

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局



(43) 国际公布日
2012年3月22日 (22.03.2012)

PCT

(10) 国际公布号
WO 2012/034477 A1

- (51) 国际专利分类号:
G06F 17/00 (2006.01)
- (21) 国际申请号: PCT/CN2011/079035
- (22) 国际申请日: 2011年8月29日 (29.08.2011)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
201010283713.5 2010年9月14日 (14.09.2010) CN
- (71) 申请人 (对除美国外的所有指定国): 腾讯科技 (深圳) 有限公司 (TENCENT TECHNOLOGY (SHENZHEN) COMPANY LIMITED) [CN/CN]; 中国广东省深圳市福田区振兴路赛格科技园2栋东4楼, Guangdong 518044 (CN)。
- (72) 发明人: 及
- (75) 发明人/申请人 (仅对美国): 彭勇 (PENG, Yong) [CN/CN]; 中国广东省深圳市福田区振兴路赛格科技园2栋东4楼, Guangdong 518044 (CN)。
- (74) 代理人: 北京德琦知识产权代理有限公司 (DEQI INTELLECTUAL PROPERTY LAW CORPORATION); 中国北京市海淀区知春路1号学院国际大厦7层, Beijing 100083 (CN)。
- (81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。
- (84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG)。

本国际公布:

- 包括国际检索报告(条约第21条(3))。

(54) Title: METHOD AND DEVICE FOR DATA STORAGE

(54) 发明名称: 一种存储数据的方法及装置

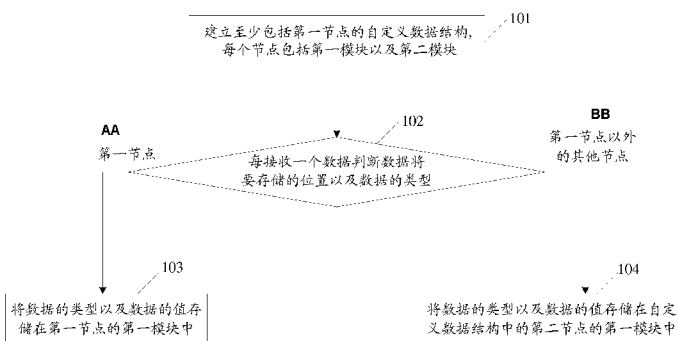


图 1 / FIG. 1

- AA First node
BB Other nodes besides the first node
101 Establish a self-defined data structure including at least a first node, with each node including a first module and a second module
102 Each time a datum is received, judge a location to store the datum and the type of the datum
103 Store the type and value of the datum in the first module of the first node
104 Store the type and value of the datum in the first module of the second node of the self-defined data structure

(57) Abstract: Provided are a method and device for data storage. The method comprises: establishing a self-defined data structure including at least a first node as the root node, wherein each node includes a first module for storing a type identifier and value of a datum, and a second module for storing a mapping list between identifiers of children nodes of a node and the children nodes; each time a datum is received, judging a location to store the datum and the type of the received datum; storing the type and value of the datum in the first module of the first node if the datum is stored in the first node; and storing the type and value of the datum in the first module of the second node if the datum is stored in the second node of the self-defined data structure. According to the method and device, conversion of different data types, and reading and transmission of data can be carried out conveniently.

[见续页]



WO 2012/034477 A1



(57) 摘要:

提供一种存储数据的方法和装置。该方法包括：建立至少包括作为根节点的第一节点的自定义数据结构，其中，每个节点包括第一模块，用于存储数据的类型标识和值，以及第二模块，用于存储节点的子节点的标识与子节点之间的映射列表；每接收一个数据，判断数据将要存储的位置以及接收的数据的类型；如果数据存储于第一节点，将数据的类型以及数据的值存储在第一节点的第一模块中；如果数据存储于自定义数据结构的第二节点，将数据的类型以及数据的值存储在第二节点的第一模块中。根据该方法和装置，可以方便地进行不同数据类型的转换、对数据进行读取和传递。

一种存储数据的方法及装置

技术领域

本发明涉及计算机技术领域，具体涉及一种存储数据的方法及装置。

发明背景

5 随着计算机技术和网络技术的发展，出现了各种各样的基于计算机的通讯系统，比如即时通讯 IM 系统，微博通讯系统等。在这类通讯系统中，需要对系统的数据进行存储以及使用，以实现系统的各种功能。其中数据的存储和使用可以通过各种程序语言来实现。在程序语言中，一般需要将数据按照一定的类型进行存储，并且编译器在编译阶段（尤
10 其是 C++ 这种强类型语言中）会对数据的类型安全进行严格的检查，不同的数据类型，是不能够互相转换、赋值的。在此情况下，当存储一些稍微复杂一点的数据时，往往需要使用 stl 结合一些自定义数据结构来存储。而对于不同场景下的数据，往往又需要定义不同的数据结构。因此，采用这种方法存储的数据使用起来比较麻烦。

15 基于上述原因，出现一些改进的存储方法，以实现不同数据类型的灵活使用，比如转换、读取或传递等。例如，在脚本语言中，在存储数据时，使用 VARIANT 结构来表示数据类型。然而，VARIANT 结构中，一次只能有一种类型是有效的，也就是每次只能处理一个数据，没办法既存储数字，又存储字符串，不能批量处理数据。

发明内容

有鉴于此，本发明实施例提供了一种存储数据的方法及装置，该方

法及装置可以处理批量数据。

根据本发明实施例，一种存储数据的方法包括：

建立至少包括作为根节点的第一节点的自定义数据结构，其中，该自定义数据结构的每个节点包括第一模块，用于存储数据的类型标识和
5 值，以及第二模块用于存储该节点的子节点的标识与子节点之间的映射列表；

每接收一个数据判断数据将要存储的位置以及接收的数据的类型；

如果数据存储于第一节点，将数据的类型以及数据的值存储在第一节
10 节点的第一模块中；

如果数据存储于自定义数据结构的第二节点，将数据的类型以及数
据的值存储在第二节点的第一模块中。

根据本发明另一实施例，一种存储数据的装置包括：

数据接口，用于接收数据；

通用数据类型组件，用于建立并保存至少包括作为根节点的第一节
15 点的自定义数据结构，其中，该自定义数据结构的每个节点包括第一模
块，用于存储数据的类型标识和值，以及第二模块用于存储该节点的子
节点的标识与子节点之间的映射列表；每当数据接口接收一个数据，判
断数据将要存储的位置以及数据的类型；如果数据存储于第一节点，则
将数据的类型以及数据的值存储在第一节的第一模块中；如果数据存
20 储于自定义数据结构的第二节点中，则将数据的类型以及数据的值存储
在第二节点的第一模块中。

根据本发明又一实施例，一种通用数据类型组件包括：节点建立模
块，判断模块和存储模块；

节点建立模块，用于建立并保存至少包括作为根节点的第一节点的
25 自定义数据结构，其中，自定义数据结构的每个节点包括第一模块，用

于存储数据的类型标识和值，以及第二模块用于存储该节点的子节点的标识与子节点之间的映射列表；

判断模块，用于每接收一个数据后，判断数据将存储于自定义数据结构的第一节点或第二节点，且判断数据的类型，并将判断结果通知给
5 存储模块；

存储模块，用于在判断模块判断数据将存储于第一节点时，将数据的类型以及数据的值存储在第一节的第一模块中；在判断模块判断数据将存储于自定义数据结构的第二节点时，将数据的类型以及数据的值存储在第二节点的第一模块中。

10 与现有技术相比，本发明实施例的自定义数据结构定义了一种通用数据类型；当将数据存储为该通用数据类型时，该通用数据类型可以判断待存储数据的基本类型，并将待存储数据的基本类型和值进行存储。而且，由于在一个自定义数据结构定义的通用数据类型中，每个节点都可以设有其子节点的<标识，子节点>的映射列表，因此，可以将大量的
15 数据存储于该通用数据类型中。这种情况下，只要将大量的数据存储成一个通用数据类型，则使用该通用数据类型定义的一个变量就可以存储大量的数据。并且，由于这种通用数据类型中各节点间存储的数据类型可以不同，因此通用数据类型可以存储不同类型的数据，则使用该通用数据类型定义的一个变量就可以表示大量的具有不同类型的数据。

20 并且，由于通用数据类型的变量中各个节点都有各自的类型标识，因此各个节点可以存储不同类型的数据，而不需要存储兼容类型的数据，比如：可以存储 LONG 和 int 数据，也可以存储 BSTR 字符串数据，等。

附图简要说明

- 图 1 为根据本发明实施例的一种存储数据的方法的流程图；
图 2 为本发明实施例中存储数据的存储模型；
图 3 示出了本发明实施例中存储数据的通用数据类型的树状结构；
5 图 4 为本发明实施例的一种存储数据的装置；
图 5 为本发明另一实施例的一种存储数据的装置。

实施本发明的方式

为使本发明的目的、技术方案和优点更加清楚明白，以下举具体实施例并参照附图，对本发明作进一步详细说明。

- 10 图 1 为根据本发明实施例的一种存储数据的方法的流程图。如图 1 所示，该方法包括：

步骤 101：建立至少包括第一节点的自定义数据结构。

- 在本步骤中，该第一节点为根节点。每个节点包括第一模块，用于存储数据的类型标识和值，以及第二模块用于存储该节点的子节点的标识与其子节点之间的映射列表。
15

为了描述方便，本发明将这种自定义数据结构定义的数据类型称为通用数据类型（记为 WData）。

步骤 102：每接收一个数据判断数据将要存储的位置以及数据的类型。

- 20 本步骤中，当用户为具有自定义数据结构的变量赋值时，则自定义数据结构接收到数据。在此，用户为变量赋值相当于将数据存储于该变量中；数据的类型可以是任何的基本类型，比如整型(int)，浮点型(float)，字节型(byte)等。

步骤 103: 如果数据存储于第一节点, 则将数据的类型以及数据的值存储在第一节点的第一模块中。

本步骤中, 当数据存储于第一节点, 即存储于根节点时, 该根节点无需对应一个节点标识与节点之间的映射列表, 即<key, WData 节点>(以下简称<key, WData>)。并且, 当自定义数据结构中仅有第一节点, 则第一节点中的第二模块的映射列表为空。当第一节点存在子节点时, 则第一节点中的第二模块的映射列表记录第一节点的子节点标识与子节点之间的映射列表<key, WData>。

步骤 104: 如果数据存储于自定义数据结构的其他节点(称为第二节点), 则将数据的类型以及数据的值存储在第二节点的第一模块中。

本步骤中, 当数据存储于第二节点时, 由于第二节点为子节点, 则该数据属于子数据。此时, 第二节点对应一个节点标识与节点之间的映射列表, 比如<key2, WData>。该映射列表存储于第二节点的父节点中, 用于索引父节点的子节点, 从而索引子节点中的数据。

在步骤 104 之前, 还包括: 判断自定义数据结构中是否包括第二节点; 如果没有, 则建立所述自定义数据结构的第二节点; 否则, 直接将数据的类型以及数据的值存储在所述第二节点的第一模块中。

建立自定义数据结构的第二节点可以包括: 在第二节点的父节点的第二模块中填写第二节点的标识与第二节点之间的映射列表。

同理, 当接收到存储于第二节点的子节点的数据时, 在第二节点中的第二模块填写子节点的标识与子节点之间的映射列表。

以上的自定义数据结构可以由 C++ 语言、或脚本语言或 VB 语言等来实现。

由以上实施例可以看出, 由于自定义数据结构定义了一种通用数据类型; 当将数据存储为该通用数据类型时, 该通用数据类型可以先判断

待存储数据的类型，并将待存储数据的类型和值进行存储。因而，用户在存储数据时，不用手动进行不同数据类型的转换。

而且，由于在一个自定义数据结构的通用数据类型中，每个节点都可以设有其子节点的<key, WData>映射列表，因此，可以将大量的数据
5 存储于该通用数据类型中。这种情况下，只要将大量的数据存储成一个通用数据类型，则可以在传递一个通用数据类型的变量的情况下传递大量的数据，从而可以大大简化通讯系统的构建和维护。

图 2 为本发明实施例中存储数据的存储模型，即示出了自定义数据结构中每个节点的存储模型。如图 2 所示，每个存储模型至少存储一个
10 WData 数据，其包括第一模块（Value 模块），用于存储 WData 数据的类型标识和值，以及第二模块（<key, WData>模块），用于存储该节点的子节点的标识与子节点之间的映射列表<key, WData>。在第二模块中，key 可以使用字符串表示。

通过以上的存储模型，通用数据类型 WData 可以构造成一个树状的
15 通用数据类型。

根据本发明实施例，上述的第一模块可以利用 VARIANT 结构来实现数据的类型标识和值的存储。具体地，VARIANT 结构的内部利用一个类型标识来表示具体的数据类型。当需要使用存储的数据时，可以根据
20 VARIANT 结构中的类型标识来判断数据的类型，然后获取存储的数据。

VARIANT 结构的定义如下：

```
struct tagVARIANT  
{  
    union  
25 {
```

```

struct __tagVARIANT
{
    VARTYPE vt; // 类型标识
    WORD wReserved1;
5    WORD wReserved2;
    WORD wReserved3;
    // 以下为具体的类型，每次只有一个有效
    // 具体根据类型标识来判断哪个类型有效
    union
10    {
        LONG lVal;
        BYTE bVal;
        SHORT iVal;
        FLOAT fltVal;
15    .....
        }
    .....
}
.....

```

20 在定义 VARIANT 结构后，则可以按照上述结构存储数据。在存储数据后，则可以使用存储的数据。

图 3 示出了本发明实施例中存储数据的通用数据类型的树状结构。如图 3 所示，第一节点（根节点）可以有一个或多个子节点，而每个子节点也可以有一个或多个子节点，依此类从。这种情况下，一个通用数
25 据类型的变量可以存储大量的数据。并且，由于<key, WData>这种映射关系的使用，可以很方便地读取各个存储的数据，并可以方便地改变存储的某个或某些数据的数据类型或数据值。图 3 中，第一节点的第一模

块存储自身的一个值，以及该值的类型标识，第二模块中存储的<key, WData>列表包括 5 个映射关系，即(key1, WData), (key2, WData), (key3, WData), (key4, WData) 和 (key5, WData)。其中，(key3, WData)对应的节点的
 5 第二模块中又存储有包括 2 个映射关系的<key, WData>列表，即 (key6, WData) 和 (key7, WData); 而(key4, WData)对应的节点的
 第二模块中又存储有包括一个映射关系的<key, WData>列表，即(key8, WData)。

由此可见，本发明实施例提出的存储模型是一种递归的存储方式，可以构成树状的通用数据类型，存储大量的数据。

图 3 所示的树状结构只是本发明的通用数据类型的一种结构，并不
 10 用于限制本发明通用数据类型的结构。本发明的通用数据类型还可以由嵌套结构等具有多级索引特性的其他结构来实现。

在通过图 2 和图 3 所示的通用数据类型定义了一个变量之后，则可以方便地利用该变量对数据进行操作，比如对数据进行读取、转换以及传递。

15 图 1 所示的存储数据的方法可以实现不同类型的数据在同一变量中的存储，并且可以实现大量的数据的多级存储，比如图 3 所示的树状存储。在以下实施例中，以 C++语言的系统为例并结合图 3 所示的树状结构，说明本发明实施例中的数据存储方法。

在 C++语言中，定义一个通用数据类型以及将数据存储存储在通用数据类型
 20 可以如下所示：

```
WData data;  data = 5;  data="xxx";
data["key1"] = 10;  data["key1"] ="sss";
data["key3"]["key6"] = 15;
data["key3"]["key6"][...][...]= ...
```

25 从以上可以看出，首先定义了一个通用数据类型的变量 data。其次，

则可以对变量 data 进行赋值,即可以在 data 中存储数据。在本实施例中,给 data 赋值整型数据“5”;再次,还可以给 data 赋值字符串数据“xxx”,即将字符串数据“xxx”存储到 WData 通用数据类型的变量 data 中,覆盖了之前存储的整型数据“5”。具体地,可以通过重载“=”操作符的方式实现将数据“5”或“xxx”存储在变量 data 的根节点的第一模块中(即 Value 5 模块中)。

除了将数据存储存储在根节点外,还可以进行多级存储,即可以在根节点的子节点存储数据。例如,在 key1 对应的子节点中存储整型数据 10 或字符串数据“sss”。再例如,在 key3 对应的子节点中的 key6 对应的子节点中存储整型数据 15。通过以上的存储方法,可以将大量的不同类型的10 数据存储在一个通用数据类型的变量中,从而可以方便数据的使用。在多级存储中,可以重载“[]”操作符来实现 key 与各个 WData 的映射。

在采用图 1 所示的方法存储数据之后,则可以方便地使用存储的数据,比如读取、转换以及传递等。以下将结合存储数据的装置来说明如何15 使用存储的数据。

图 4 为本发明实施例的一种存储数据的装置。如图 4 所示,该装置包括:

数据接口 401,用于接收数据;

通用数据类型组件 402,用于建立并存储至少包括作为根节点的第一节点的自定义数据结构,其中,每个节点包括第一模块,用于存储数据的类型标识和值,以及第二模块用于存储所述节点的子节点的标识与子节点之间的映射列表;每当数据接口 401 接收一个数据,用于判断数据将要存储的位置以及数据的类型;如果数据存储于第一节点,则将数据的类型以及数据的值存储在第一节的第一模块中;如果数据存储于自定义数据结构中的第二节点,将数据的类型以及数据的值存储在第二25

节点的第一模块中。

根据本发明实施例，通用数据类型组件 402 包括：节点建立模块，判断模块和存储模块；

节点建立模块，用于建立并保存至少包括作为根节点的第一节点的
5 自定义数据结构；

判断模块，用于判断数据存储于第一节点或第二节点以及数据的类型，并将判断结果通知给存储模块；

存储模块，用于根据判断模块的判断结果，将数据的类型以及数据的值存储在所述第一节点的第一模块中或将数据的类型以及数据的值
10 存储在第二节点的第一模块中。

根据本发明实施例，判断模块还用于在数据存储于所述自定义数据结构的第二节点时，判断自定义数据结构中是否包括第二节点；如果没有，则通知节点建立模块建立自定义数据结构的第二节点；否则，通知存储模块将数据的类型以及数据的值存储在第二节点的第一模块中。

15 根据本发明实施例，节点建立模块通过在第二节点的父节点的第二模块中填写第二节点的标识与第二节点之间的映射列表来建立第二节点。

根据本发明实施例，数据接口 401 还用于从通用数据类型组件 402 读取第一节点的第一模块中的数据，以及根据节点标识从通用数据类型
20 组件 402 读取与节点标识对应的节点中存储的数据。

根据本发明实施例，判断模块还用于判断某个节点的数据的类型是否与新接收的待存储于所述节点的数据的类型不同，如果不同，则修改节点的第一模块中的数据的类型标识以及值；和/或

判断模块还用于判断是否接收到对自定义数据结构中的某个节点的
25 数据的类型转换操作，如果是，则修改节点的第一模块中的数据的类型

标识。

根据本发明实施例，通用数据类型组件 402 还包括传递模块，用于将自定义数据结构在不同或相同程序语言环境的模块间直接传递。

5 根据本发明实施例，通用数据类型组件可以由 C++语言、或脚本语言、VB 语言来实现。

根据本发明实施例，当通用数据类型组件由 C++语言实现时，该装置还包括 C++封装类 (C++ Wrapper)，用于对通用数据类型组件 402 中的自定义数据结构进行封装。

图 5 为本发明另一实施例的一种存储数据的装置。如图 5 所示，以
10 利用 C++语言来实现本发明的自定义数据结构为例，该装置包括：数据接口 IWDData 501，用于接收或发送数据；WData 通用数据类型组件 502，用于根据图 1 所示的方法存储接收的数据；C++Wrapper 封装类 503，用于对自定义数据结构的通用数据类型 WData 进行封装，从而可以使用“=”和/或“[]”操作符来实现数据的存储和/或多级存储。而 WData 通用数据类型组件 502 可以是 C++中的 Com 组件。在本发明实施例中，通过
15 数据接口 IWDData 501 还可以读取 WData 中根节点的数据值，并可以根据 key 读取 WData 中子节点的数据值。

在图 5 所示的实施例中，C++Wrapper 封装类 503 将 WData 通用数据类型组件 502 封装成 CWData 组件。具体地，C++Wrapper 封装类 503
20 利用智能指针 CComPtr<IWDData>对数据接口 IWDData 501 进行封装，从而使得在使用 CWData 组件时与直接通过数据接口 IWDData 501 来使用 WData 通用数据类型组件 502 没有区别。同时，CWData 通过重载“=”等号操作符，即重新定义“=”的含义，使得“=”具有判断 WData 中的数据以及数据类型的功能，并通过重载类型转换操作符（比如 (int)、
25 (BYTE) 等），即对类型转换操作符重新定义，使得类型转换符具有判

断 WData 中的数据以及将 WData 中的数据进行类型转换的功能，可以实现在 C++ 中在一个通用数据类型的变量中存储不同类型的数据，以及使用任意类型的数据；通过重载“[]”方括号操作符，实现了获取子节点中的 WData 数据的操作，比如 data[“key1”]就等价于获取 data 中 key1 5 对应的子节点中的 WData 数据。由于返回的子节点中的 WData 数据仍然是 CWData，所以直接多级使用方括号操作符，类似多维数组的使用。

在本发明又一实施例中，以图 5 所示的装置来说明如何使用存储的数据。具体地，说明对存储的数据的读取、转换以及传递。

1) 数据接口 IWData 501 的定义以及数据的读取可以如下：

10 数据接口定义：

```

interface IWData : IDispatch
{
    // 读取 WData 自身 Value
    HRESULT GetValue(VARIANT* pVar);
    15           HRESULT SetValue(VARIANT var);
    //根据 key 读取子 WData 的操作,读取的子 WData 又可以
    // 有自身的 Value 和子 WData.
    HRESULT GetSubData(BSTR bsKey, IWData** ppData);
    HRESULT SetSubData(BSTR bsKey,IWData* pData);
    20
    .....
}

```

在以上实例中，每个节点中的 Value 模块可以使用 VARIANT 结构来实现。

25 2) 将 WData 用 C++Wrapper 封装类 503 封装之后得到 CWData，利用 CWData 存储数据，并对存储的数据进行类型转换：

```
CWData data;  
data = 10 ;  
data = "string" ;  
data[1] = 10 ;  
5 data["key"] = 20;  
data["key1"]["key2"]["key3"] = "test" ;  
int x = (int)data;  
data = (BYTE)3;
```

由以上的使用实例可以看出，在进行数据类型转换时，可以由用户
10 指定具体的类型。由于 CWData 通用数据类型中具有数据类型判断功能，
可以将 data 中数据的基本类型直接转换为指定的类型。又或者，可以不
需要用户指定具体的类型，而且由 CWData 通用数据类型根据存储的数
据的默认类型对 data 的基本类型进行转换。比如，当 data 中的数据是数
15 字时，则转换为数字的默认类型，当 data 中的数据是字符串时，则转换
为字符串的默认类型。

比如，“data = 10”中，利用重载的“=”操作符把 10 存储在 data 的根
节点的 Value 模块，而之后“data = “string””的操作中，利用重载的“=”操
作符把 string 存储在 data 的根节点的 Value 模块中，而此操作同时也是
实现了将 data 根节点的数据类型由整型转换为字符串型，此时类型标识
20 由整型改为字符串型。

当然，也可以不利用 C++Wrapper 封装类 503 封装 WData，而直接
使用 WData 存储数据，并对存储的数据进行使用。但是，这种情况下，
操作会比较复杂，比如：

```
//直接使用 WData 存储和使用数据  
25 CComPtr<IWData> pData;  
CreateWData(&pData);
```

```
pData->SetValue(CComVariant(10));
pData->SetValue(CComVariant("string"));
```

```
5      CComPtr<IWData> pSubData;
      pData->GetSubValue("key",&pSubData);
```

```
pSubData->SetValue(CcomVariant(20));
```

10 以上直接使用 WData 存储和使用数据的结果相当于使用 CWData 的如下操作的结果:

```
      CWData data;
      data = 10 ;
      data = "string" ;
15     data[1] = 10 ;
      data["key"] = 20;
```

由此可见, 利用 C++Wrapper 封装类 503 封装 WData 组件后, 对 WData 的数据的存储和使用更加简洁、方便, 从而可以使得通讯系统的运行速度更快, 且通讯系统的维护也会更加方便, 可以节省通讯系统的维护成本。

3) 利用 C++Wrapper 封装类 503 封装通用数据类型 WData, 在通讯系统的各模块间传递被存储为通用数据类型 WData 的数据, 例如, 可以将本发明的技术方案利用在微博系统中以存储微博的用户资料, 并在微博系统的各模块间传递存储的微博用户资料。

25 例如, 微博系统中具有模块 A 如下:

```
CWData msg;
```

```
msg["id"] = ... ;  
msg["content"] = "xxxxx" ;  
msg["time"] = ... ;  
.....
```

5

而模块 A 可以将 msg 作为参数调用模块 B，例如：

```
ShowMsg(CWData msg);
```

模块 B 如下：

```
Void ShowMsg (CWData msg)
```

10

```
{  
    // 显示内容  
    ... msg["content"] ...  
    // 显示时间  
    ... msg["time"]...  
    .....
```

15

```
}
```

从以上模块 A 和 B 之间的调用实例可以看出，模块 A 可以通过一个 msg 变量将 “id”，“content” 和 “time” 三个数据传递给模块 B，从而可以实现模块 A 和 B 之间的大量数据的传递。因此，可以简化微博系统的建立和维护过程，节省微博系统的开发和维护成本。

20

以上所述仅为本发明的较佳实施例而已，并不用以限制本发明，凡在本发明的精神和原则之内所做的任何修改、等同替换和改进等，均应包含在本发明的保护范围之内。

权利要求书

1、一种存储数据的方法，其特征在于，包括以下步骤：

建立至少包括作为根节点的第一节点的自定义数据结构，其中，
所述自定义数据结构中的每个节点包括第一模块，用于存储数据的类
5 型标识和值，以及第二模块用于存储所述节点的子节点的标识与子节
点之间的映射列表；

每接收一个数据，判断所述数据将要存储的位置以及所述接收的
数据的类型；

如果所述数据存储于所述第一节点，将数据的类型以及数据的值
10 存储在所述第一节点的第一模块中；

如果所述数据存储于所述自定义数据结构中的第二节点，将数据
的类型以及数据的值存储在所述第二节点的第一模块中。

2、如权利要求 1 所述的方法，其特征在于，还包括：如果所述
数据要存储于所述自定义数据结构中的第二节点，判断所述自定义数
15 据结构中是否包括所述第二节点；

如果没有，则建立所述自定义数据结构中的第二节点；

否则，直接将数据的类型以及数据的值存储在所述第二节点的第
一模块中。

3、如权利要求 2 所述的方法，其特征在于，所述建立所述自定
20 义数据结构的第二节点包括：

在所述第二节点的父节点的第二模块中填写所述第二节点的标
识与所述第二节点之间的映射列表。

4、如权利要求 1、2 或 3 所述的方法，其特征在于，如果一个节
点没有子节点，则其第二模块中的映射列表为空。

5、如权利要求 4 所述的方法，其特征在于，还包括：通过重载类型转换操作符以来实现对存储在所述自定义数据结构中的数据
进行类型的转换；

其中，所述判断接收的数据的类型包括：通过重载程序语言中的
5 等号操作符来实现对数据的类型的判断。

6、如权利要求 4 所述的方法，其特征在于，还包括读取存储在
所述自定义数据结构中的数据。

7、如权利要求 4 所述的方法，其特征在于，还包括通过传递所
述自定义数据结构来实现将存储于所述自定义数据结构中的一个或
10 多个或所有数据在不同或相同程序语言环境的模块之间传递。

8、如权利要求 4 所述的方法，其特征在于，所述自定义数据结
构由 C++语言、或脚本语言或 VB 语言实现。

9、一种存储数据的装置，其特征在于，包括：

数据接口，用于接收数据；

15 通用数据类型组件，用于建立并保存至少包括作为根节点的第一
节点的自定义数据结构，其中，所述自定义数据结构中的每个节点包
括第一模块，用于存储数据的类型标识和值，以及第二模块用于存储
所述节点的子节点的标识与子节点之间的映射列表；每当数据接口接
收一个数据，判断所述数据将要存储的位置以及数据的类型；如果所
20 述数据存储于第一节点，则将数据的类型以及数据的值存储在所述第
一节点的第一模块中；如果所述数据存储于所述自定义数据结构的第
二节点中，则将数据的类型以及数据的值存储在所述第二节点的第一
模块中。

10、如权利要求 9 所述的装置，其特征在于，所述通用数据类型
25 组件包括：节点建立模块、判断模块和存储模块；

所述节点建立模块，用于建立并保存至少包括作为根节点的第一节点的自定义数据结构；

所述判断模块，用于判断所述数据将存储于第一节点或第二节点以及所述数据的类型，并将判断结果通知给所述存储模块；

5 所述存储模块，用于根据判断模块的判断结果，将数据的类型以及数据的值存储在所述第一节点的第一模块中或将数据的类型以及数据的值存储在所述第二节点的第一模块中。

11、如权利要求 10 所述的装置，其特征在于，所述判断模块还用于当所述数据将存储于所述自定义数据结构的第二节点时，判断所述自定义数据结构中是否包括所述第二节点；如果没有，则通知所述节点建立模块建立所述自定义数据结构的第二节点；否则，通知所述存储模块将数据的类型以及数据的值存储在所述第二节点的第一模块中。

12、如权利要求 11 所述的装置，其特征在于，所述节点建立模块通过在所述第二节点的父节点的第二模块中填写所述第二节点的标识与所述第二节点之间的映射列表来建立所述第二节点。

13、如权利要求 10、11 或 12 所述的装置，其特征在于，所述通用数据类型组件由 C++ 程序语言实现；所述装置还包括：C++ 封装类（C++ Wrapper），用于对通用数据类型组件中的自定义数据结构进行封装。

14、如权利要求 12 所述的装置，其特征在于，所述数据接口还用于从所述通用数据类型组件读取第一节点的第一模块中的数据，以及根据节点标识从所述通用数据类型组件读取与所述节点标识对应的节点中存储的数据。

25 15、如权利要求 12 所述的装置，其特征在于，所述判断模块还

用于判断某个节点的数据的类型是否与新接收的待存储于所述节点的数据的类型不同，如果不同，则修改所述节点的第一模块中的数据的类型标识以及值；和/或

5 所述判断模块还用于判断是否接收到对所述自定义数据结构中的某个节点的数据的类型转换操作，如果是，则修改所述节点的第一模块中的数据的类型标识。

16、如权利要求 12 所述的装置，其特征在于，所述通用数据类型组件还包括传递模块，用于通过传递所述自定义数据结构来实现将存储自定义数据结构中一个或多个或所有数据在不同或相同程序语言环境的模块间的传递。

17、如权利要求 9、10 或 11 所述的装置，其特征在于，所述数据接口是 COM 接口。

18、一种通用数据类型组件，其特征在于，包括：节点建立模块，判断模块和存储模块；

15 所述节点建立模块，用于建立并保存至少包括作为根节点的第一节点的自定义数据结构，其中，所述自定义数据结构中的每个节点包括第一模块，用于存储数据的类型标识和值，以及第二模块用于存储所述节点的子节点的标识与子节点之间的映射列表；

20 所述判断模块，用于每接收一个数据后，判断所述数据将存储于第一节点或所述自定义数据结构的第二节点，判断数据的类型，并将判断结果通知给所述存储模块；

25 所述存储模块，用于当所述判断模块判断所述数据将存储于第一节点时，将数据的类型以及数据的值存储在所述第一节点的第一模块中；在所述判断模块判断所述数据将存储于所述自定义数据结构的第二节点时，将数据的类型以及数据的值存储在所述第二节点的第一模

块中。

1/2

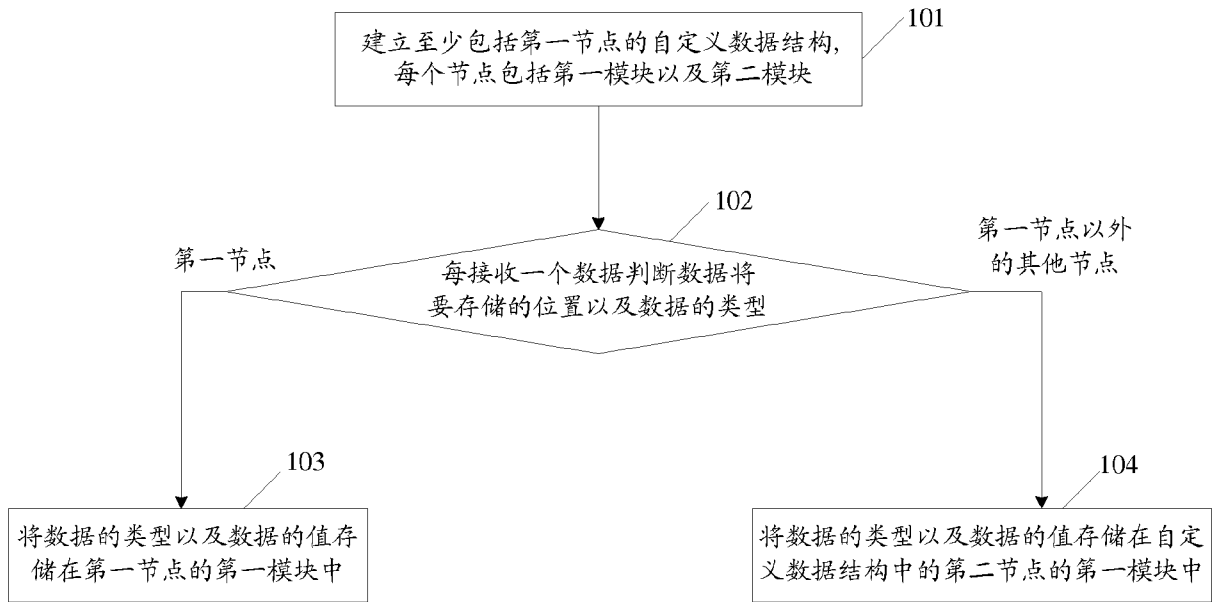


图 1

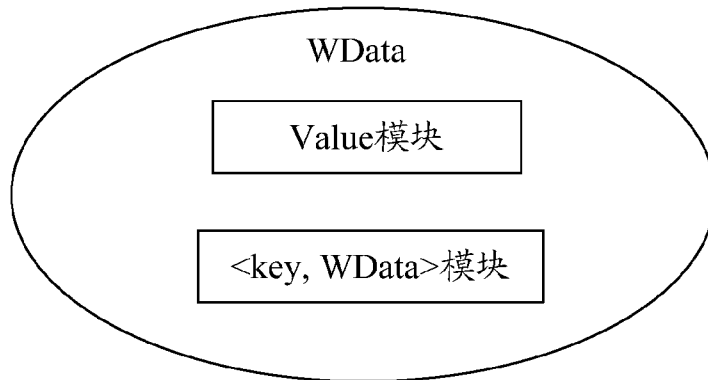


图 2

2/2

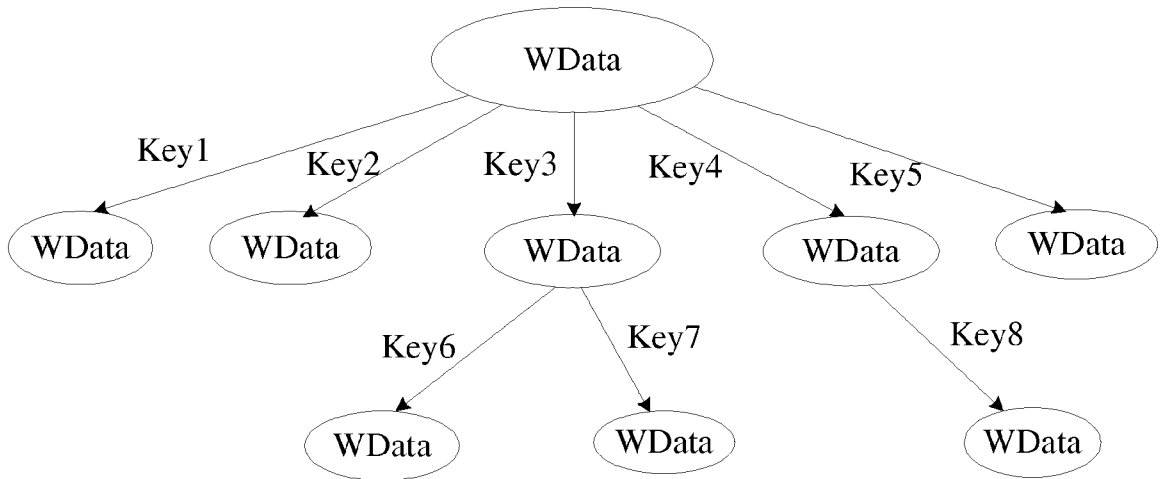


图 3

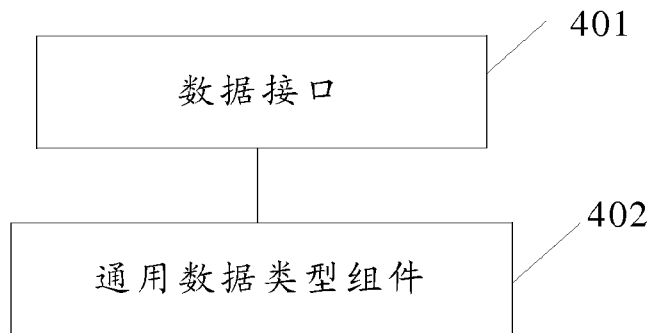


图 4

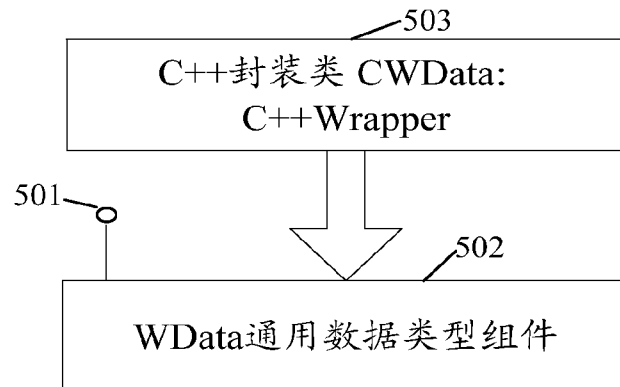


图 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2011/079035

A. CLASSIFICATION OF SUBJECT MATTER

G06F17/00 (2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC:G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CPRSABS,DWPL,SIPOABS,CNKI(data, store, type, value, convert, define, construct, data structure, node)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2005262471 A1(BEA Sytems, Inc)24 Nov. 2005(24.11.2005)the whole document	1-18
A	US 7627604 B2 (Turbo Data Laboratories, Inc)01 Dec. 2009(01.12.2009)the whole document	1-18

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&”document member of the same patent family

Date of the actual completion of the international search

20 Oct. 2011 (20.10.2011)

Date of mailing of the international search report

03 Nov. 2011(03.11.2011)

Name and mailing address of the ISA/CN
The State Intellectual Property Office, the P.R.China
6 Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China
100088
Facsimile No. (86-10)62019451

Authorized officer

WANG, Yankun
Telephone No. (86-10)62411680

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2011/079035

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
US2005262471A1	24.11.2005	US7765520B2	27.07.2010
US7627604B2	01.12.2009	JP4537391B2	01.09.2010
		US2008270435A1	30.10.2008
		WO2005088479A1	22.09.2005

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2011/079035

检索报告中引用的 专利文件	公布日期	同族专利	公布日期
US2005262471A1	24.11.2005	US7765520B2	27.07.2010
US7627604B2	01.12.2009	JP4537391B2	01.09.2010
		US2008270435A1	30.10.2008
		WO2005088479A1	22.09.2005