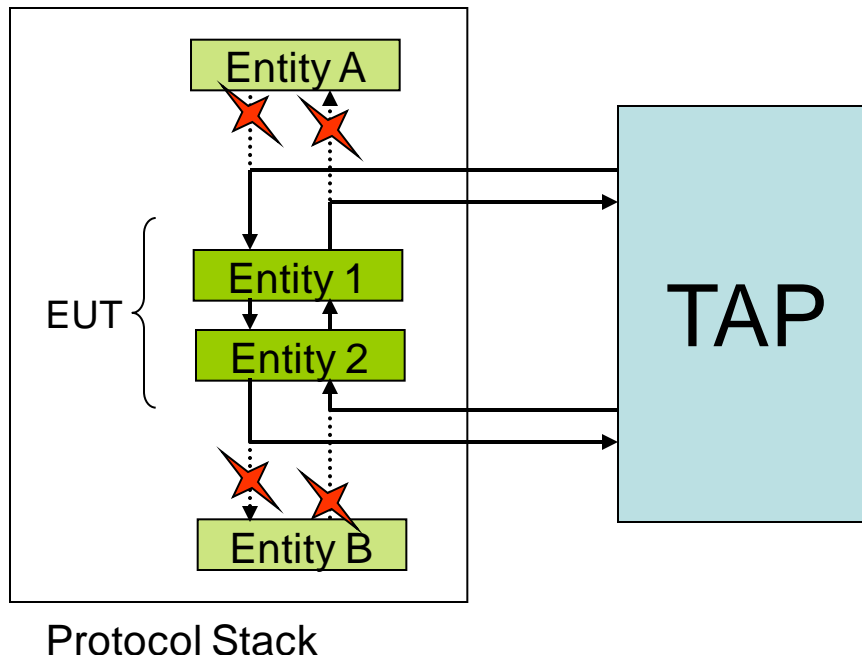
Protocol Stack

- **Adequate routing commands are contained in the test-case**
  - ... also generated by TCGen

- **This way no real air-network is needed**
  - ... to, e.g., retrace an error in a specific entity

Current Limitations

- **Only selected parts of the protocol stack can be logged**

  – ... due to insufficient data rates currently available on the test interface

- **Not all communications between entities can be duplicated**

  – ... because at some points functional interfaces are used for performance reasons

- **… but solutions are on the way !**

Future Plannings

- **Support of other test scripts**
  - ... e.g. TTCN-3

- **Enhanced possibilities in rule files**
  - ... e.g. regular expressions

- **Consideration of further input**
  - ... e.g. message sequence charts