

为什么模型化在 远程通信应用中最先进



Emmanuel Gaudin 从事 SDL 的协议栈的开发，有较深的技术背景。他 96 年加入了一个模型化工具供应商，担任现场应用工程师及培训师，最后成为了法国支部的技术总监。基于他的经验，2001 年他成立了 PragmaDev 公司，以开发 SDL-RT 工具。

本文将关注模型化技术在远程通信领域的应用，以展示相对于其它的领域它是多么的先进的。

一致性不是可选的

一个远程通信系统是一套能够相互通信的不均匀的设备。要做到这一点，这些设备必须遵从同样的协议。从技术上讲，这包含 2 个互补的方面：

一个**静态界面**——从一个设备到另一个的交换的报文格式。

一个**动态界面**——定义了交换的报文的正确的序列。

对一个远程通信系统协议，这两方面的一致性都很重要。此外，协议定义需要尽可能清楚不能有歧义。

标准化机构

标准化机构定义的协议公布了这两个方面的介绍。从历史上看，世界的每个地区都有自己的标准化机构，如欧洲的 ETSI，日本的 ARIB，美国的 ATIS。当时，因为标准是不同的，这就意味着在一个区域的设备如手机，无法在另一个领域正常工作。多亏了全球化，不同的标准化组织现已携手发布全球标准，如 3G 或即将到来的 LTE (Long Term Evolution)，它是 4G 技术的可选方案之一。

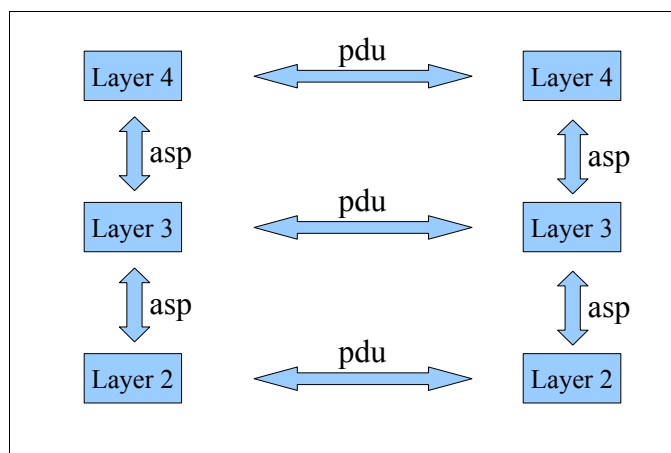
为了产生清晰且易于理解的标准，欧洲标准化机构 ETSI 已经公布了一组介绍，通过一组技术如 SDL (Specification and Description Language 规范和描述语言)，ASN.1 (Abstract Syntax Notation One 抽象语法记法一)，ECN (Encoding Control Notation 编码控制符号)，TTCN-3 (Testing and Test Control Notation 测试和测试控制表示法) 来“制作出更好的标准”。这些介绍可见于 ETSI 网站：

<http://www.etsi.org/WebSite/Technologies/ProtocolSpecification.aspx>

本文将论述这些技术如何帮助远程通信行业生成更好的标准。

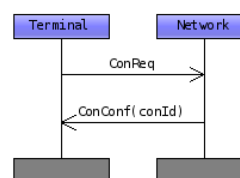
动态界面

一种协议描述了同级的从一层到另一远程层的报文交换。这样的界面称为 Protocol Data Unit 即 PDU。



这实际上是一个协议的逻辑视图，因为实际中一层与上一层或下一层交换信息。这种“垂直”的界面被称为 ASP (Abstract Service Primitive)。因此 PDU 通过 ASP 发送到远程层，但不发送描述交换的 PDU 的协议定义。

动态描述的协议逻辑视图由一堆情景组成。ETSI 建议使用 MSC (消息序列图)。我们看下一个简单的例子，一层请求连接并在 MSC 中描述。



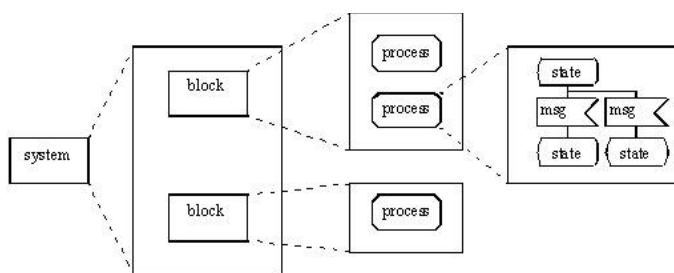
在上面的情景中，报文在终端机与网络间进行交换。一条报文可能包含参数如例子中 ConConf 报文带有 conId 参数。

大多时候，一个远程通信系统是一个分布式系统，终端是脱离网络的实体设备。它很可能有一个不同的处理器架构并运行不同的操作系统。因此，必须用一种方法来创建一个信息流，使该信息流独立于处理器和操作系统。

静态界面

为了定义一个硬件独立的信息流，要一起使用抽象的数据表示法与 ASN.1 表示法。ASN.1 指的是 Abstract Syntax Notation One，由 ITU-T 标准化。它描述的是可理解的数据类型，包括基本类型如布尔型或实数，复杂类型如结构或数组。ASN.1 描述了类型，但没有说明如何在一个给

定的语言下操纵它们。主要关注的是 ASN.1 定义了编码和解码规则，以便为某一类型的信息数据流独立于给定的硬件结构的实施。除了标准化的编码规则，如 BER（Basic Encoding Rule 基本编码规则）或 PER（Packet Encoding Rule 分组编码规则），它们也有一个符号来描述具体的编码规则，它就是 ECN：Encoding Control Notation 编码控制符号。最后一点，ASN.1 数据类型可以直接导入到 SDL 作规范，到 TTCN 进行测试。



An ASN.1 example structure:

```
PersonRecord ::= SEQUENCE {
    name      IA5String,
    age       INTEGER,
    member    BOOLEAN
}
```

An example value:

```
examplePerson PersonRecord ::= {
    name      "Smith",
    age       35,
    member    true
}
```

XML encoding of the value:

```
<PersonRecord>
  <name>Smith</name>
  <age>35</age>
  <member>true</member>
</PersonRecord>
```

协议详细的行为

MSC 描述的协议行为只是几种可能的情景。即使它描述可替代，可循环，或可选的情景，MCS 也无法描述发生在一个协议的所有可能的情况。要做到这点，它关乎要描述的该协议的详细和可执行的模型。以这种方式，它可能可以模拟和验证它的行为，这样该模型被认为是一个参考模型。为了这个目的，ETSI 推荐一个正式的建模语言，它称为 SDL 规范和描述语言。

SDL 已在 80 年代首先用来描述远程通信协议，无需说明语言是否符合我们的要求设计。但随着不同技术的革新，语言也自此跟着演变，如面向对象的介绍，UML 概要文件的定义。

从技术上讲，SDL 有 4 个互补的观点，我们以下将会简要介绍。

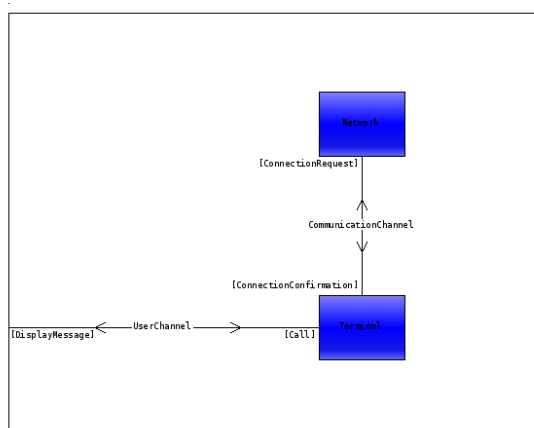
- **体系结构**

一个系统由块组成，这些块可以分解为子块。

体系结构的页就是一个进程，主要是一个带有隐含消息队列的有限状态机。

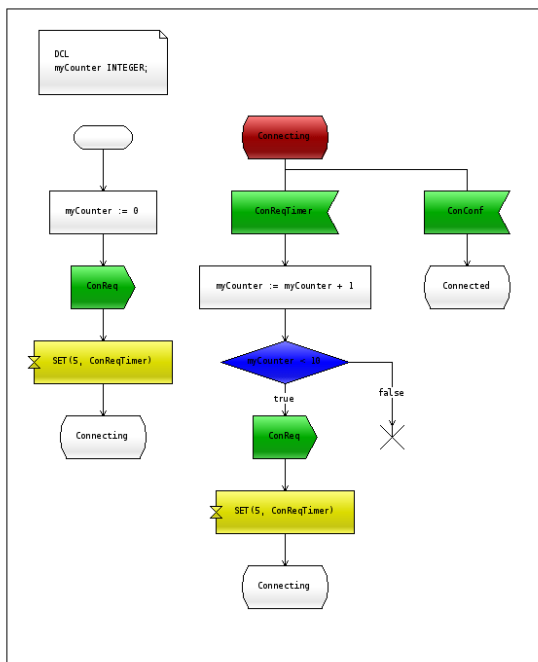
- **通信**

不同块间交换的信息流在体系结构中有描述，用来定义块的接口。



- **行为**

详细的行为被完全生动地描述。根据规格想要的精确度，它可能忽略了实施细节。



上面的例子展示了一个基本的有限状态机：初始化时 ConReq 消息发送出去，ConReqTimer 开始启动。状态机转到 Connecting 状态并等待连接确认消息 ConConf，或者计时器关闭。之后，连接请求 10 次。如果仍然不成功，进程停止。

- **数据**

为了使规格完整无歧义，SDL 嵌入抽象数据类型和语法来操纵这些数据。如前所述，最好导入 ASN.1 数据类型。

协议一致性

现在的问题是确保实现的相同协议符合 ETSI 公布的标准。对于近代的标准如 SIP 协议，IP V6，3GPP IMS 等，ETSI 的已公布基于 TTCN-3 语言的一致性测试套件。为了确保它符合需求，ETSI 还保持了由 ITU-T 标准化的 TTCN-3 语言。TTCN 代表测试和测试控制符号，它是国际标准化组织发起的一致性测试方法和框架文件 (9646-3) 的一部分。TTCN-2 专用于通信系统，TTCN-3 适用于测试任何一种系统。

TTCN-3 可视为一个经典文本的编程语言，但几个方面使用它成为了一种十分强大的测试语言。

- **抽象层**

SDL, TTCN-3 包含报文和计时器的概念，它们是一个远程通信系统的基本服务。这样就很容易发送或接收一条报文，启动或取消一个计时器。

- **模板**

当测试一个远程通信系统时，大部分的工作是核实交换的报文是否包含正确的信息。TTCN-3 定义了模板的概念，轻松地验证一组复杂参数的正确性。TTCN-3 还可选择参数（参数可能不存在），可以忽略一些参数（参数是存在的，但与它的值无关）。

- **可选方案**

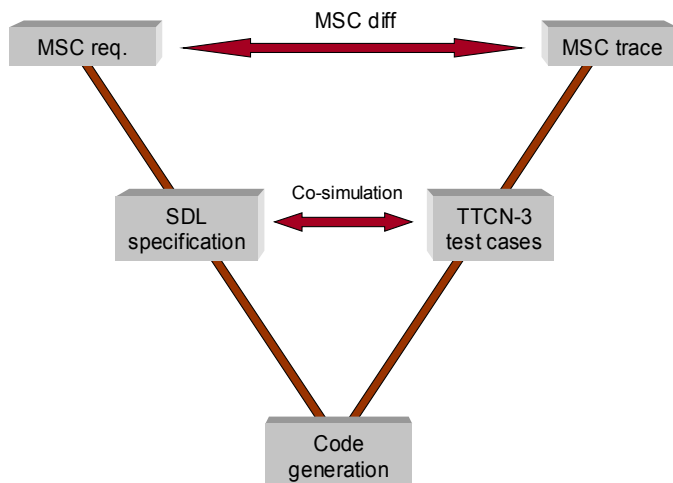
测试描述可选方案并根据不同的可选方案设置结论。TTCN-3 嵌入可选方案的概念，无论它们是基于异步信息如报文的交换还是同步信息如修改的变量值。

```
cEnv.send(ConReq);
ConReqTimer.start;
alt {
    []cEnv.receive(ConConf){
        setverdict(pass);
    }
    []ConReqTimer.timeout{
        setverdict(fail);
    }
}
```

上面的例子显示了 TTCN-3 的一个基本可选方案：一个连接请求 ConReq 通过 cEnv 端口发送，计时器 ConReqTimer 开发启动。在这个可选方案中，或是接收到 ConConf 响应或是计时器 ConReqTimer 关闭。

模型测试

由于 SDL 和 TTCN-3 有同样的抽象层，两种语言都可在实施前测试一个规范模型。由于两种语言都是正式的，意味着完整且没有歧义的，可以从 SDL 规范中生成协议实施的代码，从 TTCN-3 测试例中生成测试代码。一个早期的验证将最终节省大量的时间和精力。



最后一点，ETSI 发布了 TTCN-3 的一致性测试套件，这样远程通信制造商可以确保其实施符合标准。

结论

由于远程通信系统本质要求符合一个共同的标准，该标准包括静态和动态接口，远程标准化机构以及远程通信设备制造商采用多年的先进技术如 MSC, ASN.1, SDL 和 TTCN。事实上，这些技术涵盖整个开发周期，从需求，规划到设计，测试。这就是为什么在开发通信系统时必须考虑清楚的原因。