一种自适应GPS通信协议分析算法的设计与实现①

金　珏 (宁波大红鹰学院 软件学院 浙江 宁波 315175)

摘　要： 由于缺少 GPS 通信服务标准，不同的 GPS 监控设备在 GPRS 上传输的数据通信协议有较大差别，这给综合性的 GPS 监控信息接入带来了困难，协议版本的不断变化也加剧了数据分析的困难。基于有限自动机模型，设计了一种自动协议分析算法，能够实现一套程序同时对多种协议进行分析和转换。

关键词： GPS 通信；DFA；协议自动分析

Design and Implementation of an Adaptive GPS Communication Protocol Analysis Algorithm

JIN Jue (College of Software, Ningbo Dahongying University, Ningbo 315175, China)

Abstract: Due to absence of the standards of GPS communication services, the data communication protocols transmitted on GPS through different GPS monitoring devices are quite different, which caused difficulties to the comprehensive GPS monitoring information access. And the changing of the protocol versions makes the data analysis more difficult. Based on DFA, an automatic analysis algorithm of protocols is designed so that a program can analyze and transform multi-sort protocols at the same time.

Keywords: GPS communication; DFA; protocol automatic analysis

1　引言

在信息监控、物流调度、通信规划等领域经常需要实现 GPS 设备的实时数据通过 GPRS 网络发送到信息中心一端。由于没有相关的标准，中心的接入层必须自己解决不同型号设备之间通信协议不兼容问题，目前大多的做法是为每一种设备实现专门的解析算法。这大大降低了软件的可维护性。

如果能够设计一个统一的 GPS 通信协议分析程序，支持对各种不同的通信协议进行分析，则可以大大提高GPS设备监控的能力。由于GPS通信领域众多，协议复杂，这个问题目前尚缺乏理论上的答案，文献[1-3]给出了一种基于状态机的协议分析算法；文献[4,5]给出了基于 CSP 的自动验证算法，这些算法只能用于验证工作，即验证给定的数据流是否满足该协议，数据流的分析、转换和记录工作需要另外编码。

针对 GPS 通信协议的特点，设计了一种 GPS 通信协议的确定型自动机，能够利用该自动机，将通信数据流直接转换成程序设计语言中的类和对象。对于任何一种 GPS 通信协议，只要事先根据协议规范建立协议的自动机描述，满足该协议的数据流就会被转换成一组标准的 UML 对象，从而实现了自适应的通信协议分析。

基于这种技术，协议分析通常可以采用两个阶段：

1) 协议定义阶段：定义协议的自动机模型，并设计了一种 XML Schema，将模型以 XML 文件的形式保存下来。

2) 数据流转换阶段：截获数据通信包，并读取相应的模型定义文件，根据自动机模型对数据进行分析；并设计了一种标准 GPS 信息的 UML 类图，可以将流式的数据自动转换成标准的 UML 对象，这组 UML 对象目前采用 C#实现，进一步的处理，如入库、显示等全部转换为对对象的操作，而不再是无类型的字节数组。

2　GPS通信协议的自动机模型

2.1 自动机模型

自动机模型在通信协议中普遍得到采用，主要用

① 收稿时间:2009-06-30

于通信协议的形式化定义、完整性验证等。GPS 通信协议的自动机模型由一组状态集合、操作集合、输入流字节组、初始状态集合和终止状态集合组成的五元组。通过这五元组，实际上描述了 GPS 通信协议的分析流程，如图 1 所示，该自动机模型忽略了异常情况。

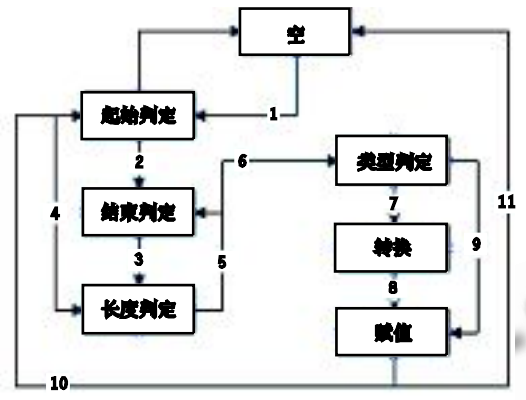

图 1 通信协议的自动机模型

这个模型描述了将一个满足协议的字节数据流转换成 UML 对象过程或者反过程的状态变化。所谓状态是一个二元组，包括一组游标的集合，用于表示数据流的处理范文范围；当前 UML 模型对象实例的集合，表示当前转换对象。模型中的状态转换是由一系列操作来完成。

(1) 标尺转换：记录并改变字节流数据的标尺位置。

(2) 模式匹配：起始标尺与结束标尺之间进行匹配，找到相应的协议单元

(3) 字节映射：将字节流的数据映射为对象的属性。

(4) 表达式计算：在字节映射的同时，在数据类型匹配之前，还应进行相应的表达式计算包括布尔表达式和算术表达式。

(5) 类型转换：将某个值赋给 UML 对象的某个属性。

(6) 实例化：创建某一个 UML 类型的对象实例。

图 1 中操作 1 对应的是模式匹配操作；操作 2 对应的是标尺转换和模式匹配操作；操作 3 对应的是表达式计算操作；操作 5 对应的是标尺转换操作；操作 6 对应的是实例化操作；操作 7 对应的是类型转换和表达式计算操作；操作 8 和操作 9 对应的是字节映射、实例化操作；操作 10 对应的是标尺转换操作。

这 6 个操作不断的改变状态二元组，在 GPS 通信协议分析过程中一共形成了七种状态：

A. 空状态：没有待分析的字节流的状态，此时起始游标终止游标都是 0，对象集合和当前对象都是空。

B. 起始判定：出现字节流或者字节流中某个部分分析结束会进入起始判定，起始判定用于查找字节流中某个部分的起始，例如经度数据的起始。起始判定结束后将改变状态中的起始游标。

C. 终止判定：起始判定结束后起始游标被确定，进一步需要确定终止游标的位置，由于协议不同，终止判定有时候是在长度判定状态之后进行的，终止判定将改变终止游标的值。

D. 长度判定：很多协议标明了一次数据或者一次数据中的一个部分(如坐标信息)的数据长度，由数据长度进行终止判定。

E. 类型判定：在确定了数据流中一个完整协议的起始和终止，或者确定了一个完整协议的一个组成部分的起始和终止后，下面进一步判定该数据的类型，并根据类型判定数据协议对应的 UML 对象，该状态将在 UML 对象集合中插入适当类型的对象。

F. 转换：一个数据流的部分的起始和终止被确定，类型也被确定，有时候需要对该部分数据进行相应的转换，例如“#TYPE;A1021;X;Y;TIME; SPEED; ANGLE*”的第 13 字节和 15 字节之间的是坐标，需要将字符串转换成浮点数，进一步需要将 GPS 坐标转换成某种地理坐标等。转换状态不是必须的。

G. 赋值状态：将转换后的数据赋值给指定的 UML 对象，或者赋值给指定的 UML 对象的指定属性。赋值状态完成表示分析的一个阶段结束，例如坐标的分析结束，进一步需要分析 GPS 速度部分，赋值状态改变了当前 UML 对象，并进一步回到起始判定状态或者空状态。

例如：在数据串“#ID,101111; TIME*”中“101111”两位一组分别表示报警等级、报警类型、环境温度信息。按照此自动机机分析过程如下：

(1) 接收#进入起始判定，建立起始游标

(2) 接收*进入结束判定，建立终止游标

(3) 立即进入长度判定状态，终止游标与起始游标相减

(4) 接收#下一个字符，再次进入起始判定

(5) 接收;进入结束判定，同时生成相应的 UML

对象

(6) 取得 ID 部分，进入类型判定状态

(7) 判定 ID 是否有效，进入类型转换和赋值状态

(8) 游标回到;，再次进入起始判定

(9) 起始游标加 2，进入结束判定

(10) 取得起始游标和结束游标中间的字符“10”

(11) 同上，依次进入类型判定(本例中是否是有效数值)、转换(本例中是字符 10 转换成数字 10)、赋值(将 10 赋值给相应的 UML 对象)

(12) 如此反复进行，直到整个接收字符都被处理。

通过以上分析过程可以看出，该自动机模型的状态是操作状态，状态的转移是文法满足条件后进行相应的状态变迁，因此描述的是动态分析过程。

2.2 GPS 通信协议自动机的 Schema 描述

用以上自动机模型描述的动态分析过程可以以 XML 文件的形式记录下来，下面给出了 GPS 通信协议的自动机的 XML Schema 的主体内容.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schemaelementFormDefault="qualified" xmlns:xs=" ">
<xs:element name="GPSProtocol" nillable= "true" type="GPSProtocol" />
<xs:complexType name="GPSProtocol">
<xs:sequence>
<xs:element maxOccurs="1" name= "Start-Operation" type="Operation" />
<xs:element maxOccurs="1" name="Op-erations" type="ArrayOfOperation" />
</xs:sequence>
<xs:attribute name="ID" type="xs:int" use ="required" />
<xs:attribute name="Caption" type="xs: string" />
<xs:attribute name="Version" type="xs: unsignedInt" use="required" />
</xs:complexType>
<xs:complexType name="Operation">
<xs:sequence>
<xs:element minOccurs="1" maxOccurs= "1" name="Type" type="OperType" />
<xs:element maxOccurs="1" name="Value" type="xs:string" />
<xs:element maxOccurs="1" name= "Val-ueType" type="xs:string" />
</xs:sequence>
</xs:complexType>
<xs:simpleType name="OperType">
<xs:restriction base="xs:string">
<xs:enumeration value="Move" />
<xs:enumeration value="Match" />
<xs:enumeration value="Copy" />
<xs:enumeration value="Assign" />
<xs:enumeration value="Transform" />
<xs:enumeration value="Expression" />
<xs:enumeration value="Instance" />
</xs:restriction>
</xs:simpleType>
<xs:complexType name= "ArrayOfOpera- tion">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs= "unbounded" name="Operation" nillable="true" type="Operation" />
</xs:sequence>
</xs:complexType>
</xs:schema>
```

这个 Schema 中的主体是 Operations，即一个有序的操作数组，每个操作记录了操作所需要的值，值的类型，不同的操作值的含义是不同的，例如游标移动操作，值是整数，记录游标移动的距离；表达式计算操作值是一个字符串形式的表达式；对于实例化操作，值记录的是实例化的 UML 类的名称。

3 实时数据包分析

3.1 标准 GPS 通信信息类设计

在以上的自动机中每一个状态都包含一组 UML 对象的集合，数据流最终被转换成了这组 GPS 信息标准对象，图 2 给出了 GPS 标准对象的类图的主体部分。

其中 Protocol 描述了每个协议的抽象类，Position 是定位通信协议的数据要素；Business 是简化的业务数据通信协议内容；Status 是设备状态的信息类，Alarm 是报警信息，并且一般其中包括了定位

信息和状态信息。

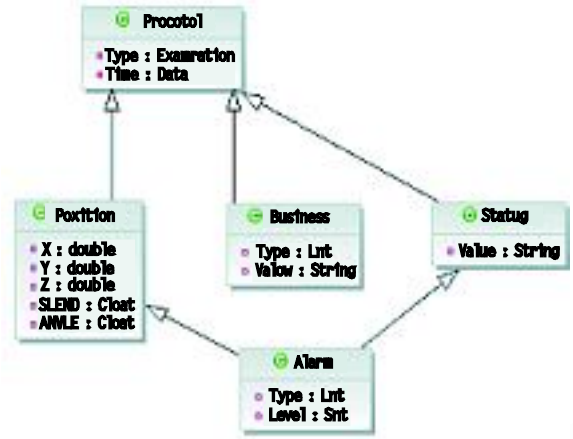

图 2 GPS 通信协议的 UML 模型

通过上例可以看出，图 2 定义的通信协议的 UML 模型不是一个完整的通信协议语义模型，它只说明了一个协议是由哪些元素组成的，但是没有说明组成的具体方式，即满足协议的数据包是如何生成的动态过程。

3.2 分析算法

通信协议的 UML 模型定义了一个通信协议的无结构组成要素；通信协议的自动机模型定义了一个通信协议的组成和分解过程。两者结合起来可以实现通信协议的自动分析，给定一个字节流，首先建立字节流的自动机模型，然后将自动机模型转换为 UML 模型，从而实现了将字节流转换成一个或者一组对象的过程。模型转换算法的伪代码如下：

```
void Transform(byte[] b,GPSProtocol p, Contex c)
{
if(!Match(b,p.startOperation))return; //如果起始与元素不匹配，返回
Cusors(START,o).Move(b,0);//移动起始游标到指定位置
foreach(Operation o in p.Operations) //遍历协议中的所有操作
{
//执行指定的操作
Object obj = o.Execute(b,c);
If(obj == null)
throw Exeception(c);
Cursors[START,0].Move(b,Assign(b,Cursors[0],child));
//如果已经判定了该数据包的类型，则递归调用 Transform，转换数据包的每个子元素
If(c.currentState==TYPE && obj.Type== c.currentMsg.Type)
Transform(b,o.Operations,c);;
//如果当前消息的 UML 对象的所有属性都已经正确赋值，则结束本次转换，、、、//currentMsg 就是将数据流转换后的对象
If(c.currentState==Assign && c.currentMsg.Check() || obj.Check())
return;
}
}
```

转换算法是严格按照自动机进行的，由于自动机模型中的状态包括了待生成对象 Object obj，通过不断的递归，指导所有的基本数据类型的属性都完成了赋值，字节流最终被转换成了对象。这个转换过程是与具体的协议无关的，只要通信协议事先给出了 UML 模型定义和状态机定义，这个转换算法都适用。

4 应用分析

在 GPS 通信系统中，移动设备发送的 GPS 通信数据具有多样性，即不同的移动设备发送的数据协议是不同的；同一个设备的通信数据又包括多种协议内容的。如果为每一种设备专门开发一种通信协议分析程序，开发的工作量会很大。通过通用的模型转换算法，可以实现不同的移动设备统一信息接入服务软件。

该算法和模型已经在城市特种车辆 GPS 定位与监控系统中得到了应用。由于特种车辆来自不同的管理单位，导致其上的 GPS 设备型号和协议有很大差别。传统上都是为每一个单位的设备单独做一个接入系统，最终在数据库端集成起来，这种做法涉及的软件很多，维护困难。本文的成果实现了用一套系统接入所有设备。

以图 2 的 UML 模型为例，应用此算法，可以同时将两种不同的数据流映射到 UML 对象，从而完成通信协议分析和转换。

协议 A: 00 73 C4 C8 71 26 01 0A 02 11 03 03 04 01 05 00…

协 议 B: $GPRMC,160019,A,3017.3639,N,

12003.798,E, 00,345.6,120800, ARM_V2.40,0*

其中协议 A 是一种二进制状态信息传输协议，上面用 16 进制表示；00 是起始标志；73 是包的有效长度;C4C87126 是设备编号，网络序表示；下面的每两个字节分别表示设备的某种状态，第一个字节是状态类型，第二个字节是状态值，例如“01 0A”，01 表示当前有效卫星；0A 表示卫星个数(16 进制)。

协议 B 是一种字节流定位传输协议。其中的“，”是分割了协议的每一个组成部分，$GPRMC 为起始标志,0*为结束标志，不固定长度，N 前面为纬度，E 前面为经度，“120800”分别表示卫星数、有效卫星数、报警 ID。

协议 A 和协议 B 两者看起来区别比较大，是两种表示信息完全不同、表达方法完全不同的协议，但是通过上节中的算法，可以将它们映射到相同的 UML 模型中，对于转换算法而说，两者区别只是在于自动机的状态转换过程不同，例如协议 A，在进入元素开始状态后，起始判定状态后就进入长度判定状态，而对于协议 B 起始状态判定需要先进入结束判定。

5 结论

建立了一种 GPS 通信协议的自动机模型，并设计了该模型的 XML Schema 和对应到 UML 对象的转化算法，实现了从无类型的字节流转换成 OO 语言的对象的自动转换,并在实际 GPS 通信系统中得到了应用，证明了其可行性。转换算法是否适用于 GPS 通信协议以外的协议，例如 TCP、SMTP 等，需要进一步的理论分析。

参考文献

1 余晓峰,余新宇.主/从分布式系统多机通信程序的状态机模型.计算技术与自动化, 2006,25(1):64－66.
2 许毅平,余霞,周曼丽.BACnet应用层测试状态机自动生成研究.微电子学与计算机, 2006,23(12):91－95.
3 王之梁,吴建平,尹霞.基于通信多端口有限状态机的协议互操作性测试生成研究.计算机学报, 2006, 29(11):1909－1919.
4 薛明.CSP 方式的安全协议建模研究.科技信息(学术版), 2008,(12):79－81.
5 陈华,付小青.安全协议的一种 CSP 开发框架.信息安全与通信保密, 2007,(6):124－125,128.