

[19]中华人民共和国国家知识产权局

[51]Int. Cl<sup>7</sup>

H04L 29/06

## [12] 发明专利申请公开说明书

[21] 申请号 98810147.5

[43]公开日 2000年12月6日

[11]公开号 CN 1276123A

[22]申请日 1998.10.13 [21]申请号 98810147.5

[30]优先权

[32]1997.10.13EP [33]EP [31]97203174.4

[86]国际申请 PCT/NL98/00586 1998.10.13

[87]国际公布 WO99/20025 英 1999.4.22

[85]进入国家阶段日期 2000.4.13

[71]申请人 艾克斯-伟瑞茨公司

地址 荷兰弗尔德霍芬

[72]发明人 汉德·范德海登 罗伯特·德布勒

奇普·布兰克

[74]专利代理机构 永新专利商标代理有限公司

代理人 韩 宏

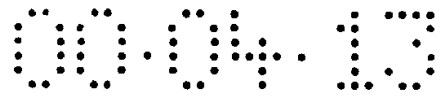
权利要求书 6 页 说明书 40 页 附图页数 3 页

[54]发明名称 面向对象的点到点通信的方法及实施该方法  
的通信设备

[57]摘要

通过具有灵活消息格式(ILMF)的消息在发送者(SRV(m))与接收者(SRV(m))之间进行点到点通信的方法和设备,所述的消息包括:一个头标,至少具有消息定义参照(MSG ID、MSG CLASS、MSG VERSION、MSG CREATOR),一个发送者标识符(SENDER ID)和一个目标地址(DESTINATION ADDRESS);消息内容涉及:字段数量(FIELD COUNT)和任意字段的内容(FIELD(1)···);对象数量(OBJECT COUNT)和任意对象的内容(OBJECT(1)···);字段映射的数量(MAP COUNT)和任意字段映射的内容(MAP(1)···),任何字段映射都可由预先确定的字段使用;动作的数量(ACTION COUNT)和任意动作的内容(ACTION(1)···),所有动作都至少可由预先确定的字段使用。

ISSN 1000-4274



## 权 利 要 求 书

---

1、 通过具有灵活消息格式 (ILMF) 的消息在一个发送者 (SRV (m)) 和一个接收者 (SRV (m)) 之间进行点到点通信的方法, 其特征在于所述消息中的任何一个都包括:

一个头标, 至少包括消息定义参照 (MSG ID、MSG CLASS、MSG VERSION、MSG CREATOR), 一个发送者标识符 (SENDER ID) 和一个目标地址 (DESTINATION ADDRESS);

\* 字段数量 (FIELD COUNT) 和任意字段的内容 (FIELD (1) ...);

\* 对象数量 (OBJECT COUNT) 和任意对象的内容 (OBJECT (1) ...);

\* 字段映射的数量 (MAP COUNT) 和任意字段映射的内容 (MAP (1) ...), 任何字段映射都可由预先确定的字段使用;

\* 动作的数量 (ACTION COUNT) 和任意动作的内容 (ACTION (1) ...), 所有动作都至少可由预先确定的字段使用。

2、 按照权利要求 1 的方法, 其中所述的消息定义参照包括一个用于标识所有消息的消息标识 (MSG ID)。

3、 按照权利要求 1 或 2 的方法, 其中所述的消息定义参照包括一个消息类别标识 (MSG CLASS), 用于标识消息的类别, 如邮件、商业消息、定单或发货单。

4、 按照前述任一权利要求的方法, 其中所述的消息定义参照包括一个消息版本标识符 (MSG VERSION), 用于标识任一消息的版

本号。

5、 按照前述任一权利要求的方法，其中所述的消息定义参照包括一个消息创建者标识符（MSG CREATOR），用于标识任一个消息的创建者。

6、 按照前述任一权利要求的方法，其中所述的头标包括一个对所实施的加密类型（ENCRYPTION TYPE）的参照。

7、 按照前述任一权利要求的方法，其中所述的头标包括一个对所实施的压缩类型（COMPRESSION TYPE）的参照。

8、 按照前述任一权利要求的方法，其中所述的头标包括一个对一个应用（APPLICATION NAME）的参照，用于标明任何一个消息是否是一起形成所述的应用的一系列消息中的成员。

9、 按照前述任一权利要求的方法，其中所述的任何一个消息都包括一个数字签名。

10、 一种包括处理装置（ILMS）和数据库（ILMDB）的通信设备，用于通过具有灵活消息格式（ILMF）的消息与另一个通信设备（SRV（m））进行点到点通信，所述的消息包括：

一个头标，至少包括消息定义参照（MSG ID、MSG CLASS、MSG



VERSION、MSG CREATOR), 一个发送者标识符 (SENDER ID) 和一个目标地址 (DESTINATION ADDRESS);

- \* 字段数量 (FIELD COUNT) 和任意字段的内容 (FIELD (1) ...);
- \* 对象数量 (OBJECT COUNT) 和任意对象的内容 (OBJECT (1) ...);
- \* 字段映射的数量 (MAP COUNT) 和任意字段映射的内容 (MAP (1) ...), 任何字段映射都可由预先确定的字段使用;
- \* 动作的数量 (ACTION COUNT) 和任意动作的内容 (ACTION (1) ...), 所有动作都至少可由预先确定的字段使用。

所述的处理装置 (ILMS) 被配置成: 使用所述的消息定义参照作为对所述预定的消息定义表的参照查询存储在所述数据库 (ILMDB) 中的一个预定的消息定义表 (msgdef)。

11、按照权利要求 10 的通信设备, 其中所述的预定的消息定义表 (msgdef) 包括一个用于标识所有消息的消息标识符 (msgid)。

12、按照权利要求 10 或 11 的通信设备, 其中所述的预定的消息定义表 (msgdef) 包括一个用于标识所有消息的消息类别如邮件、商业消息、定单或发货等的消息类别标识符 (msgclass)。

13、按照权利要求 10 到 12 中任一个的通信设备, 其中所述的预定的消息定义表 (msgdef) 包括一个用于标识所有消息的版本号的消息版本号标识符 (msgver)。

14、按照权利要求 10 到 13 中任一个的通信设备，其中所述的预定的消息定义表（msgdef）包括一个用于标识所有消息的创建者的消息创建者标识符（creatid）。

15、按照权利要求 10 到 14 中任一个的通信设备，其中所述的预定的消息定义表（msgdef）包括一个对所施加的加密类型（encrtype）的参照。

16、按照权利要求 10 到 15 中任一个的通信设备，其中所述的预定的消息定义表（msgdef）包括一个对所施加的数字签名类型（sigtype）的参照。

17、按照权利要求 10 到 16 中任一个的通信设备，其中所述的预定的消息定义表（msgdef）包括一个消息系统标识符（msyid），用作为对所述数据库（ILMDB）中进一步表的参照。

18、按照权利要求 17 的通信系统，其中所述的进一步表还包括一个字段定义表（flddef），用于保存对所述消息的任一字段的主要定义。

19、按照权利要求 17 或 18 的通信系统，其中所述的进一步表包括一个字段映射表（fldmap），该映射表包含由预定字段使用的映射信息，例如对超文本标记语言字段、数据库字段、平文件字段和

其它消息字段的映射，所述的数据库字段和平文件字段存储在客户数据库（CDB）中。

20、按照权利要求 17 到 19 中任一个的通信系统，其中所述的进一步表包括一个字段动作表（fldact），该动作表包含由预定的字段使用的动作信息。

21、按照权利要求 17 到 20 中任一个的通信系统，其中所述的进一步表包括一个消息预先处理表（msgpre）和一个消息后处理表（msgpost），该预先处理表包含对所接收或要发送的消息预处理而要执行的动作清单，而该消息后处理表包含对所接收的消息进行后处理而要执行的动作清单。

22、按照权利要求 20 或 21 中任一个的通信系统，其中所述的字段动作表（fldact）、所述的消息预先处理表（msgpre）和所述的消息后处理表（msgpost）包含对从下列动作组中选择的动作类型的参照：动作的数据库类型和包括数学计算、赋值、逻辑操作、条件操作的动作逻辑类型以及命令。

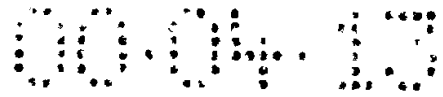
23、按照权利要求 10 到 22 中任一个的通信系统，其中所述的消息定义表（msgdef）包含一个用于标识所接收的消息是否是一个应用的第一消息的应用字段（appmain）和一个用于参照所述的应用名称的应用名称字段（appname）。

24、按照权利要求 23 的通信设备，其中所述的应用是一个分布在多个通信设备上的分布式应用。

25、按照权利要求 10 到 24 中任一个的通信设备，其中所述的设备被配置成：如果所接收的消息参照的消息定义没有存在于它的数据库（ILMDB）中，则从一个发送者那里请求一个新的消息定义，并从所述的发送者那里接收所述的新的消息定义并将其存储在所述数据库（ILMDB）内的所述的消息定义表（msgdef）中。

26、按照权利要求 10 到 25 中任一个的通信设备，其中所述的处理装置（ILMS）被配置成或者将所接收的消息与一个指定的 HTML 文件进行溶合或者，如果该处理装置（ILMS）没有找到所指定的 HTML 文件，就生成一个缺省的动态 HTML 文件。

27、包含按照权利要求 10 到 25 中任一个的通信设备（SRV(m））和连接到所述通信设备的终端（ILMC）的系统，其中所述的终端包含一个终端处理器（1），一个显示单元（6）和用于用户输入数据用的输入装置（12，13），所述的通信设备被配置成：如果所述的终端在该消息中被指明为目标地址，则把所接收的消息传递给所述的终端，所述的终端处理器（1）被配置成把一个消息与一个指定的 HTML 文件溶合或者，如果该终端处理器（1）没有找到所指定的 HTML 文件，就生成一个缺省的动态 HTML 文件。



# 说明书

---

## 面向对象的点到点通信的方法及实施该方法的通信设备

### 发明领域

本发明属于电子数据通信、基于数据的有组织的灵活格式的电子数据消息管理、客户服务器及分布式软件应用领域。

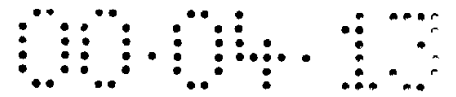
尤其是，本发明使用关系数据库来描述和管理对灵活格式的电子数据消息的生成、通信、说明、显示和实时的响应以及作为灵活格式的电子数据消息的一个集合而组织的应用，该应用可以是有选择地分布在一个或多个电子数据网络上的一个或多个计算机系统上。

### 发明背景

一般用途和商业特殊用途的电子数据通信要满足四个问题，其中每一个问题包括很多的子问题，我们将讨论关于商业通信需要的议题，尽管下面的论点适用于大多数电子数据通信的范畴。这四个问题是：

1) 要发送和接收的电子数据必须清晰地识别以便确保其含义，每个公司都有不同的方式来表达即使是最基本的商业信息、定单表格或或发货帐单等；

2) 从另一个公司接收到一个电子数据并且正确地加以识别，它就必须集成进接收公司的数据库和业务处理程序。在此，又一次，每一个公司在它们的数据库的结构上有很大的不同，这意味着它们



在其计算机上表示和存储他们的电子数据的方式的不同；

3) 如果该电子数据必须要显示给一个终端用户并且允许终端用户的交互，在此再一次，每个公司可以使用不同类型的计算机显示终端、各种类型的显示软件、各种系统要求及商业要求。通常有必要建立非常昂贵的具有定制的图形用户显示的电子数据显示程序。

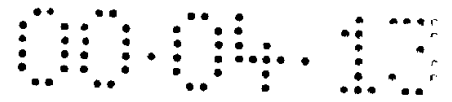
4) 由于需要使用特定系统服务的集合及其相关的数据以完成特定的商业用途，存在着这样一种需求：以一种协调的方式执行分布在多个计算机系统计算机化的商业功能的处理。这意味着在多于一个的位置上需要有计算机化的商业系统，在一个或多个位置上分别地运行着，但在功能上本质上是一个单一的系统。解决该问题的方案通常是作为分布式应用而分布，或者，就象最近出现的，作为可按“中间块”划分的分布式对象管理系统。

与问题 4 相关的有两个非常明显的主要议题，首先是需要或期望一个公司在多个不同的系统上使用同一个商业应用，以避免为每一个不同的系统重写需要在其上运行的商业逻辑、用户接口和网络通信接口；第二个议题是需要将一个单一应用的功能在一个通信网络的多个系统上进行划分，并且使其以一种整体协调的方式运行。

目前没有一个单一系统能解决所有这四个问题。每年要消耗数百万美元来购买、建立、重建以及维持电子数据通信和存储系统，来不够有效、不完善地解决这四个问题的各种组合。

## 先有技术

### 问题 1



为了解决第一个问题，在进行实际的通信之前，必须由所有通信认可一个针对特定信息类型的一套电子数据格式。这种对格式的认可以必须通过使用通信软件来进行实效化，而通信软件也必须使用同一认可的格式。EDI（电子数据交换）是解决这种问题的一个途径。EDI 对于很多类型的商业事务来说是一套电子数据消息格式，这种格式不够灵活，也就是说它们不能够被用户改变，同时也有很多的EDI 消息格式的版本。一个使用 EDI 消息通信系统的公司可能还不能与使用另外一个版本的 EDI 的公司交换数据。将已有的系统转换成 EDI 系统所花费的金钱与时间是非常大的，在需要时是很难并且也很昂贵来改变或增加另一种格式。

使用硬编码（即，在软件中定义）消息格式的方法和方案具有与 EDI 相关的所有问题，其实质上是同一方案，只是具有不同的固定消息格式。每当固定的消息格式必须要改变时，就要求在软件中有类似的改变以及数据库或用户接口的改变。即使一个非常小的改变有时也可能要耗时数月。当然，在使用特定格式的每个商业计算机上也要改变这种电子数据格式以确保电子通信能正确地持续运行。

## 问题 2

第二个更难于解决，一般地，大多数的公司都使用大量的数据录入人员，他们读取印刷的材料并且输入他们已读进计算机程序的数据，该计算机程序运行在处理该数据的终端上，将其转换成正确的格式并放进公司的数据库。这些使用 EDI 或类似通用电子消息格式的公司通常要求已经为其定制的软件付出大量的费用，这种定制



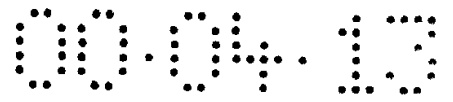
的软件自动地为他们处理并转换通用电子消息格式，并与已有的数据库结构进行交互来进行数据的存储和检索。如果它们需要改变或增加到该通用消息格式，或需要改变或扩展其数据库结构，那么他们就必须再一次花大量的时间和金钱来改变和重写其电子消息格式数据库接口程序。

一个更新的公共域通用方案以映射特定数据项到客户数据库或存储设备中的特定位置为中心而展开。这种映射不是在软件中生成，而是在与该软件应用和客户数据库分离的、更易于改变和灵活的表述中，如软件应用读取的独立的平（flat）文件。这就消除了在数据结构或数据表述改变时需要重新编程和重新建立软件应用的要求。数据段映射技术在很多数据库公司中广泛采用，如 Progress、Oracle 和 SQL Base（SQL：数据库查询语言）。这种已知方法的一个版本可构成溶合进本发明的通信设备的的功能和方法论的一部份。

### 问题 3

第三个问题具有很不容易使用的通用方案。一类方案是要求为各种视窗（图形用户接口）系统写入大量的软件代码，如 X-Windows, Microsoft Windows。对于大量的多功能的商业应用，该方案要求大量的时间、金钱和人力，并且使得对用户接口的改变和增加非常耗时耗力。

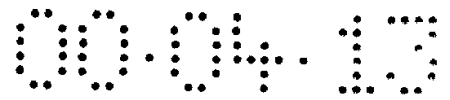
一种新的且更有希望的方案是：因为它们用在 WWW 和因特网上，以使用 HTML（超文本标记语言）和网页浏览器为中心，它能生成非常好看的图形用户接口，通常称为“网页”。这引起了众多公司的兴趣，他们能够看出利用 HTML 能快速而又简捷地建立非常好看的图形



用户界面，并且想知道是否能以同样的方式在与网站浏览器系统同样的网络系统上为其电子数据通信和处理程序生成该用户接口。事实上公司们是希望通过因特网发送和接收所有的商业和个人数据与消息。

目前，有几个系统利用 HTML 和网站浏览器可以允许通过因特网来进行有限的商业和一般数据通信，数据通信类型受限的原因是网站服务器/网站浏览器系统没有被设计成提供完全的数据通信服务，它们是被设计成为动态文件检索和显示系统。它们可检索由用户请求首先初始化的信息，它们并没有给用户向一个或多个设计者发送信息的能力，除非通过扩展编程。进行发送的能力并非系统所固有。这种类型的例子是 Progress 公司的 Websped，IBM 的 WebEDI 和 Sappjire 公司的 PageBlazer。

即使是采用“推出”技术的网站浏览器和网站服务器也仅允许接收数据，而不能发送。而这种接收实际上又是作为一个仿真广播接收器、由网站浏览器和网站服务器系统发出的对信息的重复隐藏请求（轮流检测）而建立的。扩展这项“推出”技术的计划仅包括使用真实的广播和多点传送技术。“推出”技术的示例有 Marimba 公司的 Castanet 和 Tibco 的 TIB API，这种“推出”技术对于意欲以一种新的方式达到很大的市场的广告商来说是很有吸引力的，但对于希望或需要使用 HTML 和因特网的高级文件显示特性的用户来说，在采用点到点协议（该协议可以很容易地集成进它们的已有数据库和商业系统）完全地集成进双路信息通信的环境下，则没有多大的用途。



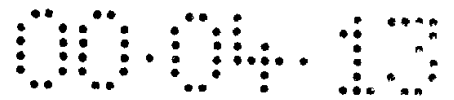
针对因特网的网站浏览器和网站服务器接口被建设成允许最终用户向另一个浏览器的用户发送一个消息或发送一个网页，通过电子邮件除外。一个浏览器的用户决不会收到一个特定的消息，如果该消息是没有被首先请求的，即便是通过轮流检测请求了所谓的“推出”技术信息，并且该信息的本质特征可以与一个电视广播或一个集群邮件相比较，即使它是定制的。该信息不会发送给一个特定的接收，例如，将其与电子邮件（email）相对比。作为电子邮件，一个人可以接收任意数量的特定的、个人消息而不用事先向发送者请求这些信息，也可以向特定的接收用户发送任意数量的特定的、个人消息而不用事先被请求，这是消息最普通最必要的本质，但是网站浏览器系统却没有设计成这样。

为什么人们乐于使用这种方式呢？为什么它们不仅使用 email 呢，答案在于 email 没有内建其中的数据格式化服务（而 HTML 有），也没有内建其中的任何用户接口生成系统（而网站浏览器有）。一个 email 消息仅占用中等数量的时间（几秒到几天）到达其终点，而有时可能会丢失，这对于标准的商业电子数据通信来说是不能接受的，这就是人们为什么喜欢网站服务器和网站浏览器系统，因为它有很好看的内建其中的数据显示能力以及中等速度的通信，它是如上面所提到的，但没有点到点通信的能力或数据表述或映射能力。

在本发明中，可以将公共域 HTML 标准集成进来，作为生成非常容易建立、快速而又非常灵活的图形用户接口的方案。

#### 问题 4

历史上问题 4 以一种通用的方式来对待，直到最近才由关系数



数据库管理系统（RDBMS）加以解决，该类方案受限于解决分布式查询和分布式数据库管理的正常功能，而不解决前面所说的问题 4 的两个主要议题。

另一个较新的方案可在当前称为“中间块”或分布式对象管理系统的系统中找到，其中一个较好的例子是称为 CORBA（共同对象请求中间结构）的系统，关于 CORBA 的发展历史、目标、当前功能及 CORBA 2.0 的不足的文章可在 Warren Kayffill 的“CORBA masterminds object management”（DBMS 杂志 卷 10 No. 3, P 42 1997. 3）以及在 DBMS 的同一刊的 T. J. Hart 的“Questioning CORBA. Bringing Corba-based designs to life faces a multitude of obstacles”（P52）一文中可以见到。

该类型的解决方案并不致力于数据库的互操作性，通用的数据格式出现在应用层或用户接口层，它单独地致力于用标准通信接口写应用程序和在通信层上写通信数据的能力。有必要为该类型的通信系统提供一个开发环境、一个运行环境和软件库集以便允许程序创建能在任意的系统上运行的应用程序，然后，在码（也称为对象）中定义了数据结构之后，这此数据结构也可以与被编码成识别和使用那些特定数据结构的其它软件应用进行交换。

这种功能和能力实际上与一个程序员写一个 Java 形式的消息通信系统没有什么不同，它几乎可以运行在所有系统上，且可以使用 EDI 消息或其它的双方认可的作为通信媒介的固定消息格式。CORBA 系统并不解释任何消息，而由程序员所写的应用程序必须能识别该消息，这使得 CORBA 应用程序作为一个 EDI 或固定消息格式应用系

统具有同样的问题，即开发、变更和维护上的费时费力。即使 CORBA 系统所可能具有的分布式功能不是 CORBA 所固有的，它也能存在于设计和编码该应用程序并能使用该 CORBA 通信库的程序员的技术中。实际上，CORBA 系统仅有的好处是在多个运行环境上的一个应用系统的部件间的互操作性，它能比其它装置如 Java 获得更大的好处。

实际上，在大多数的情况下，“中间块”好像是另一个电子数据通信环境，该通信环境可以在多数的计算机上提供标准的通信环境和某些标准的软件功能，但却不能解决软件应用建立的所有方面，而因此要求使用依赖于系统的数据库接口和用户接口代码，因此使应用如 CORBA 这样系统的目标受到挫折。

有一些系统将 CORBA 的功能溶进 Java 并且包括一个标准的称为 ODBC（或在 Java 系统称为 JDBC）的数据库接口，这是一个非常有意义的开发，但由于 ODBC 数据库接口仅对于只读查询是有效的，而不能成功地创建要求能完整地进行数据库访问的标准商业事务系统。而且，这种组合也不能完全地解决问题。而且，由于使用 ODBC 和 CORBA 对象中间（broker）系统的不实用的过高的管理费用，它也不是建立本发明所期望的平台。

另外从以下将要简要描述的专利文献中可以看到某些先有技术。美国专利 5257369（Tibco Inc）揭示了一种较早出现的“中间件”，而且该文献也提供了广播功能并且是基于广播功能。主要是提供一种方案来解决立即传送实时数据给多个接收者的问题，如实时股票报价或市场信息，但却并不致力于点到点通信或问题 2 或

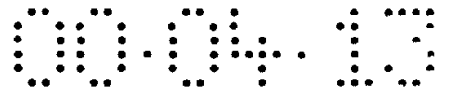
3。

该消息可以改变并为一个数据结构或对象进行定义，如前面在 CORBA 中所描述的。但这种改变或增加消息格式的能力却又必须得改变应用的代码，而且也不能将消息格式的数据映射到一个用户接口或数据库字段。

WO 96/20553 (Alphanet Telecom , Inc) 致力于增强电子邮件服务以包含语音邮件和传真信息，对该系统的描述是一个中心化的服务，用户可以通过因特网、连接到同一设备的电话或传真机与其连接以接收语音邮件、电子邮件或传真。这项发明并不是特意地针对问题 1、2、3 或 4，但是从通信的观点来看，却提供一种方式使得电子邮件更有效。

WO 96/34341 (Charles Bobo II) 致力于一个中心化的服务，其收集、存储和允许针对所连接用户的数据访问，这完全不同于解决问题 1 至 4 的点到点通信，以及对于大多数的标准商业应用。而且，有很多的商业可用系统已出现多年，并解决与其同样的问题，如 GEIS 和 IBM 专用数据网络。

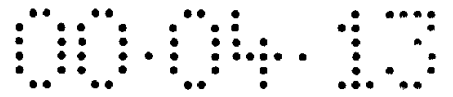
EP-0747840 A1, EP-0747841 A1, EP-0747842 A1, EP-0747843 A1, EP-0747844 A1, 以及 EP-0747845 A1, (都是 IBM 的) 致力于利用某种形式的数据访问增强当前网站服务器的功能，因此仍然受限于前面讨论的网站服务器的通信限制，因此 该发明根本不能解决问题 1 和 4 及只能部份地解决问题 2 和 3。而且已经有几个系统在网站服务器的增强方面提供与其类似的功能，如 Progress Database Corporation 的 WebSpeed。



WO 95/11560 (Sybase, Inc) 属于“中间件”的范畴，它只解决这样的问题，即不管所使用的硬件或软件如何，它可生成应用所使用的连续的电子数据通信软件接口。该文档没有以任何方式致力于正在通信的电子数据的格式，因此该方案与 CORBA 的极相似，并且从本发明所致力的问题的观点来看，它还具有着同样的限制。

EP 0 421 624 A2 (Texas Instruments Incorporate) 是诸如 CORBA 这样“中间件”系统和应用开发环境的集合。利用一个数据库控制和管理软件开发环境，该发明允许程序员开发 C、COBOL 和 SQL 的应用程序。该数据库包含允许大群的程序员对复杂的数据库事务应用开发、修改和使用源代码控制技术的信息。该数据库也包含允许该应用能在其它硬件系统上进行再编辑的信息，并且能在每一种类型的计算机上以同样的方式进行工作，尽管在该发明中仅讨论了 IBM 主机系统和某些 UNIX 系统，最终的结果是一系列硬编码的、编辑的应用程序，它解决了问题 1 和 2 中的所有议题，但只是部份地解决了问题 3 和 4。

EP-0456, 249 A2 (Hewlett-packard Company) 是诸如 CORBA 这样的“中间件”系统与一个部份应用开发环境的集合，该开发环境主要是连接以不同的高级开发语言所开发的已有应用程序。在该发明中所引用的高级语言具有其数据的不同的运行内存内的组织形式，并且必须从一种语言的组合形式向另一种形式传送数据。为此目标，定义了中间数据表述和数据传送命令，以便在编译了该数据表述和传送命令后使得该发明能从一种语言的运行时表述向另一种传送所标识的数据结构。本发明并不是致力于包含一个应用独立运



行时所必须的所有数据和命令的消息格式，只是针对在先前已有的两个应用间传送特定数据所必须的数据和命令。因此，本发明根本没有解决问题 2 和 3 中所述的问题，而且也仅是在编程语言一级上解决问题 1 和 4 中问题，而不是在这里所讨论的应用层上解决问题。相对比，在本发明中，该应用完全由消息构成，因此没有数据从一种语言向另一种语言的翻译，也不需要对所要求的消息格式或命令的编译，所有的数据保存在消息格式中，而不管它是在传送中或驻留在计算机内存中。

在给出上述这些之后，就会感到急切需要一种针对单一系统的通信方式，它解决上面讨论四个问题中的每一个，并且把这些方案溶合成一个统一的整体，这对于那些必须建立复杂的电子数据通信系统和每天必须面对这些问题的个人与公司来说是非常有益的，这样的系统和通信方法将大大降低建立、维护电子数据通信系统的时间、金钱或复杂性。

为取得上述的目标，本发明声明一种通过具有灵活消息格式的消息在一个发送者与一个接收者之间进行点到点通信的方法，其特征在于该消息包括：

一个头标，至少包括消息定义参照、一个发送者标识符和一个终点地址；

消息内容涉及：

\*字段数量及任意字段的内容，

\*对象的数量及任意对象的内容，

\*字段映射的数量及任意字段映射的内容，先前确定的字段可使

用任意的字段映射；

\*动作（action）的数量及任意动作的内容，先前确定的动作至少可使用任意一个事件。

随后，定义应用层功能的所有单元都出现在消息格式中，这些单元早已存在于网络通信系统的内容中，因此可以很容易地通过简单改变一个位置及目标地址来进行分布。

而且，根据本发明，也声明了一种通信设备，它包括处理装置和一个数据库，其通过具有灵活消息格式的消息被配置成用于与另一个通信设备进行点到点的通信，该消息包括：

一个头标，至少包括消息定义参照、一个发送者标识符和一个终点地址；

消息内容涉及：

\*字段数量及任意字段的内容，

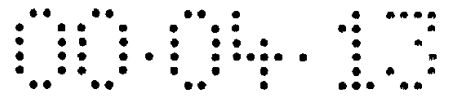
\*对象的数量及任意对象的内容，

\*字段映射的数量及任意字段映射的内容，先前确定的字段可使用任意的字段映射；

\*动作的数量及任意动作的内容，先前确定的动作至少可使用任意一个事件。

该处理装置被配置成：利用消息定义参照作为预先确定的消息定义表的参考，查询存储在数据库中的预先定义的消息定义表。

这样，在本发明中定义了一个灵活的消息格式，它能包含任意类型的数据和完全描述该消息本身内的数据的足够的信息。任意一个消息包含消息定义参照，该消息定义参照由一个接收消息以标识



一个预先装载在该数据库内的消息定义的本发明所声明的通信设备所使用。但是，尽管对于一次或所有情形来说，预先装载的消息定义不是灵活的，但本技术是灵活的。在接收该消息的通信设备内，通过插入、更新或删除关系数据库表中的入口项，可以生成、编辑或删除消息格式的特定事例，该入口项完整地描述了该消息并将其与一个唯一的标识符相参照。而且如果在一个期望的位置缺少消息的定义，可针对特定的消息或消息组自动地交换整个数据库的消息定义，由此为多个用户生成易于交换的通用格式消息。本发明的另一个优点是必要的企业管理费用很少。

在消息定义参照内的字段的例子是一个用于标识消息的任意部份的消息标识符、一个用于对任意消息标识其消息类别的消息类别标识符，例如邮件、商业消息、定单或发货，以及一个用于标识消息的任意部份的版本号的消息版本标识符，和一个用于标识消息的任意部份的创建者的消息创建者标识符。在该消息定义参照内的所有字段都在所声明的通信设备内的消息定义表中有一个副本。

而且，消息的头标可包括一个对加密类型和/或所用压缩类型的参照，而且所有这些参照都会在该通信设备的信息定义表中有一个副本。

最好，该消息的内容通过数字签名的方式加以保护。

最好，该通信设备的数据库是以几个表形式加以组织，而对该表的参照是从消息定义表中来制取。然后，该预先确定的消息定义表包括一个用作为对该数据库中的更进一步表的参照的消息系统标识符。

这些更进一步的表的其中之一是字段定义表，用于容纳对该消息的任意字段的主要定义。

最好这些更进一步的表的其中之一是一个字段映射表，该表包含由前面预先确定的字段所用的映射信息，例如，映射到超文本标记语言字段、数据库字段、平文件字段和其它消息字段。所述的数据库字段和平文件字段存储在客户的数据库中。然后根据本发明的该消息格式不仅包含数据，也包括针对用户接口、客户数据库字段的映射命令以及包含计算和条件逻辑的动作序列。随后，定义应用层功能的所有单元都出现在消息格式中，这些单元早已存在于网络通信系统的内容中，因此可以很容易地通过简单改变一个位置及目标地址来进行分布。而且，以这种方式可以结合进 HTML 标准，用于生成非常容易建立的、快速而又灵活的图形用户接口，允许直接映射到由该发明定义的消息格式或从该消息格式进行映射，由所声明的通信设备进行管理、定义和控制，并与一个连接到该通信设备上的用户终端进行交互并在其上进行显示，象一个微处理器。

另外，这种更进一步的表的其中之一是字段动作表，它包含由预先确定的字段所使用的动作信息。

最好，这些表是包含要对接收或发送的消息进行预先处理而要执行的动作清单的消息预先处理表和要对所接收的消息进行后处理而要执行的动作清单的消息后处理表。

该字段动作表、消息预先处理表和消息后处理表包含对从下列事件组中选出的动作类型的参照：动作的数据类型包含数学计算、赋值、逻辑操作、条件操作的动作逻辑类型和命令等。

最好在本发明中，应用被定义为一个或多个消息的所谓的组。为此，消息定义表包含一个用于标识所接收的消息是否是一个应用的第一消息的应用字段和一个用于参照该应用的名称的应用名字段。

按照本发明，所声明的通信设备最好被配置成：如果所接收到的消息参照的消息定义没有出现在它的数据库中，则向发送者请求一个新的消息定义，并且从所述的发送者接收所述的新的消息定义并将其存储在所述数据库的消息定义表中。

该通信设备的处理装置可配置为或者将所接收到的消息与一个 HTML 文件相融合，或者如果该处理装置没有找到所指定的 HTML 文件就生成一个缺省的动态 HTML 文件。但是，作为替换，该通信设备的处理装置也可以指示一个连接的终端来进行这些功能。因此本发明也涉及一种包含上面所描述的通信设备和连接到该通信设备的终端的系统，该终端包括一个终端处理器、一个显示单元、一个由用户进行数据输入的输入装置，该通信设备被配置成向该终端传递所接收的数据，如果该终端已在该消息中指明是最终地址的话，并且，该终端处理器可配置为或者将该消息与一个指定的 HTML 文件相融合，或者如果该终端处理器没有找到所指定的 HTML 文件就生成一个缺省的动态 HTML 文件。

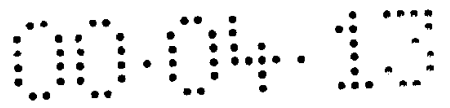
这样，接收一个该终端处理器对其没有相应的消息定义的消息的用户终端就可以根据系统的自动说明和显示来进行应答，其中的显示基于所接收的消息内包含的自我描述信息。如果由该消息所参照的数据库消息定义能够找到的话，同一个自我描述信息就可以被

该终端处理器用来自动生成一个缺省数据库消息定义。这种缺省的数据库消息定义可以被编辑和适应于该终端用户的本地系统。

本发明包含一个对问题 2 和 3 的技术方案的陈述 (state)，它以一种独特的方式与全新的针对问题 1 和 4 的方案综合在一起，它们都是围绕一个独特、灵活的格式电子数据消息管理系统而建立，而该系统又是由所声明的灵活消息格式和通信设备组成。

本发明的消息管理系统（是由存储在一个关系数据库中的表内的详细的信息定义和控制的）能以更灵活的格式实现更有效、完整、简单易用的传送电子数据，该灵活的消息格式可能包括任何形式的电子数据。该灵活消息格式包含允许接收者说明、理解和显示该消息的内容的信息，即使是在以前它从没有接收过。该接收器也可以把这种未知的消息格式加到它自己的消息数据库中并很容易、快速地将其连接进其商业数据库和 HTML 用户接口，而不必编写或改变任何软件。在没有特别声明的、先前已创建的 HTML 文件的情况下，也可以由任何终端的软件动态地生成一个 HTML 用户接口。这解决了第一个议题。

在本发明中，提供了一个与主要关系数据库的 ANSI 标准的 SQL 接口，该接口提供自动地将用户选择的一个数据消息的字段链接到一关系数据库中的该字段的服务，以便随后的存储或检索。这种数据映射接口也可以链接到平文件格式，如固定长度字段和 CSV（逗号分隔值）文件。这种数据映射接口也可扩展为将消息数据字段链接到多种其它的格式，如用于自动处理控制的 SNMP 和制造协议。该数据映射字段由在 ILMDB 中的入口项定义并由 ILMS 控制，这解决了



第二个问题。

在本发明中提供了一个消息控制用户接口终端，其中一个用户可生成、编辑、发送和查看消息。该消息控制用户接口与一个网络和利用本发明事先配置的消息管理服务器进行通信。本质上，本发明替换了针对那些需要完全数据通信能力的个人或公司的当前的网站服务器/网站浏览器，并将该系统域扩展为包含任何数据通信网络。这就使得可以进行完全地发送和接收 HTML、VRML 和 Java 小应用程序以及用户定义的一个特定消息中任何特定数据。当与先前描述的特征进行组合以后，就提供了在一个消息的数据、从中进行检索和存储的客户数据源以及它所用来的显示的图形用户接口（HTML 等）之间的完全的数据库控制链接。

除了上面讨论的议题，本发明还增加了自动定义系统执行的动作的能力，该动作可以作为一个消息在处理之前或之后进行组织，也可以在消息与一个图形用户接口、字段选择动作进行链接的情况下组织。这些动作也可以定义为 SQL 或“Interlingua™ Script”（IS）。本发明定义了“Interlingua™ Script”，它是一个简单、强有力的计算、执行和条件逻辑控制声明的集合，它是以实时方式动态解释和执行的，并且可以与所有的消息数据字段进行交互。这些动作可以产生一个新的消息，可以建立系统功能和外部程序或者发送和接收数据库查询，这此动作也可以用于产生一个自动消息响应、自动字段校验、执行计算、运行 Java 小应用程序或一个复杂的数据库事务等。

在本发明中我们还真正地提出了一种方法，即将应用定义为一

个消息和它的相关动作的集合。由于在消息间的交互的丰富和灵活，其运行数据库事务的能力与图形用户接口相溶合或产生该接口，或作为一个后台在系统中静静地运行，是有可能把一个复杂的应用完全定义为数据消息和与其相关的动作集合。它还在分布式应用处理领域具有非常大的含义，因为此消息作为系统的一项基本功能，将是非常容易发送、接收、理解和组织。这意味着第一次一个分布式应用系统可以很容易、灵活地定义，而不必更改或增加每一个区域的软件代码、本地用户接口和客户数据源接口（如商业数据）而仍然使用同一个应用。

### 附图的简要描述

本发明将参照附图进行描述，这些附只是用于说明的目的而不具有限制本发明的范围的作用。

图 1 示出了按照本发明的系统框图的概览；

图 2a、2b、2c、2d、2e 示出了一个按照本发明的消息格式。

### 优选实施例的描述

图 1 中示了多个服务器  $SRV(m) = 1, \dots, M$ ，其中任一个服务器均可连接到一个或多个终端，连接到服务器  $SRV(m)$  的终端用  $ILMC(n)$  ( $n=1, \dots, N$ ) 来表示，“ILMC”是“Interlingua™ Message Client”的缩写，连接是用 4、5 来编号。

而且，服务器  $SRV(1)$  连接到一个客户数据库  $CDB(1)$ 。尽管只示出了一个客户数据库  $CDB(1)$ ，也可以根据需要应用多个客户

数据库。

终端 ILMC (1) 已经在图中进行了详细的描述, 所示的终端 ILMC (1) 包含一个监视器屏 6、一个处理单元 1、一个键盘 12 和一个鼠标 13。键盘 12 和鼠标 13 被连接到该处理单元 1。当然其它的用户能将数据输入到处理单元 1 的装置也是可用的, 而不仅是键盘和鼠标。

监视器屏 6 示出了一些“按键” 7, 8, 9, 10, 11, 这些按键可由用户通过使用鼠标来起作用, 以便指示处理单元执行一些预定的任务, 这些预定的任务可以是一些选项的打开、建立连接、发送消息、接收消息和/或将数据装进处理单元 1 的内存中。

服务器 SRV (1) 设置有一个称为 Interlingua™ 消息服务器 ILMS 的处理器, 用于按照本发明的方法接收、传递或发送消息。此 ILMS 处理器部份用 IS (Interlingua™ Script) 来表示, 其功能将在下面进行论述。

服务器 SRV (1) 也包括一个数据库 ILMDB (= Interlingua™ Message Database), 用于存储按照本发明的几个表, 也将在后面进行叙述。

服务器 SRV (1) 被配置为通过一个通信路径 2 与其它服务器进行点到点的通信, 该通信路径可以任何一种已知的方式建立, 如有线或无线的。

图 1 也示出连接到一个或多个终端 ILMC 的几个其它服务器 SRV (m), 这些服务器 SRV (m) 也是按照本发明进行组织, 在下面进行描述。

图 2a、2b、2c、2d、2e 给出了消息格式。在图 2a 中，给出的消息格式包括四个字段，即：MSG Id, MSG Class, MSG Version 和 MSG Creator, 用作为接收该消息的任何设备的消息定义参照。

字段 MSG Id (消息标识) 用于定义该消息。

MSG Class (消息类别) 用于指定消息的目录或类型, 如邮件、商业消息、定单或发货等。

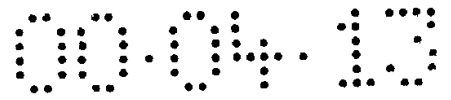
字段 MSG VERSION (消息版本) 包括一个消息的版本标识符, 用于标识所涉及消息的版本号。

第四个字段 MSG CREATOR (消息创建者) 包括消息创建者的标识符, 用于标识所涉及消息的创建者。

MSG Id, Class 和 Version 字段是一个整数值。例如, 每个 MSG Class 中有 1000000 个 MSG Id 和 10000 个 MSG Class。每个 MSG Id 和 MSG Class 对可以有例如 100 个 MSG Version。MSG Class 可以例如按如下来组织。MSG Class 1-20 可以保留用于 ILMS, MSG Class 21-100 可以预先定义为目录, 如定单、介绍、发货等。余下的 9900 个 MSG Class 则给用户定义用。MSG Creator 可以有例如 80 个字符, 这是为了便于标识由特定用户或公司创建的整个消息集, 以便使自动传送很容易。

与其它五个字段一起, 这些消息定义参照形成了一个消息的头标, 这五个其它字段是: SENDER Id、DESTINATION ADDRESS、ENCRYPTION TYPE、COMPRESSION TYPE 和 APPLICATION NAME。

该 Sender Id (发送者标识) 是一个用户名, 由具有附属有服务器 SRV (m) 的名称的服务器 SRV (m) 来加以识别, 并且是以一种例如类似电子邮件地址的方式来进行 (如 user-



xyz@ilserver.com)。字段 DESTINATION ADDRESS (目标地址) 是用于标识所涉及的消息要传送的地址, 该目标地址可以与 SENDER Id 具有同样的格式。

字段 ENCRYPTION TYPE (加密类型) 用于对所施加的加密类型的参照 (如果需要加密的话)。字段 COMPRESSION TYPE (压缩类型) 用于标识所施加的压缩的类型 (如果要求的话)。

字段 APPLICATION NAME (应用名) 用于标识所涉及的该消息是否是构成一个应用的一系列消息中的一员。

Field Count (字段数量) 给出了字段数量, 例如从 0 到 256。一个“字段”其内容是指一个特定的数据类型和数据本身。实际的字段消息跟随在该字段之后 (如果数量是大于 0 的话)。参照图 2b, 一个字段由一个 TYPE (类型) 标识符、SIZE (大小) 值、用于内部识别的字段 Name (名称)、用于缺省显示的字段 Label (标签) 和同样用于缺省显示的字段 Description (描述) 来定义。其后的数据长度可以通过组合 TYPE 和 SIZE 字段知道。一个字段的定义直接跟随在另一个之后, 中间没有分隔符。

经常, 尤其是在 HTML 型数据中, 通常有一个对包含二进制图像、声音或程序数据的本地文件的参照。为了保证正确的传送和对照这些外部引用对象, 在该消息的末尾通常将它们自动地进行编组, 如图 2a 所示, 让参照它们的字段指向这些对象。当接收该消息时, 这些对象自动地进行本地存储, 然后该对象参照自动地调整为访问在新的位置上的对象。Object Count (对象数量) 字段给出了在该消息中的对象的个数, 范围是从 0 到 256。对象数据紧随在该字段之后 (如果该数量是大于 0 的话)。一个对象最好由 MIMEType (MIME 类型) / 内容标识符、一个 SIZE (大小) 值以及实际的对象数据来

定义，如图 2c 所示，一个对象的定义直接跟随在另一个之后，而没有另外的分隔符。

参照图 2d，一个字段映射例如可由 FIELD IDENTIFIER（字段标识符），MAPPING TYPE（映射类型）和 MAPPING DATA（映射数据）来定义，如图 2e 所示，动作可以简单地由 ACTION TYPE（动作类型）及其后的 ACTION DATA（动作数据）定义。

一个使用服务器 SRV 钥的数字签名最好在整条消息上创建并加到该对象清单的末尾。

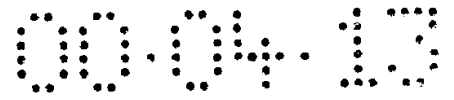
在图 2a、2b、2c、2d 和 2e 中定义的可用的数据类型举例如下：

- 1) 统一编码字符串；
- 2) 独立于计算机硬件的任意大小的整数；
- 3) IEEE 8 位格式浮点数；
- 4) 统一资源定位符；
- 5) GIF 和 JPEG 格式的图像数据
- 6) MIMI 声明的数据表述。

事实上，GIF 和 JPEG 数据也可以包括在 MIME 类型。在这些格式中对图像指定一种特定的数据类型的判定是由它们在 HTML 类型显示系统中的使用频率来驱动的，最经常使用的数据类型有其自己的定义，任何其它的数据类型都可以包括为已有的 MIME 类型或特定的 MIME 扩展，这此数据类型仅由举例的方式来声明，本发明的保护范围并不仅限于优选的数据类型。

每一个消息都可以有其自己的数字签名和可选的加密和、或压缩，这是由服务器 SRV (m) 自动地管理的。

### 消息数据库



一个灵活的消息格式定义管理和控制系统通过使用 ILMDB 数据库来实施，该数据库作为一个关系数据库包括：

- a) 详细的消息定义；
- b) 消息映射指令：
  - b. 1) 映射到客户数据库字段
  - b. 2) 映射到客户平文件格式
  - b. 3) 映射到客户用户接口（HTML）字段；
  - b. 4) 映射到其它消息字段；

- c) 消息事件列表：

(SQL, Java, 计算、逻辑流、消息呼叫、外部程序呼叫)

- c. 1) 消息预处理动作；
- c. 2) 消息后处理动作；
- c. 3) 消息字段用户接口事件动作。

该信息保存在关系数据库 ILMDB 的表中，并且允许系统完全地说明和处理任何消息，可以增加新的消息定义，也可以改变已有的定义。消息定义可以从一个服务器 SRV (m) 发送给另一个服务器以便组合进消息数据库 ILMDB。每一个灵活的消息格式包括足够的自我描述信息以便使服务器 SRV (m) 解释一个先前未曾见过的消息，然后在 ILMDB 中自动地为其生成一个新的消息定义入口。该新的消息可以由一个系统管理员编辑或增强功能以便完成映射进该接收者的系统。

消息数据并不保存在数据库 ILMDB 中，仅是其定义而已。消息数据被保存在一磁盘存储系统上的压缩文件中，根据由系统管理员

选择的服务器消息周期选项，这些文件被周期性地归档或删除。

在数据库 ILMDB 内需要定义且控制或者接收或发送的消息的主要关系表如下：

表名	描述
msgdef	InterLingua™ 消息定义
flddef	消息字段定义
fldmap	字段映射
fldact	字段动作定义（每字段一个动作）
msgpre	消息预先处理（消息动作序列）
msgpost	消息后处理（消息动作处理）

在本技术领域内熟练的系统开发者会意识到：会有很多其它的相关信息表能使整个系统更易于使用且可控制次级系统特征，这其中的一个例子是所有与用户帐户和用户信息、系统管理与系统安全相关的表。而这里所提供的表是按照优选实施例实施本发明所必须的一些核心表，其它的扩展和增强都将在本发明的范围内。

表：msgdef

该表用于保持消息的主要定义，一个消息完全由该表的前四个字段来标识，第五个字段（msysid）是由服务器处理器 ILMS 内部分配的一个标识，并且也用作为对该数据库内的其它表的标识和快速访问的参照。

字段名	类型	描述
msgid	整数	消息标识
msgver	整数	消息版本
msgclass	整数	消息类别
creatid	字符	创建者标识符
msysid	整数	消息系统 Id
creatdate	日期	创建日期
usedate	日期	使用日期

*这两个字段控制一条消息在 ILMS 中存储的时间*

durdays	整数	周期天数
durmins	整数	周期分钟

*这两个字段标识加密类型和数字签名算法*

enctype	字符	加密类型
sigtype	字符	数字签名类型

*该字段指示消息是否显示*

silent	布尔	消息静默标志
--------	----	--------

*该字段指示该消息是否能初始化或响应事件*

active	布尔	处理使能标志
msgicon	字符	消息图标文件（用于 ILMC 显示）

msghtml          字符          消息 HTML 文件（用于 ILMC 显示）

*该字段指示该消息是否是应用中的第一个*

appmain          布尔          应用主标志

*如果该消息是应用的一部份则该字段包含一个名称*

appname          字符          应用名称

descr            字符          消息描述

#### 表: flddef

该表用于保持对所有消息的所有字段的主要定义，一个字段唯一地由 'msgdef' 表中的 'msysid' 或一个字段序列标识号码 (fldseq) 或一个字段名 (fldname) 来标识。

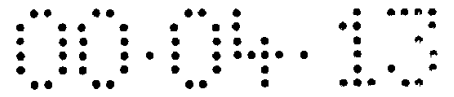
字段名	类型	描述
msyid	整型	消息系统 Id
fldseq	整型	字段序列
fldname	字符	字段名

*这三个字段用于生成 1、2（表）或 3 维阵列*

ar1              整型          阵列维度 1

ar2              整型          阵列维度 2

ar3              整型          阵列维度 3



fldtype          字符          字段数据类型

*该字段的大小根据不同的数据类型有不同的说明*

fldsize          整型          字段大小

*如果该字段是用于对另一个作为一个数据结构的消息的参照，  
则使用该字段。*

Smsysid          整型          子消息系统 Id

*如果该字段包含 MIME 数据，则使用这两个字段*

mimecont        字符          MIME 内容标识符

mimetype        字符          MIME 类型标识符

fldlabel        字符          字段标号（用于 ILMC 显示）

fldcom          字符          字段注释（用于 ILMC 显示）

#### 表: fldmap

该表用于保持一个字段所用的所有映射信息，该表可以定义到数据库字段、平文件字段和其它消息字段的映射，该字段以与表‘flddef’同样的方式进行标识。对于每一字段，有一个针对数据的映射和一个用于显示的映射，但在一个字段被定义为一个阵列时除外。一个数据映射和一个显示映射可被分配给一个或多个元件或一个阵列的维度（dimensional）行。

<u>字段名</u>	<u>类型</u>	<u>描述</u>
msysid	整型	消息系统 Id
fldseq	整型	字段序列
fldname	字符	字段名

这三个字段允许对特定阵列单元或维度行的映射

ar1	整型	阵列维度 1
ar2	整型	阵列维度 2
ar3	整型	阵列维度 3

这个字段允许该数据从在同一个消息文本内的最近活动的前一个消息进行传送，对于一个特定的用户连接来说，该消息是当前活动的消息集。

cpyfld	字符	拷贝字段（从 Msg）
--------	----	-------------

这三个字段用于声明 SQL 可访问的关系数据库中的一个特定的字段。

dbnam	字符	数据库名
tblnam	字符	表名
fldnam	字符	字段名

该字段声明 HTML 文件 (msgdef.msghtml) 中一个用于显示的字

段。

tagnam	字符	HTML 标签名
--------	----	----------

这两个字段声明一个固定文件长度或逗号分隔值的平文件。

flatfile	字符	平文件名
----------	----	------

filetype	字符	文件类型（固定的，CSV）
----------	----	---------------

这四个字段声明了该平文件的字段位置。

rowbegin	整型	行开始
----------	----	-----

colbegin	整型	列开始（或 CSV 的字段#）
----------	----	-----------------

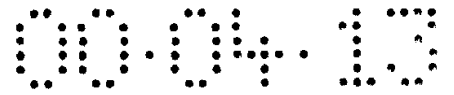
rowend	整型	行结束
--------	----	-----

colend	整型	列结束
--------	----	-----

表: fldact

该表用于保持一个字段可能使用的所有动作信息。动作可以是数据库类型的动作或逻辑类型的动作。每个字段有一个动作，除非该字段被定义为一个阵列。一个动作可以被分配给一个或多个单元或一个阵列的维度行。

<u>字段名</u>	<u>类型</u>	<u>描述</u>
msysid	整型	消息系统 Id



fldseq	整型	字段序列
fldname	字符	字段名称
ar1	整型	阵列维度 1
ar2	整型	阵列维度 2
ar3	整型	阵列维度 3

*该事件定义了动作类型：SQL 或 IS、文件或本地*

acttype	字符	动作类型
---------	----	------

*该字段包含 SQL、IS 或一个其内容是 SQL 或 IS 的文件名*

actscript	字符	动作代码
-----------	----	------

#### 表：msgpre 和 msgpost

这两个表 msgpre 和 msgpost 具有一个相同的结构，表 msgpe 拥有要由服务器处理器 ILMS 执行的动作清单，作为对一个特定消息的预处理。这意味着在任何显示或字段动作发生之前，该服务器处理器 ILMS 完成该清单或预处理功能。表 ‘msgpost’ 拥有要由服务器处理器 ILMS 执行的动作清单，作为对一个特定消息的后处理。这意味着在所有其它消息动作或显示完成之后，在从主存储器中删除该消息之前，该服务器处理器 ILMS 完成该后处理功能的清单。该处理序列号用于生成该动作清单的一个控制执行。先前的序列号必须在下一个开始之前完成。如果有不只一个动作具有同一序列号，它们就同时开始或者，在一个多 CPU 或大规模并行环境内作为真正的并

行处理。

表: msgpre

<u>字段名</u>	<u>类型</u>	<u>描述</u>
msysid	整型	消息系统 Id
procseq	整型	处理序列

*该字段定义动作类型: SQL 或 IS、文件或本地*

proctype                      字符    预处理类型

*该字段包含 SQL、IS 或一个其内容是 SQL 或 IS 的文件名*

proscript      字符      预处理代码

表: msgpost

<u>字段名</u>	<u>类型</u>	<u>描述</u>
msysid	整型	消息系统 Id
procseq	整型	处理序列

*该字段定义动作类型: SQL 或 IS、文件或本地*

proctype      字符      后处理类型

该字段包含 SQL、IS 或一个其内容是 SQL 或 IS 的文件名

procsript 字符 后处理代码

### InterLingua™代码（动作定义和控制语言）

针对消息预处理、后处理、或字段动作的任何动作可以是一个 SQL 动作或 IS 动作。SQL(结构化查询语言)是针对关系数据库的 ANSI 标准接口语言，而 IS (InterLingus™Script) 是服务器处理器 ILMS 的一个预先确定部份（参照图 1），且是一个小但强有力的计算、执行条件逻辑声明的集合，它能给消息一个应用层功能。在一个优选的实施例中，此 IS 的定义如下：

#### 计算

根据所要处理的整型和浮点型值可以进行加（+）、减（-）、乘（\*）、除（/）操作。括号‘（）’用于对数学操作进行分组。

#### 赋值

赋值（=）用于整型、浮点型值、字符串和 URL。

#### 逻辑和条件

符号表述：根据整型、浮点型值、字符串和 URL，可以使用以下字符：与（AND）、或（OR）、大于（>）、小于（<）、大于等于（>=）、小于等于（<=）、等于（=）、不等于（! =）和非（!）

#### 控制流

END-MSG- 结束一条消息，该消息必须是其自己预处理清单中的最后一个命令。

END-APP- 结束一个应用，它必须是一个成员消息后处理清单中的最后一个命令。

For 循环:

```
FOR i=x TO y
  BEGIN
    <声明>
  END
```

While 循环:

```
WHILE <条件>
  BEGIN
    <声明>
  END
```

条件结构:

```
IF <条件> THEN
  BEGIN
    <声明>
  END
ELSE
  BEGIN
    <声明>
```

END

### 命令

装载映射字段数据:

LOAD {WHERE <外部字段名>=<值>}

存储映射字段数据:

STORE {WHERE <外部字段名>=<值>}

执行外部程序:

CALL [JAVA | EXEC | SHELL] <程序名> <参数 1>.... <参数 n>

执行消息:

ILMP: //<主机名>/<消息 id>

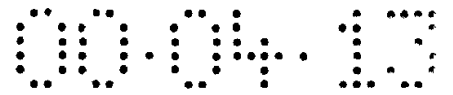
执行内部功能（这主要用于用户接口动作）

ILMP://<主机名>/<消息 id>/<字段 id >[UI-ACT | MSG-END | APP-END]

ILMP://<主机名>/<消息 id>/<字段 id >[ADD-DATA | DEL-DATA | CHG-DATA]<数据>

### 应用

一个 InterLingua™ 消息的集合可以被组成在一起，单独地命名



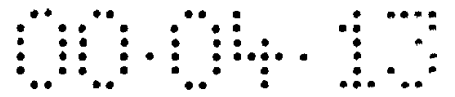
并用为 InterLingua™ 应用。为了便于讨论，对于任何编程语言或系统平台，都可以将一个应用定义为一个或多个其执行序列可以根据输入数据、用户和系统事件而改变的函数，它可以与外部存储设备访问、交互，也可以通过用户接口与用户进行交互。

这种所谓的消息集合的全部或一部份可以驻留在一个或多个位置处的一个或多个服务器 SRV (m) 上，这种所谓的消息集或应用实现：

- a) 一个直接的函数序列
- b) 本地或共享数据
- c) 用户接口屏幕
- d) 网络分布式应用功能

该直接序列函数可包括所有的动作类型，如数据库查询、计算以及其它消息的激发。在本文中激发一个消息的使用相当于调用一个应用函数、显示下一个用户接口屏或，发送一个消息给一本地或外部的网络目标。该操作顺序的实时结构化流，或者换句话说，该直接序列函数由消息动作表和由该用户接口产生的事件动作（如一个字段的鼠标选择或一个按键）所决定的。

该本地和共享数据包括所有允许的数据类型，这些数据类型可以编组成数据结构。该消息数据也可以映射到用户数据库、用户接口或从其中获得，并被传递到另一个消息。这种思想可以使用消息作为一个函数，它有其自己的本地数据和可调用其它的函数（消息），可以向该函数传递所有的本地数据。当该本地数据在两个或更多的消息之间传递时，它就变成了共享数据。



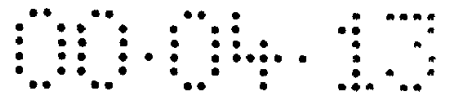
该网络化分布式应用功能性是通过使用 InterLingua™ 客户 ILMC (n) 和服务器作为消息的发送者或接收者的内部功能来实施的。与一个在单一服务器上运行的单一应用相对比，如果所生成的应用集是以分布环境运行在多个服务器上，其中的主要差别是作为一个应用功能而动作的一个或多个消息将使一个远程服务器作为其目标，而不是本地服务器，这就使得生成网络化分布式应用非常直截了当，具有前所未见的简单和易用性。

### InterLingua™ 消息客户 (ILMC) 和服务器 SRV

InterLingua™ 消息客户 (ILMC) 的软件最好完全由 Java 来写，这样就可以使其运行在 Java 使能的计算机系统上，这就几乎包括了所有通用的计算机。ILMC 作为一个 HTML/VRML/Java 小应用显示控制台和对服务器 SRV (m) 的网络通信接口。

该用户接口屏幕利用这种能力来将一个消息与一个 HTML 定义相关联，并映射该消息和 HTML 显示间的数据。这就使得所选择的功能 (消息) 具有所需的用户接口。用户键盘和鼠标事件被连接到字段动作。

这种实施从一个服务器到另一个服务器以及一个服务器与一个终端 ILMC 之间的消息发送/接收功能的通信子系统，被设计成以一种高度优化、多线程方式运行，并通过使用数字签名来替换使用监视消息的集成。这就可以保证对所接收到消息的非损坏状态的完全确认，并且同时，对发送者的标识和认证的安全校验。此通信子系统存在于服务器 SRV 和终端 ILMC 之间，该通信子系统也设计成与网



络通信协议无关。在当前的版本中，它运行在 TCP/IP, XTP/IP, 和 XTP/ATM/AAL5, 但是它也可以很容易地运行在 AppleTalk, IPX, SNA, X. 25, DECNET, OSI 以及其它所有可使用套接字接口和点到点访问方案的协议。

该通信子系统并不是本发明的一个关键部份，本发明的优点可以通过很多不同类型的通信子系统得以实施。在本技术领域内的人会意识到，所选择的网络通信子系统的类型和实施细节并不会改变本发明的功能性，只要此通信子系统具有可靠的传送协议，以保证高速地传送成功和数据集成。

在欧州专利 97202945.8 中所声明的消息优先权和服务机制可以增加至该通信子系统，这种机制对不同的优先级别使用了独立的连接以及一种简单网络和系统缓存算法，以自动地保证对所有的消息优先权级别的公平服务。这种机制能快速且连续地传送高优先权消息，而又不需要一个复杂的规划算法来监视消息的大小或防止低优先级别的消息停止 (lock-out)。

### 接收一个消息

当通过一个服务器 SRV(m) 接收消息时，它就会通过通信子系统，将检验数字签名并剥离传输层的头标。如果该消息已经加密或压缩，就会在此时进行解密和解压缩，然后消息被传送给数据库层 ILMDB。

如果在希望的位置缺少消息定义，将针对特定的消息或消息组自动地交换整个数据库消息定义，由此为多个用户生成易于交换的通用格式消息。

作为替换地，如果对于所接收到的消息没有存在一个消息定义，它就会发送给系统管理员的收件箱，该管理员然后就可以使用终端 ILMC 检查该消息，自动地增加基本的消息定义并将该消息字段连接到本地数据库字段和 HTML 显示字段（如果必要）。该系统管理员也可以自动地请求该消息的发送者发送完整的消息定义，该消息定义然后自动地被放置在本地数据库 ILMDB。该请求出也可以作为一个消息从一个服务器发送到另一个服务器。

如果对于所接收到的消息，存在有一个消息定义，它就会从数据库 ILMDB 中读出。由该服务器 SRV (m) 执行消息预处理动作清单，起到消息字段校验、数据库查询、计算和其它消息的功能。如果一个消息被发送给一个用户用于显示，则终端 ILMC 的处理器 1 或者将该消息与所指定的 HTML 文件进行溶合或者，如果处理器 1 没有发现所指定的 HTML 文件，它就会生成一个缺省的动态 HTML。该字段动作可以特定鼠标和键盘触发（如果该字段动作已在消息中加以定义）。字段动作可以调用一个标准的 HTTP URL，发送一个显示本地图像的请求，触发一个 SQL 数据库事务，与更新本地显示数据或运行一个 InterLingua™ Script (IS) 命令。如果该消息没有定义了的用户接口，它就标记为“静默”，并且在系统中只能由管理员看得见它的存在。

在完成了静默消息预处理及遇到 END-MSG 命令之后，或者该终端消息用户接口发送一个 END-MSG 命令之后，服务器 SRV (m) 执行消息后处理动作清单。如果该消息是一个应用集的一部份，它就保留激活状态和驻留在内存中直到服务器 SRV (m) 遇到或接收 END-

APP。当一个应用集完成后，如果所有的成员消息后处理清单还没有被执行，就由服务器 SRV (m) 来执行它们。这种功能性也保证了事务文本和应用数据及状态的自动维护。当一条单个的消息或应用集完成后，将清除该系统的内存并返回供其它消息和应用使用。

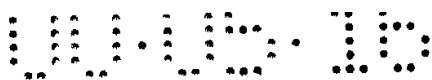
### 发送一个消息

从一个终端 ILMC:

一条消息格式 (ILMF) 还没有从 ILMDB/-ILMS 请求或接收。该消息的内容和地址可以由该终端用户增加。该终端用户通过一个用户接口动作起动一个消息发送到该用户已在其上进行注册了的服务器。

从一个服务器 SRV (m):

一条消息可以通过 InterLingua™ Script (IS) 作为直接呼叫的结果自动地起动以便触发服务器 SRV (m) 上的一条消息，或者也可以作为首先接收并由服务器 SRV (m) 处理的、一个连接的终端的请求。如果所有消息都进行了定义，则该服务器执行消息的预处理。如果该目标地址是针对服务器本身，则在遇到或接收到一个 END-MSG 后，就执行该消息的后处理，然后终止该消息；如果其目标地址是针对另一个服务器的，把该消息传送给通信子系统，经过压缩、加密或给一个数字签名后，通过点到点连接发送给接收服务器。



## 缩写清单

ANSI=美国国家标准委员会

CORBA=共同对象请求中间结构

CSV=逗号分隔值

EDI=电子数据交换

GIF=图形交换格式

HTML=超文本标记语言

HTTP=超文本传输协议

ILMC=InterLingua™ 消息客户

ILMDB= InterLingua™ 消息数据库

ILMF= InterLingua™ 消息格式

ILMP= InterLingua™ 消息协议

ILMS= InterLingua™ 消息服务器

IS= InterLingua™ 消息代码

JDBC=Java 数据库连接

JPEG=联合图像专家组

MIME=多目标因特网邮件扩展

ODBC=开放数据库连接

RDBMS=关系数据库管理系统

SNMP=简单网络管理协议

SQL=结构化查询语言

URL=统一资源定位符

VRML=虚拟现实标记语言

# 说明书附图

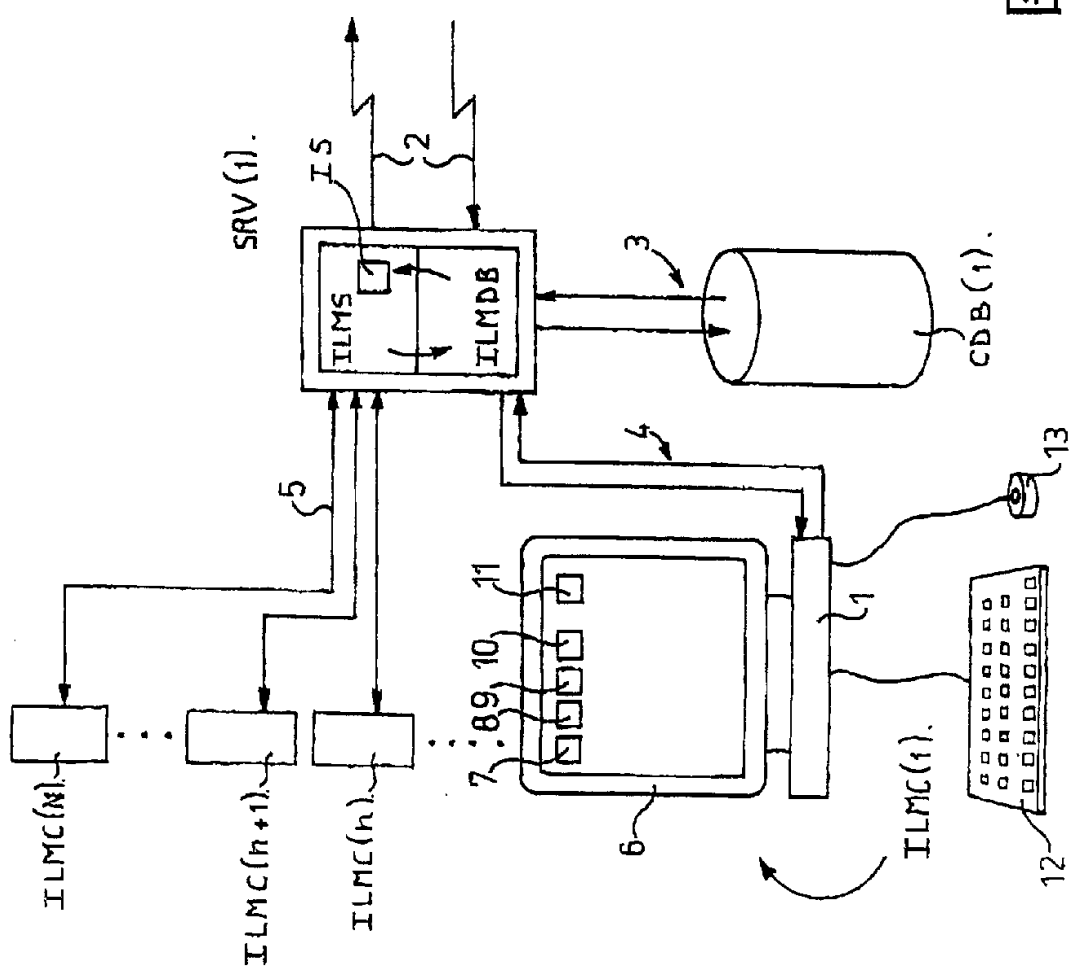
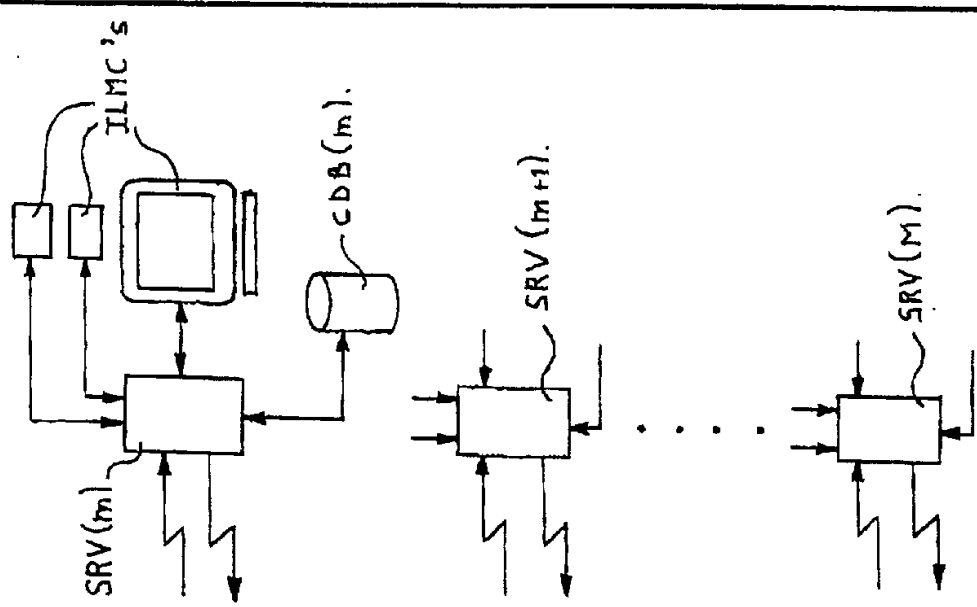


图 1

消息标识	消息类别	消息版本	消息创建者
发送者标识		目标地址	
加密类型	压缩类型	应用名	
字段数量	FIELD (1) .....		
对象数量	OBJECT (1) .....		
映射数量	MAP (1) .....		
动作数量	ACTION (1) .....		
数字签名			

} 头标

图2a

