



(12)发明专利

(10)授权公告号 CN 106528823 B

(45)授权公告日 2019.08.30

(21)申请号 201611027224.7

(22)申请日 2016.11.18

(65)同一申请的已公布的文献号  
申请公布号 CN 106528823 A

(43)申请公布日 2017.03.22

(73)专利权人 中国银行股份有限公司  
地址 100818 北京市西城区复兴门内大街1号

(72)发明人 王同庆

(74)专利代理机构 北京三友知识产权代理有限公司 11127

代理人 郭晓宇

(51)Int.Cl.

G06F 16/81(2019.01)

G06F 16/83(2019.01)

(56)对比文件

CN 104572122 A,2015.04.29,

CN 106027330 A,2016.10.12,

CN 101697546 A,2010.04.21,

US 8392561 B1,2013.03.05,

审查员 王佩文

权利要求书2页 说明书12页 附图1页

(54)发明名称

一种报文配置及输出方法及装置

(57)摘要

本发明提供了一种报文配置及输出方法及装置。所述方法包括：使用XML文件进行报文配置，定义每个报文数据项的长度和先后顺序；根据所述报文数据项的长度和先后顺序输出报文。本发明实施例的报文配置及输出方法及装置，摒弃了现有的System.arraycopy方式，减少了开发人员工作量，且支持多种报文，应用广泛，提高了开发效率；另外，本发明摒弃了读位的方法，使用顺序输出报文的方法，能够快速定位出错位置，减少错误查找时间，减少报文测试时间，提高报文测试效率。



1. 一种报文配置及输出方法,其特征在于,包括:

使用XML文件进行报文配置,定义每个报文数据项的长度和先后顺序;

根据所述报文数据项的长度和先后顺序输出报文;

其中,使用XML文件进行报文配置,定义每个报文数据项的长度和先后顺序,包括:

使用XML文件分别定义报文头和报文体,其中,定义所述报文头,包括定义报文长度、交易码和产品信息,定义所述报文体,包括根据所述交易码定义报文数据项的长度和先后顺序;

使用不同关键字定义报文数据项的数据类型;

根据定义后的所述报文头、报文体以及报文数据项的数据类型组成数据流;

所述根据所述交易码定义报文数据项的长度和先后顺序,包括:

根据所述交易码确定报文体定义文件。

2. 根据权利要求1所述的报文配置及输出方法,其特征在于,使用不同关键字定义报文数据项的数据类型,包括:

使用String关键字定义报文数据项的数据类型,则所述报文数据项就按照字符型组装报文;

使用Number关键字定义报文数据项的数据类型,则所述报文数据项就按照数字型组装报文;

使用Date关键字定义报文数据项的数据类型,则所述报文数据项就按照日期型组装报文,并且按照pattern定义的模式对日期数据项进行格式化;

使用Amount关键字定义报文数据项的数据类型,则所述报文数据项就按照金额型组装报文,并且按照decimal定义的模式对金额数据项增加或减去小数点。

3. 根据权利要求1所述的报文配置及输出方法,其特征在于,根据所述报文数据项的长度和先后顺序输出报文,包括:

将报文格式定义工厂transformFactory的debug属性设置成true,则按照所述报文数据项的长度和先后顺序输出报文。

4. 一种报文配置及输出装置,其特征在于,包括:

报文配置模块,用于使用XML文件进行报文配置,定义每个报文数据项的长度和先后顺序;

输出模块,用于根据所述报文数据项的长度和先后顺序输出报文;

所述报文配置模块用于使用XML文件进行报文配置,定义每个报文数据项的长度和先后顺序,其具体包括:

报文格式定义单元,用于使用XML文件分别定义报文头和报文体;

数据类型定义单元,用于使用不同关键字定义报文数据项的数据类型;

数据流组成单元,用于根据定义后的所述报文头、报文体以及报文数据项的数据类型组成数据流;

所述报文格式定义单元用于使用XML文件分别定义报文头和报文体,其具体包括:

报文头定义单元,用于定义所述报文头,包括定义报文长度、交易码和产品信息;

报文体定义单元,用于定义所述报文体,包括根据所述交易码定义报文数据项的长度和先后顺序;所述根据所述交易码定义报文数据项的长度和先后顺序,包括:根据所述交易

码确定报文体定义文件。

5. 根据权利要求4所述的报文配置及输出装置,其特征在于,所述数据类型定义单元用于使用不同关键字定义报文数据项的数据类型,包括:

使用String关键字定义报文数据项的数据类型,则所述报文数据项就按照字符型组装报文;

使用Number关键字定义报文数据项的数据类型,则所述报文数据项就按照数字型组装报文;

使用Date关键字定义报文数据项的数据类型,则所述报文数据项就按照日期型组装报文,并且按照pattern定义的模式对日期数据项进行格式化;

使用Amount关键字定义报文数据项的数据类型,则所述报文数据项就按照金额型组装报文,并且按照decimal定义的模式对金额数据项增加或减去小数点。

6. 根据权利要求4所述的报文配置及输出装置,其特征在于,所述输出模块用于根据所述报文数据项的长度和先后顺序输出报文,包括:

将报文格式定义工厂transformFactory的debug属性设置成true,则按照所述报文数据项的长度和先后顺序输出报文。

## 一种报文配置及输出方法及装置

### 技术领域

[0001] 本发明涉及数据处理技术领域,尤其涉及一种报文配置及输出方法及装置。

### 背景技术

[0002] 本部分旨在为权利要求书中陈述的本发明的实施方式提供背景或上下文。此处的描述不因为包括在本部分中就承认是现有技术。

[0003] 目前常用的报文配置方法,都是使用System.arraycopy方法来组装报文,存在着技术复杂、容易出错、效率低下等缺点。

[0004] 以下面的一个组装报文操作为例:

[0005] System.arraycopy (retBytes,msgCdeOffset,msgCde,msgCdeStart,msgCdeLength); //将后台返回码的前9位放入字节数组;

[0006] retBytes为源字节数组报文,msgCdeOffset为源字节数组开始截取的位置,msgCde为目标字节数组报文,msgCdeStart为目标字节数组开始的位置,msgCdeLength为要截取的长度。

[0007] 上面的System.arraycopy的方法的含义为,从字节数组retBytes的msgCdeOffset位置开始,截取msgCdeLength长度的字节,放入msgCde字节数组的msgCdeStart开始位置。

[0008] 这种组装报文的方法十分复杂,操作也很繁琐,在开发人员的使用中非常容易出错,而且很容易造成开发人员的返工,因为报文都是一项一项的组装的,如果中间任意一项出错,都要从做开始进行排查,逐项排除错误,如果是第一项有问题,那么所有的报文项都要重写一遍。

[0009] 另外,目前的报文测试方法也很原始,耗费开发人员的体力,浪费开发人员的时间。以下面的一个5项报文内容为例,付款金额123.56元,报文格式如下表1所示:

[0010] 表1

[0011]

报文数据项-参数	长度	报文数据项-含义	数据类型	备注
payerAcct	9 (17)	付款账号	数字型	
payeeAcct	9 (17)	收款账号	数字型	
currency	X (3)	付款币种	字符型	CNY
tranAmount	9 (15)	付款金额	金额型	
tranDate	X (10)	付款日期	日期型	yyyy-MM-dd

[0012] 发送报文内容如下:

[0013] 0000010111234567800000101356789012CNY000000000123562016-03-21

[0014] 如果第4项的付款金额报文有误,开发人员在排查时,要根据报文数据项的长度进行读位,逐个排查:

[0015] 付款账号17位:00000101112345678

[0016] 收款账号17位:00000101356789012

[0017] 付款币种3位:CNY

[0018] 付款金额15位:000000000123562

[0019] 付款日期10位:016-03-21

[0020] 经过读位操作,发现日期少了1位,金额的报文也不对,经过比对金额123.56元,发现付款金额少了一个0。

[0021] 因此,如何快速定位报文出错的位置,并对报文进行快速、高效的测试,也是急需解决的一大问题。

## 发明内容

[0022] 本发明提供一种报文配置及输出方法,以解决了现有的System.arraycopy方法带来的弊端。

[0023] 为了解决上述问题,本发明实施例提供一种报文配置及输出方法,包括:

[0024] 使用XML文件进行报文配置,定义每个报文数据项的长度和先后顺序;根据所述报文数据项的长度和先后顺序输出报文。

[0025] 进一步地,在一实施例中,使用XML文件进行报文配置,定义每个报文数据项的长度和先后顺序,包括:使用XML文件分别定义报文头和报文体;使用不同关键字定义报文数据项的数据类型;根据定义后的所述报文头、报文体以及报文数据项的数据类型组成数据流。

[0026] 进一步地,在一实施例中,使用XML文件分别定义报文头和报文体,包括:定义所述报文头,包括定义报文长度、交易码和产品信息;定义所述报文体,包括根据所述交易码定义报文数据项的长度和先后顺序。

[0027] 进一步地,在一实施例中,使用不同关键字定义报文数据项的数据类型,包括:使用String关键字定义报文数据项的数据类型,则所述报文数据项就按照字符型组装报文;使用Number关键字定义报文数据项的数据类型,则所述报文数据项就按照数字型组装报文;使用Date关键字定义报文数据项的数据类型,则所述报文数据项就按照日期型组装报文,并且按照pattern定义的模式对日期数据项进行格式化;使用Amount关键字定义报文数据项的数据类型,则所述报文数据项就按照金额型组装报文,并且按照decimal定义的模式对金额数据项增加或减去小数点。

[0028] 进一步地,在一实施例中,根据所述报文数据项的长度和先后顺序输出报文,包括:将报文格式定义工厂transformFactory的debug属性设置成true,则按照所述报文数据项的长度和先后顺序输出报文。

[0029] 为了解决上述问题,本发明实施例还提供一种报文配置及输出装置,包括:报文配置模块,用于使用XML文件进行报文配置,定义每个报文数据项的长度和先后顺序;输出模块,用于根据所述报文数据项的长度和先后顺序输出报文。

[0030] 进一步地,在一实施例中,所述报文配置模块用于使用XML文件进行报文配置,定义每个报文数据项的长度和先后顺序,其具体包括:报文格式定义单元,用于使用XML文件分别定义报文头和报文体;数据类型定义单元,用于使用不同关键字定义报文数据项的数据类型;数据流组成单元,用于根据定义后的所述报文头、报文体以及报文数据项的数据类型组成数据流。

[0031] 进一步地,在一实施例中,所述报文格式定义单元用于使用XML文件分别定义报文头和报文体,其具体包括:报文头定义单元,用于定义所述报文头,包括定义报文长度、交易码和产品信息;报文体定义单元,用于定义所述报文体,包括根据所述交易码定义报文数据项的长度和先后顺序。

[0032] 进一步地,在一实施例中,所述数据类型定义单元用于使用不同关键字定义报文数据项的数据类型,包括:使用String关键字定义报文数据项的数据类型,则所述报文数据项就按照字符型组装报文;使用Number关键字定义报文数据项的数据类型,则所述报文数据项就按照数字型组装报文;使用Date关键字定义报文数据项的数据类型,则所述报文数据项就按照日期型组装报文,并且按照pattern定义的模式对日期数据项进行格式化;使用Amount关键字定义报文数据项的数据类型,则所述报文数据项就按照金额型组装报文,并且按照decimal定义的模式对金额数据项增加或减去小数点。

[0033] 进一步地,在一实施例中,所述输出模块用于根据所述报文数据项的长度和先后顺序输出报文,包括:将报文格式定义工厂transformFactory的debug属性设置成true,则按照所述报文数据项的长度和先后顺序输出报文。

[0034] 本发明实施例的报文配置及输出方法及装置,摒弃了现有的System.arraycopy方式,减少了开发人员工作量,且支持多种报文,应用广泛,提高了开发效率;另外,本发明摒弃了读位的方法,使用顺序输出报文的方法,能够快速定位出错位置,减少错误查找时间,减少报文测试时间,提高报文测试效率。

## 附图说明

[0035] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0036] 图1为本发明实施例的报文配置及输出方法的处理流程图;

[0037] 图2为本发明实施例的报文配置及输出装置的结构示意图;

[0038] 图3为图2所示实施例中的报文配置模块的结构示意图。

## 具体实施方式

[0039] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0040] 本领域技术人员知道,本发明的实施方式可以实现为一种系统、装置、设备、方法或计算机程序产品。因此,本公开可以具体实现为以下形式,即:完全的硬件、完全的软件(包括固件、驻留软件、微代码等),或者硬件和软件结合的形式。

[0041] 下面参考本发明的若干代表性实施方式,详细阐释本发明的原理和精神。

[0042] 本发明提出的报文配置与输出方法的主要思想是:使用XML文件进行报文配置,在XML文件中定义每个报文数据项的长度和先后顺序,报文定义清晰、简单;对不同的数据类

型,定义不同的关键字,支持字符型、数字型、日期型、金额型等数据项,使用include关键字,报文头和报文体可分别定义,报文配置方便,能大大减少开发人员的工作量,减少人工劳动;并且,本发明支持定长报文、变长报文、XML报文、8583报文、混合报文(定长+变长+XML)的配置和测试,涵盖范围广,能够帮助开发人员进行各种报文的配置和测试,提高开发效率。

[0043] 另外,本发明提供的顺序打印报文配置项的技术,能够根据报文定义的先后顺序和长度打印报文,一目了然的让开发人员看到所有的报文项,能够快速定位出错报文的位置,提高报文测试的效率。因此,如果两个系统间对报文的正确性存在疑问,对于报文上送错误,本发明可以提供发送的报文,对于收到的报文异常,可以解析收到的报文,在技术方案阶段对接口定义进行技术验证。在无法获得debug等详细测试信息时,本发明可以测试、检验生产报文的异常(报文异常结束、报文超长的的问题、报文错位、报文数据项非法)。

[0044] 图1为本发明实施例的报文配置及输出方法的处理流程图。如图1所示,包括:

[0045] 步骤S101,使用XML文件进行报文配置,定义每个报文数据项的长度和先后顺序;步骤S102,根据所述报文数据项的长度和先后顺序输出报文。

[0046] 在本实施例中的步骤S101中,使用XML文件进行报文配置,定义每个报文数据项的长度和先后顺序,包括:步骤S1011,使用XML文件分别定义报文头和报文体;步骤S1012,使用不同关键字定义报文数据项的数据类型;步骤S1013,根据定义后的所述报文头、报文体以及报文数据项的数据类型组成数据流。

[0047] 其中,步骤S1011和步骤S1012并没有先后顺序关系,两个步骤可以同时执行。

[0048] 在本实施例中的步骤S1011中,使用XML文件分别定义报文头和报文体,包括:定义所述报文头,包括定义报文长度、交易码和产品信息;以及定义所述报文体,包括根据所述交易码定义报文数据项的长度和先后顺序。

[0049] 一般系统都会定义报文头与报文体,报文头一般包括报文长度、交易码、产品信息等公共信息,报文体包含具体某个交易的信息。

[0050] 以下面的报文头定义文件DemoFormatPacket.xml为例,对于报文头定义,如表2所示,包括报文长度+交易码,而对于报文体定义,具体根据交易码不同:

[0051] 表2

[0052]

报文头数据项-参数	长度	报文头数据项-含义	数据类型	备注
DATA_LEN	9(4)	报文长度	数字型	
TRAN_CODE	X(5)	交易码	字符型	

<segment>

<fixNumber name="DATA\_LEN" length="4" add="5"></fixNumber>

[0053]

<fixString name="TRAN\_CODE" length="5"></fixString> </segment>

<include keyName="TRAN\_CODE"></include>

[0054] 上面的报文配置中,DATA\_LEN代表报文长度,程序会计算出交易码长度+报文体的长度放到DATA\_LEN对应的数据域中,TRAN\_CODE代表交易码,当交易码为D0002时,通过

include关键字和前缀DemoFormat就会找到报文体定义文件DemoFormatD0002.xml，DemoFormatD0002.xml为报文体定义文件，对应了D0002交易：

[0055]

```

<segment>
  <fixNumber name="payerAcct" length="17" /><!--付款账号 -->
  <fixNumber name="payeeAcct" length="17" /><!--收款账号 -->
  <fixString name="currency" length="3" /><!--付款币种 -->
  <fixAmount name="tranAmount" length="15" decimal="2"/><!--付款金额 -->
  <fixDate name="tranDate" pattern="yyyy-MM-dd" /><!--付款日期 yyy-MM-dd
-->
</segment>

```

[0056] 在上面的XML报文配置文件中，对应下面的表3的“上送报文格式定义”，程序首先读取报文头DemoFormatPacket.xml配置文件，然后读取报文体DemoFormatD0002.xml报文配置文件，在两个配置文件中，定义了报文数据项的先后顺序和长度，程序根据报文数据项的先后顺序和长度组装报文，还是以下面的一个5项报文内容为例，付款金额123.56元。

[0057] 表3

[0058]

报文数据项-参数	长度	报文数据项-含义	数据类型	备注
payerAcct	9 (17)	付款账号	数字型	
payeeAcct	9 (17)	收款账号	数字型	
currency	X (3)	付款币种	字符型	CNY
tranAmount	9 (15)	付款金额	金额型	
tranDate	X (10)	付款日期	日期型	yyyy-MM-dd

[0059] 程序根据XML文件的定义，组装报文如下：

[0060]

```

// -----格式化上送数据-----
sendDatas:00620000010111234567800000101356789012CNY000000000012356201

```

6-03-21

```

format->payerAcct=101112345678
format->payeeAcct=101356789012
format->currency=CNY
format->tranAmount=123.56
format->tranDate=2016-03-21

```

[0061] 如果其中一项发生了变化，只要修改XML文件配置即可，简单易行，开发人员也易于理解，极大的提高了开发的效率。以金额变为16位为例：

[0062]

```
<fixAmount name="tranAmount" length="16" decimal="2"/><!--付款金额 -->
// -----格式化上送数据-----
sendDats:0063D00020000010111234567800000101356789012CNY000000000001
23562016-03-21
format->payerAcct=101112345678
format->payeeAcct=101356789012
format->currency=CNY
format->tranAmount=123.56
format->tranDate=2016-03-21
format->DATA_LEN=63
```

[0063] format->TRAN\_CODE=D0002

[0064] 在本实施例中的步骤S1012中,使用不同关键字定义报文数据项的数据类型,即:对不同的数据类型,定义不同的关键字,本发明中支持字符型、数字型、日期型、金额型等数据项。

[0065] 1、字符型为String关键字,无论是变长还是定长定义,只要是包含String关键字,这个数据项就按照字符型组装报文;

[0066] 2、数字型为Number关键字,无论是变长还是定长定义,只要是包含Number关键字,这个数据项就按照数字型组装报文,如果数据项内容为非数字型,程序就会报错提示开发人员;

[0067] 3、日期型为Date关键字,无论是变长还是定长定义,只要是包含Date关键字,这个数据项就按照日期型组装报文,并且按照pattern定义的模式可以对日期数据项进行格式化;

[0068] 4、金额型为Amount关键字,无论是变长还是定长定义,只要是包含Amount关键字,这个数据项就按照金额型组装报文,并且按照decimal定义的模式可以对金额数据项增加或减去小数点。

[0069] 本实施例的步骤S1013中,在完成报文头、报文体以及数据项的配置后,则根据定义后的所述报文头、报文体以及报文数据项的数据类型组成数据流。

[0070] 本发明支持定长报文、变长报文、XML报文、8583报文、混合报文(定长+变长+XML)。

[0071] 1、定长报文处理

[0072] 定长报文顾名思义就是报文的长度是固定的,不因为每次交易内容的变化报文长度而改变。例如接口定义如下表4所示,输入:

[0073] 表4

[0074]

序号	字段	类型	说明
1	DATA_LEN	9(4)	报文长度
2	TRAN_CODE	X(5)	交易码 D0001
3	payerAcct	X(17)	付款账号 F123456
4	payeeAcct	X(17)	收款账号 R123456

[0075]

5	currency	X(3)	币种 CNY
6	tranAmount	9(13)V9(2)	交易金额 100.00

[0076] 则生成的报文如下：

[0077] 0057D0001F1234567890 R1234567890 CNY000000000010000

[0078] 账号和金额变化后,报文长度不变：

[0079] 0057D0001F2345678901 R1234567890 CNY000000000020000

[0080] 定长报文的优点是格式固定、长度可控,缺点是会产生空格等冗余,浪费通讯空间。

[0081] 对于定长报文的设置,在XML文件中都是以fix开头的,例如,&lt;fixStringname="payerAcct"length="17"/&gt;代表17位固定长度的账号报文,程序读取XML文件的定义,如果遇到fixString就按照定长报文处理。

[0082] 2、变长报文处理

[0083] 变长报文就是每次交易内容的变化报文长度会改变。变长报文有两种形式,一种是每个数据项都定义长度和内容,另外一种数据项之间用分隔符分隔。

[0084] 第一种形式的报文如下：

[0085] 0051005D0001010F1234567890010R1234567890003CNY005010000

[0086] 第二种形式的报文如下：

[0087] 0041\$D0001\$F1234567890\$R1234567890\$CNY\$010000

[0088] 变长报文的优点是格式灵活、节省数据空间,缺点是对报文的内容不可控。

[0089] 对于变长报文的设置,在XML文件中都是以var开头的,例如,&lt;varStringname="KEY"type="LL"&gt;&lt;/varString&gt;代表用两个十位数表示的变长报文,【0512345】表示KEY为收到的报文,05代表报文的长度,12345表示KEY的值,程序读取XML文件的定义,如果遇到varString就按照变长报文处理。

[0090] 3、XML报文处理

[0091] XML报文就是报文的内容为XML文档格式。

[0092] 对于XML报文的设置,在XML文件中都是以xmlTag开头的,例如,&lt;&lt;xmlTagformat="true"&gt;&lt;Number name="AC\_CNT"length="4"option="false"&gt;&lt;/Number&gt;&lt;/xmlTag&gt;代表这个报文数据项要按照XML报文标准进行组装,组装形式为数字型,xmlTag的名字为AC\_CNT,组装后为&lt;AC\_CNT&gt;0123456789&lt;/AC\_CNT&gt;;程序读取XML文件的定义,如果遇到

xmlTag就按照XML报文处理。

[0093] <xmlTag format="true">

[0094] <Number name="AC\_CNT"length="4"option="false"></Number>

[0095] </xmlTag><!--客户号-->

[0096] 在上送数据时,会出现上送XML文件格式数据的情况,数据按XML标准格式定义,在报文的前后增加XML报文头和XML报文末的定义,XML报文定义如下:

[0097] XML报文头定义+XML报文体定义+XML报文末定义。

[0098] 4、ISO8583报文处理

[0099] 对于iso8583报文的设置,在XML文件中都是以iso8583Fields开头的,如下面的报文定义,bitMap定义了这段报文包含8583报文的哪几个域,在iso8583Fields定义的子集中就定义了每个8583报文域的具体格式。

[0100] <iso8583Fields bitMap="7,39,63,70"beginOffset="0">

[0101] 上面的内容代表此类报文为ISO8583报文,包含的BITMAP为7,39,63,70等4项;

[0102] <!--7-Transmission Date&Time N10 (MMDDhhmmss) 交易时间-->

[0103] <fixBullBCD name="TIME"length="10"></fixBullBCD>

[0104] 上面的内容代表此报文为ISO8583报文中的第7域,为10位的时间数据项;

[0105] </iso8583Fields>

[0106] 5、混合报文处理

[0107] 对于混合报文(定长+变长+XML)的处理设置,就是Baffle挡板系统宝收到的数据分为报文头和报文体两部分,报文头和包问题的报文格式可以不同,他们的组合方式可以为:1) 报文头(定长报文)+报文体(变长报文);2) 报文头(定长报文)+报文体(XML报文);3) 报文头(变长报文)+报文体(XML报文)。

[0108] 在本实施例中的步骤S103中,根据所述报文数据项的长度和先后顺序输出报文,其方法为:将报文格式定义工厂transformFactory的debug属性设置成true,就可以支持顺序打印报文配置项和报文长度等内容:

[0109] <transformerFactory id="TransformerFactory">

[0110] <param name="debug">true</param>

[0111] </transformerFactory>

[0112] 本发明提所供的顺序输出报文配置项的技术,能够根据报文定义的先后顺序和长度打印报文,一目了然的让开发人员看到所有的报文项,能够快速定位出错报文的位置,提高报文测试的效率。

[0113] 程序会读取debug标志位,如果为true,程序就会按照XML配置文件定义的先后顺序和长度在控制台上输出报文内容,如下:

[0114]

```

// -----格式化上送数据-----
sendData:0063D00020000010111234567800000101356789012CNY000000000001
23562016-03-21
format->payerAcct=101112345678
format->payeeAcct=101356789012
format->currency=CNY
format->tranAmount=123.56
format->tranDate=2016-03-21
format->DATA_LEN=63
format->TRAN_CODE=D0002

```

[0115] 在测试或生产环境,往往只输出报文内容,而不会输出每个报文配置项的内容。如果两个系统间对报文的正确性存在疑问,对于报文上送错误,本发明可以提供发送的报文,对于收到的报文异常,可以解析收到的报文。在无法获得debug等详细测试信息时,只要提供报文内容,本发明就可以快速定位非法的数据项。

[0116] 对于客户端发送了报文,服务器端也收到了报文,但是服务器端返回报文非法错误,客户端就要提供发送的报文给服务器端,快速定位非法的数据项。

[0117] 例如,提供报文内容如下:

```

String resData=new String("0182
0000010111234567800000101356789012CNY000000000012356A1111A1111
[0118] A2222A2222          01B2222B2222
");

```

[0119] 本发明根据报文的定义如下:

```

<segment>
  <fixDate name="date" pattern="yyyyMMdd" />
  <fixNumber name="payerAcct" length="17" />
[0120] <fixNumber name="payeeAcct" length="17" />
  <fixString name="currency" length="3" />
  <fixAmount name="tranAmount" length="15" />
</segment>

```

[0121] 测试报文数据项如下:

```
// -----解析返回数据-----0182
0000010111234567800000101356789012CNY000000000012356A1111A1111
A2222A2222                01B2222B2222
```

DATA\_LEN=182

[0122] date=null  
payerAcct=101112345678  
payeeAcct=101356789012  
currency=CNY  
tranAmount=123.56

[0123] 这样两个系统就可以根据测试的输出结果来进行排查,迅速找出出错的位置和报文数据项。

[0124] 本发明还可以测试、检验生产报文的异常(例如报文异常结束、报文超长的问题)。

[0125] 因为本发明会输出报文长度,只要把输出的报文长度与报文定义的长度进行对比,就可以发现报文异常结束、报文超长的问题。

[0126] 例如报文定义如下,正常的长度应改为67,如下表5所示:

[0127] 表5

[0128]

序号	字段	类型	说明
1	DATA_LEN	9(4)	报文长度
2	TRAN_CODE	X(5)	交易码 D0001
3	payerAcct	X(17)	付款账号 F123456
4	payeeAcct	X(17)	收款账号 R123456

[0129]

5	currency	X(3)	币种 CNY
6	tranAmount	9(13)V9(2)	交易金额 100.00

[0130] 输出报文如下:

[0131] sendDatas:0068D00020000010111234567800000101356789012CNY00000000000123562016-03-21;报文长度为0068,说明报文超长。

[0132] 应当注意,尽管在附图中以特定顺序描述了本发明方法的操作,但是,这并非要求或者暗示必须按照该特定顺序来执行这些操作,或是必须执行全部所示的操作才能实现期望的结果。附加地或备选地,可以省略某些步骤,将多个步骤合并为一个步骤执行,和/或将一个步骤分解为多个步骤执行。

[0133] 在介绍了本发明示例性实施方式的方法之后,接下来,参考图2对本发明示例性实施方式的报文配置及输出装置进行介绍。该装置的实施可以参见上述方法的实施,重复之

处不再赘述。以下所使用的术语“模块”和“单元”，可以是实现预定功能的软件和/或硬件。尽管以下实施例所描述的模块较佳地以软件来实现，但是硬件，或者软件和硬件的组合的实现也是可能并被构想的。

[0134] 图2为本发明实施例的报文配置及输出装置的结构示意图。如图2所示，包括：

[0135] 报文配置模块101，用于使用XML文件进行报文配置，定义每个报文数据项的长度和先后顺序；输出模块102，用于根据所述报文数据项的长度和先后顺序输出报文。

[0136] 进一步地，在本实施例中，如图3所示，所述报文配置模块101用于使用XML文件进行报文配置，定义每个报文数据项的长度和先后顺序，其具体包括：

[0137] 报文格式定义单元1011，用于使用XML文件分别定义报文头和报文体；

[0138] 数据类型定义单元1012，用于使用不同关键字定义报文数据项的数据类型；

[0139] 数据流组成单元1013，用于根据定义后的所述报文头、报文体以及报文数据项的数据类型组成数据流。

[0140] 进一步地，在本实施例中，所述报文格式定义单元1011用于使用XML文件分别定义报文头和报文体，其具体包括：报文头定义单元，用于定义所述报文头，包括定义报文长度、交易码和产品信息；以及报文体定义单元，用于定义所述报文体，包括根据所述交易码定义报文数据项的长度和先后顺序。

[0141] 进一步地，在本实施例中，所述数据类型定义单元1012用于使用不同关键字定义报文数据项的数据类型，包括：使用String关键字定义报文数据项的数据类型，则所述报文数据项就按照字符型组装报文；使用Number关键字定义报文数据项的数据类型，则所述报文数据项就按照数字型组装报文；使用Date关键字定义报文数据项的数据类型，则所述报文数据项就按照日期型组装报文，并且按照pattern定义的模式对日期数据项进行格式化；使用Amount关键字定义报文数据项的数据类型，则所述报文数据项就按照金额型组装报文，并且按照decimal定义的模式对金额数据项增加或减去小数点。

[0142] 进一步地，在本实施例中，所述输出模块103用于根据所述报文数据项的长度和先后顺序输出报文，包括：将报文格式定义工厂transformFactory的debug属性设置成true，则按照所述报文数据项的长度和先后顺序输出报文。

[0143] 本发明实施例提出的报文配置与输出方法，使用XML文件进行报文配置，在XML文件中定义每个报文数据项的长度和先后顺序，报文定义清晰、简单；对不同的数据类型，定义不同的关键字，支持字符型、数字型、日期型、金额型等数据项，使用include关键字，报文头和报文体可分别定义，报文配置方便，能大大减少开发人员的工作量，减少人工劳动；并且，本发明支持定长报文、变长报文、XML报文、8583报文、混合报文（定长+变长+XML）的配置和测试，涵盖范围广，能够帮助开发人员进行各种报文的配置和测试，提高开发效率。另外，本发明提供的顺序打印报文配置项的方法，能够根据报文定义的先后顺序和长度打印报文，一目了然的让开发人员看到所有的报文项，能够快速定位出错报文的位置，提高报文测试的效率。

[0144] 本领域内的技术人员应明白，本发明的实施例可提供为方法、系统、或计算机程序产品。因此，本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产

品的形式。

[0145] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0146] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0147] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0148] 本发明中应用了具体实施例对本发明的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本发明的方法及其核心思想;同时,对于本领域的一般技术人员,依据本发明的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本发明的限制。

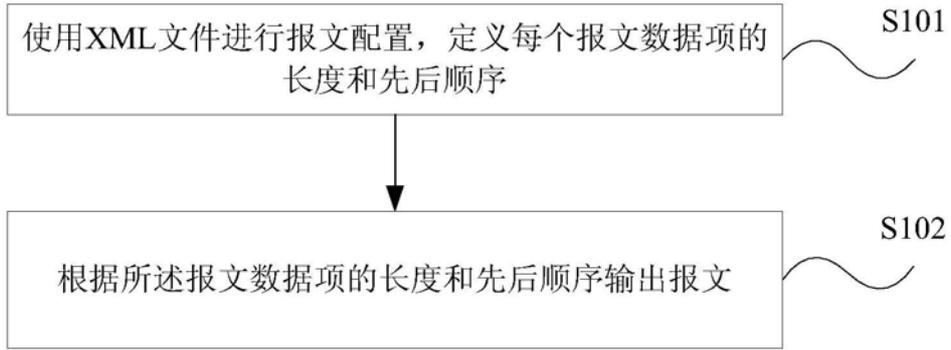


图1



图2

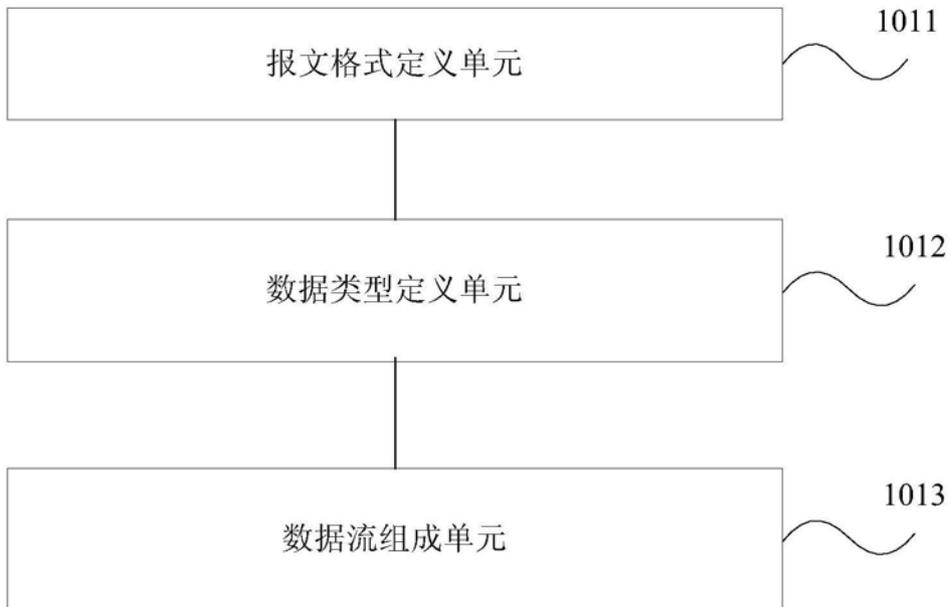


图3