

银行国际贸易结算系统构件化设计与实现

余金山¹, 陈志强²

(1. 华侨大学电脑系, 福建泉州市 362011; 2. 中国银行澳门分行电脑部, 中国 澳门)

[摘要] 国际贸易结算是银行业中一种必不可少的业务, 占据着重要的地位, 而且随着电子商务的发展, 其重要性也越来越突出。但是, 这一业务的计算机化相当困难。主要原因是: 操作流程复杂; 组合变化太多, 对灵活性有很高要求; 涉及面相当广, 几乎涉及银行的每一个业务品种, 是一个压缩版的银行系统。因此, 开发这样的系统必须采用先进的而且较为合适的方法和技术。文章提出了以构建半成品构件系统为总体思想的开发方法; 在设计和实现上, 提出了以数据总线为统一接口, 以菜单/功能图标为总控主体框架, 以基本业务操作作为基本组件, 并能把基本组件包装裁剪成可在主体框架控制下运行的功能构件, 这样构建的系统可按实际需要进行组装、剪裁、参数设定、重新包装或经少量改写而形成一个实际可运行的系统。并提供了与银行总系统集成的灵活接口, 具备了向网上银行和分布式系统发展的良好基础。

[关键词] 银行; 国际贸易结算系统; 构件; 半成品系统

[中图分类号] TP311.5; TP391 **[文献标识码]** A **[文章编号]** 1009-1742(2004)09-0081-07

1 引言

银行业是计算机应用较多, 较具规模和历史较长的一个领域。存、放、汇是银行的基础业务, 作为现代银行的职能, 国际贸易结算业务是必不可少的一种业务, 它是除了放款以外的最主要盈利来源^[1]。

近年来, 由于因特网技术快速蓬勃的发展, 至使各类应用系统得以活跃于因特网上, 尤其是电子商务活动。所谓电子商务, 简单而言, 就是利用互联网技术, 把商品/原料的采购、物料运输和国际贸易结算信息流在网络上实现, 使得商务在买方、卖方间透过互联网展开。银行在国际贸易结算业务中一直处于不可缺少的地位, 但在网际空间这个崭新的媒体中, 如何介入电子商务, 履行“支付”的中介人角色, 成为国际贸易结算业务转型的重要契机^[2]。中国已加入世贸组织, 国际贸易结算系统

的开发、完善和国际社会接轨, 更是刻不容缓的大事。

然而, 这一业务的计算机化相当困难, 主要原因是: 涉及面相当广, 几乎涉及银行的每一个业务品种, 是一个压缩版的银行系统; 操作流程复杂——在整个进出口贸易的环节中, 都有相应的协作处理, 以配合进口商和出口商之间的往来; 组合变化太多, 由简单的汇款, 或单据托收, 或信用证, 或中间有否融资等, 是由客户根据需要而组合, 收费利息计算等, 都要灵活组合, 故此, 对系统设计、程序编写的要求都很高, 而且在计算机化过程中存在不少不利因素^[2]。过去, 一间银行的计算机程度中的一个指标, 就是该银行在国际贸易结算业务中计算机系统的自动化程度。因此, 开发这样的系统必须采用先进的而且较为合适的方法和技术。笔者适逢参与了所在单位有关系统的设计和开发工作, 对银行国际贸易结算业务系统进行了深入

[收稿日期] 2003-12-17; 修回日期 2004-02-03

[基金项目] 福建省自然科学基金资助项目(A0210018)

[作者简介] 余金山(1952-), 男, 福建泉州市人, 华侨大学教授, 主要研究领域为软件工程、网络计算和人工智能应用。

的研究,做了相当长时间的实践。认为这样的系统应该应用信息工程的理论方法,并采用以用户为中心的技术对系统进行分析和设计。本文提出了以构建半成品构件系统为总体思想的开发方法;在设计和实现上,提出了以数据总线为统一接口,以菜单/功能图标为总控主体框架^[3,4],以基本业务操作为基本组件,并能把基本组件包装裁剪成可在主体框架控制下运行的功能构件技术,这样构建的系统可按实际需要进行组装、剪裁、参数设定、重新包装或经少量改写而形成一个实际可运行的系统。这样的系统还提供了与银行总系统集成的灵活接口,具备了向网上银行和分布式系统发展的良好基础。

2 相关概念

银行国际贸易结算系统设计与实践中使用的思想和相关概念,主要为:

1) 数据总线,是一个按一定标准构成的数据交换区/数据暂存区。它为系统的各组成部分,即总控主体和功能构件之间以及各功能构件之间的数据交换和通信提供标准接口。

2) 基本组件,是能完成某项基本业务操作的程序组件,基本业务操作是指业务发展到某一阶段便需要进行的一项操作。

3) 功能构件,是一个或多个基本组件的包装,它的外部接口是数据总线。调度它们运行的是菜单/功能图标总控体。功能构件能按照具体的业务交易即定的操作流程去完成相应的业务操作。

4) 菜单/功能图标总控体,是一个以菜单/功能图标为用户界面,能够插入和组装功能构件,并可以通过参数进行控制的总控程序,是系统的主体框架。

5) 半成品的构件系统。它是一个针对某个特定应用领域的、且至少包含有菜单/功能图标总控体和该领域的基本业务组件的系统^[5,6]。通过把基本组件包装成功能构件,再对功能构件和总控主体框架进行裁剪和组装即可构成一个实际可运行的应用系统。当业务发生变化时,可以对系统进行调整或重构。这些工作均可由用户自行操作。构件系统本身也可能对某些组件做了包装,也可能已进行了一定的裁剪和组装,此外,该系统还允许用户添加自己的基本业务组件或对已有的某些成分进行少量的改写。

3 数据模型

根据长期的实践,并在实践经验的基础上对国际贸易结算系统进行了深入和详尽的分析,总结出了该领域的特点和共性。认为构成系统的主体是各种业务品种或信息差异比较大的环节,因此,国际贸易结算业务半成品构件系统可以将整个业务系统分割成若干个模块。模块分割的原则就是业务品种或者信息差异较大的环节。在总体思路上,以数据模型的建立作为整个设计过程的基础。这个数据模型一方面应该按企业需要的信息和数据来制定,它应该是整个企业机构而非局部的一个应用,另一方面它应能满足一个半成品构件系统的需要和运作时所需的结构。数据模型由数据结构和数据总线组成。

3.1 数据结构

首先需要有一个数据结构去存放业务的数据,而且当数据更新时,很多时候还需要将每个时刻的状态保留下来,称历程记录;已经完成处理的记录,亦需要定期抽离,形成历史记录。其主要组成如下:

1) 主表:存放模块记录的数据表;

2) 历程记录表:一个和主表结构基本相同,用来存放某个时刻的数据切面;

3) 历史记录表:一个和主表结构基本相同,用来存放已经完成处理的记录,这些记录需要定期抽离,形成历史记录来存放;

4) 暂存表:一个和主表结构基本相同,用来存放正在编辑的数据,作为一个缓冲寄存的地方,可以让用户有足够的时间去编辑主表以外的文档文件、电讯报文等,而中途又可暂存起来。

因为,要实际开发的是一个半成品构件系统,在数据结构的建立期间,既要考虑到领域的实质要求,亦要抽离具体,精心设计。

另外,很重要的一点是,既然被开发的是一个半成品构件系统,就不能具体定义实质的模块和它的属性。这个步骤,要留待完成半成品构件系统后,进入剪裁系统至实际应用时去完成。用数据字典的方法来存放模块属性定义,它是半成品构件系统的一个参数表。

3.2 数据总线

数据总线主要应用在系统控制主体和嵌入功能构件间,传送标准的数据交换、数据通信和控制信

息。若一个功能处理要由一组程序去完成，无论这一组程序是以调用嵌套的方式处理或是串行的一个跟一个地进行操作，亦力求以数据总线作为它们之间的数据交换、数据通信和控制信息传送之标准参数。从而为系统的可组装性和可裁剪性提供支持。

另外，数据总线也作为国际贸易结算系统和银行其他子系统之间的接口标准。银行子系统之间的接口，也是随着时间的推移而在不断变化的，例如，笔者所在的银行就经历了这样的变动：早先是利用手工输入数据，打印数据传到另一系统的方案，称人工接口。后来，将手工输入和打印，变成电子文件，再用程序录入，变成半自动式的结构。如果需要实时响应的，接口部分就需要考虑用数据通信的方法传送数据，或者改写部分系统结构。而到目前为止，尚无法确定接口应做到哪一个层次。数据总线的应用为子系统之间的接口提供了灵活的支持。

数据总线由两部分组成：

1) 数据交换暂存区。它包含下列内容：a. 主表数据存贮工作区；b. 主表数据字典临时存放区；c. 常量、工作变量存放区；d. 当前环境参数存放区：如当前模块识别码、会计日期、当前操作员、当前终端位置、打印设备参数、系统工作模式等；e. 功能处理控制参数：如当前功能项目代号、操作代码、数据更新控制字等等。

2) 客户数据模块和代理银行数据模块。客户，作为银行系统的最主要的必然对象，有必要将其专有的信息存放下来。而且从整体银行的角度看，客户的信息应不仅只供单一的系统应用，而是供银行的所有业务共享的。

客户的数据，设计成系统中的一个模块，具有其他模块的基本功能和特点，如：可以在剪裁系统时才具体定义模块的属性。但因为其特殊性，它具有其他模块没有的特性和功能。

代理银行，是指在结算业务中，协办业务的另一方银行，亦包括资金、外汇清算的代理银行。同客户数据类似，系统中存放的是银行整体代理银行档案中的一个分支部分，这里亦以一个特殊模块的模式出现。

4 模块操作和功能构件

对模块所属的业务交易操作，是一组功能项目的组合。业务操作一般是按业务发展到某一阶段，

进行的一项操作，相应地需要设计成不同阶段会用到的功能项目。将这些操作，设计成一个个的小程序组件，并包装成为一个个实用构件。这些构件配合菜单选项或功能图标，便形成了各种各样的模块功能项目。执行功能项目，按照既定的操作流程去完成各种业务操作。

功能构件是一个或多个组件的一个包装，它们的外部接口是数据总线，调度它们运行的是菜单/功能图标总控体。功能构件本身并不能马上操作起来，只是经过细心考虑、设计和编写出来的组件。当进一步要把系统投入生产，必须经过剪裁系统的步骤。剪裁系统的设计目标是给一些经过简单训练的业务人员进行，但不是每一个前线业务人员，当然是可以由信息科技部门的人员担当。所谓剪裁系统，主要的工作是通过一些操作平台，去为功能构件或总控程序，填入具体的属性和参数值。

经过组合选取，设计并实现了基本业务功能构件29个，系统管理和维护构件8个。虽然限于篇幅无法在此一一详加介绍，但下面的几个实例已足以说明功能构件和基本组件的概念及其设计意义。

1) 新增确认的业务交易。直接在模块主表中生成记录。构件由取默认值组件、显示输入域组件、读取和校验输入域组件、第二次读入数据组件、条件式执行组件、表达式执行组件、授信额度检查组件、权限控制检查组件、会计分录生成组件、利息计算历程记录生成组件、输出显示/打印组件、文档文件生成组件、文档文件编辑组件、电文输出组件、记录状态控制组件、主表读写组件等组成。

2) 衍生新的确认业务交易。根据已有的业务交易，衍生新的交易。所谓衍生，可以由原来模块的其他记录衍生出来，也可由别的模块衍生出来。构件由数据传递传入组件、取默认值组件、显示输入域组件、读取和校验输入域组件、第二次读入数据组件、条件式执行组件、表达式执行组件、授信额度检查组件、权限控制检查组件、会计分录生成组件、利息计算历程记录生成组件、输出显示/打印组件、文档文件生成组件、文档文件编辑组件、电文输出组件、记录状态控制组件、主表读写组件、数据传递输出组件等组成。

3) 新增待确认的业务交易。此构件先在暂存表上存放将会输出至主表之数据及控制信息，再由“确认处理构件”生成模块主表记录。构件由取默

认值组件、显示输入域组件、读取和校验输入域组件、第二次读入数据组件、条件式执行组件、表达式执行组件、授信额度检查组件、权限控制检查组件、会计分录生成组件、利息计算历程记录生成组件、输出显示/打印组件、文档文件生成组件、文档文件编辑组件、记录状态控制组件、暂存表读写组件等组成。

4) 直接修正(非金融性)业务交易。根据已有的业务交易,修改某些较次要的域,并直接更新模块主表记录。所谓非金融性,是指不会有授信额度、会计分录、利息计算等处理的功能项目,不会有数据传递的发生,甚至连记录状态亦不会更改。构件由主表读写组件、取默认值组件、显示输入域组件、读取和校验输入域组件、第二次读入数据组件、条件式执行组件、表达式执行组件、权限控制检查组件、输出显示/打印组件、文档文件生成组件、文档文件编辑组件、电文输出组件等组成。

5) 成批更新(含金融性)业务交易。同样是根据已有的业务交易,修改或调整某些域,并直接更新模块主表记录,但只根据预设定的条件或记录状态去选定主表业交易记录,没有数据传递的传入或输出,没有输入或输出显示画面,只限成批更新主表本身。构件由主表选择条件处理组件、主表读写组件、取默认值组件、第二次读入数据组件、条件式执行组件、表达式执行组件、权限控制检查组件、输出显示/打印组件、文档文件生成组件、电文输出组件等组成。

6) 成批查询(打印)业务交易主表记录。同样是取出已有的业务交易输出,但只会生成文档,输出电文或打印。需用预设定的条件或记录状态去选定主表业务交易记录,没有输入或输出显示画面。构件由主表读写组件、条件式执行组件、表达式执行组件、权限控制检查组件、输出显示/打印组件、文档文件生成组件、电文输出组件等组成。

7) 直接修正(非金融性)业务交易并生成历程记录。和“直接修正(非金融性)业务交易”构件有类似的功能,并且同时生成历程记录。构件由主表读写组件、取默认值组件、显示输入域组件、读取和校验输入域组件、第二次读入数据组件、条件式执行组件、表达式执行组件、权限控制检查组件、输出显示/打印组件、文档文件生成组件、文档文件编辑组件、电文输出组件、历程记录读写组件等组成。

8) 成批更新(含金融性)业务交易并生成历程记录。和“成批更新(含金融性)业务交易”构件有类似的功能。并且同时生成历程记录。构件由主表选择条件处理组件、主表读写组件、取默认值组件、第二次读入数据组件、条件式执行组件、表达式执行组件、权限控制检查组件、输出显示/打印组件、文档文件生成组件、电文输出组件、历程记录读写组件等组成。

另外,为了组装、裁剪和定制具体的可运行系统的需要,我们还一些基本组件和功能构件中设计了可设置和可调整的“状态字”和(或)参数表。

5 系统的剪裁和设置

金融系统的安全保密制度是非常严谨的,为安全性起见,负责剪裁系统的人员,只能执行剪裁的工作不参与系统的正式操作应用。他一般可由信息科技部门的主要负责人员担任。

剪裁和设置的主要工作包括:定义各功能项目、定义和生成数据结构、填写设置各参数表的参数、安全性设置等。系统为此提供了具有良好GUI界面的专用工具。操作人员只要通过GUI的可视化操作,按照系统的相关提示和引导进行,即可完成。特定系统由于业务和对系统的要求可能不尽相同,剪裁和设置工作也有所不同。下面是一些主要的工作:

1) 菜单/功能图标总控主体框架的调整和设置。菜单/功能图标总控体即是系统的总控框架,也是系统运行时的主界面。用户可以根据自己业务需求中功能项目的多寡、操作顺序和操作习惯等,对框架和界面的屏幕布局(包括界面的分层和快捷键)进行调整和设置。

2) 模块状态字的设定。为了适应业务调整的灵活性,设计了一个四位字节的状态字,用以标识一个模块中数据记录的特殊的特征信息。因为,虽然模块分割以“业务品种或信息差异较大原则”较好地解决了按相同特征进行数据划分的问题,即按这样的分割,一个模块的数据,已是一种业务或阶段数据集合,但常会有一些特殊的情况。状态字的前二字节,是用来标识需要再细分的分类;另外,数据记录由生成至结清,都有不同的阶段状态,另二字节作为状态的变化识别。这个状态字,也可用于主表、暂存表。

在有“记录状态控制组件”的功能构件中,剪

裁人员可以输入参数设定，将四位字节的状态字，分两段作为过滤记录的过滤条件，并可以用“?”作为万用字符。另一方面，亦可设定更新记录后的值。暂存表的存取，亦会用到状态字记录记录状态的变化，以实现分阶段输入数据的设定方式，但当确认更新时将以原主表记录的状态值为依据去更新状态。标识分类的两位状态字，还会和主键字编号生成挂钩，在模块编号编定参数表中，设定这个联系。

3) 批处理。这里的批处理，是指选定一批业务交易，逐一按设定的操作进行处理，在我们的半成品构件系统中，为剪裁人员提供了一个书写SQL的画面。用户可以根据自己的业务的不同在此编写SQL程序以完成相应的操作。

4) 数据传入设定。数据传入指从另一模块获取数据或从同一模块的另一个业务交易中获取数据，为完整性保护，剪裁功能项目时，要决定传入数据是否要上锁。另外，还需设定属性的对应表(mapping)。设定对应表的目的是为了数据的合并。不是传入数据记录的每一个属性值都是需要的，故对应表上，只选需要的填入，执行时组件只对这些属性进行处理。进行数据合并的方法有：a. 覆盖；b. 当前主表有值保留，否则填入，有值是指：字符不为空/空白，数值不为零；c. 传入数据有值才覆盖；d. 数值相加；e. 数值减去传入的值；f. 数值传入值减原值；g. 求较大者；h. 求较小者等。进行合并时，需要看传入的数值类型是否匹配，不匹配的不进行合并处理。

5) 默认值设置。默认值的设置和数据传入设置类似，不同的是合并至数据总线暂存区的数据是来自：参数表、常数、内建环境变量、客户数据文件的属性值、代理银行文件的属性值等。设定进行数据合并的几种方法和数据传入中所说的相同。

6) 输入画面。输入画面，是基于数据总线暂存区的域而建立，数据总线所定义的域，相应于主表在数据字典的属性定义，当定义输入画面时，就是按这些属性域来定义哪些用来作输入画面。在设定时，需要选定属性域是用来显示及输入、只供显示、只供输入，或不选。

7) 利息计算设定。利息计算设定是一项很重要的设定。利息计算一般分为预收息和后收息两种。如果是收预收息的，可设定利率表，也可不设利率表，由执行功能项目时直接输入或由其他方法

取得。选定要后收利息处理的，则必须设定利率表。后收利息处理由一独立的组件执行，并需要一套管理机制和数据结构，如计息历程记录等。

后收利息处理，需要设定利率选取各参数和取得方法。设定通过一个对应表，后收利息处理则通过这个表，读取有关的属性值。同时，亦要设定利息历程记录的数据来源和计算结果的反馈对应域。需要的参数有：客户号、业务编号、币别、本金金额、还款到期日、计息起期日、计息截止日、不计逾期罚息标志、停计息标志、累计利息、承上利息截止日、承上累计已收利息、应收利息、本次收取利息。

如果模块被定义为计收后收利息，则每一涉及利率或本金余额更改或取利息的功能项目，都应触发计息组件进行处理。因此，在设定功能项时，需要设定触发计息组件的参数。有时候，不是模块本身的功能项目会引起计息分段的因素变化，因此变成不能主动触发计息组件进行处理，解决这个问题有两种选择：a. 每晚（或每天订定一个固定时间）进行例行批处理，以便计息组件进行检查是否需要处理；b. 设定一个主动触发计息处理的功能项目，当遇计息分段的因素变化，便执行一次，这时最终用户要十分注意起计息分段的因素的变化情况。设定触发功能有下列选择，同时历程记录也有相应分类：业务交易生成、取消业务交易生成、还款、取消还款、还息、取消还息、还款付息、取消还款付息、定期计息、定期还息、余额增加、取消余额增加、利率变化、取消利率变化等。

8) 条件式和表达式。在一个功能项目中，总少不了需要一些运算。在设定时，可为每一功能项目给出一系列的条件式和运算表达式。同时为了国际贸易结算的需要，还提供了一些特别的功能函数作为扩展系统能力的手段。

9) 会计分录设定。在一个金融业务系统中，帐务处理是一个重要环节，尤其是会计分录的产生。在国际贸易结算系统中，所有金融性的功能项目，都需要有会计分录生成这一环节，并透过帐务接口，送到银行系统中的会计系统。剪裁半成品构件系统时，通过一个会计分录清单，将全部有可能出现的会计分录都填表。

在设定会计分录中的科目代号/帐号、币别、金额等各种数据来源，可以用直接设定值的方法，也可填入属性域名或中间变量由数据总线传入。执

行功能项目时,由处理组件,按当时的数据,具体生成传票套,然后传到帐务接口。所谓具体生成,除了填入各种需要的值外,还需剔除那些不需要的项,因为设定时,都是将全部可能需要的分录都填进设定清单中,真正执行时,不需要的项将得不到足够的信息而无法生成帐项。

一套会计分录是否完整,其最基本的要求是借贷平衡,即借方的帐项金额相加等于贷方金额相加。但是会计分录帐项会因为币别的差异,无法平衡,这种情况设定时,必须预先加上有关兑换的要求来解决。会计分录在处理时,要将数据转化为兑换(外汇买卖)的数据,传给兑换接口。兑换会引起外汇头寸申报问题,在直接执行、生成待确认记录、确认待确认记录的不同处理中,安排不同的申报和接收信息机制,使兑换顺利执行。

会计分录设定有分支机构独立结算的问题。即是说,一套会计分录除了最基本的借贷平衡外,当以分支机构独立结算时,各自的会计分录亦要求借贷平衡。会计分录设定的另一用途是经帐务上的数据转化为授信额度上的控制数据。会计分录设定中,涉及代理银行账户的帐项,有资金头寸问题。因应资金头寸管理系统的要求,亦需要经帐务上的数据转化为资金头盘上的控制数据,传给接口。

10) 输出/打印数据的设定。这里提供可自由设定的项目包括: a. 显示或打印文档; b. 文档的格式; c. 是否提供编辑的功能; d. 打印时所用打印机等。

一个功能项目,可以有多份不同格式的输出。首先,为各种文档建立样板格式,并给定名称。文档格式,通过专门的剪裁工具去完成建立和修订的工作。文件格式中,可以嵌入属性域名或中间变量名,只需留给足够的位置便可。

在功能项目执行时,输出组件根据文档文件格式定义,合并数据总线传入或中间变量后,形成文档。若文件提供编辑功能,则调用编辑文件组件,给用户提供一个编辑文件环境。文档格式可为个别的行或域设定为可编辑或保护。文档输出,可显示或直接输出打印,或显示后再选是否打印。在设定文档输出时,亦要填入有关打印的控制信息,供打印时选定打印机和字体字型种类。

11) 电文的设定。电文设定和打印输出设定类似,只是生成的是电文草稿或生成电文,分 TELEX 和 SWIFT。SWIFT 电文,是一种有标准

格式定义的电文,另需专门的维护去设定格式和检验方法,配合电讯系统。

12) 报表处理和工作排程。对于报表处理,考虑到目前商用报表生成器,电子表格等产品的成熟性,在总控框架中,设定一些可以嵌入其他软件产品的机制,把它作为一个外来构件嵌入到系统中,作为一个功能项目。为了达到自动执行的目的,可通过“工作排程”去触发成批处理的功能项目、报表功能项目,或其他外嵌的批量项目。所谓“工作排程”,它类似于 Windows 系统“排定的工作”选项。填写排程表是剪裁系统时的一项工作。

除上述设定外,系统还需要一些参数表去控制运作,有日历参数表、汇率表、系统控制预定值参数表、模块编号编定表、出错信息码参数表等。每个参数表,都有一套管理程序,供剪裁用户剪裁系统和进行日后的维护。

6 集成与前瞻

基于历史等原因,新一代国际贸易结算系统的设计与实现还必须考虑到与现在银行系统的集成问题和如何向分布式系统和网上银行发展的问题。采用的构件化及其相关技术也为它们的实现提供了良好的基础。整个系统的结构如图 1。

1) 主机系统 OS/390: 是原银行系统的核心,是一个以 DB2 为数据库管理系统,配以强大的联机交易处理调度系统 CICS,去实现几乎所有的银行联机业务,本系统的银行系统接口部分在这一平台上运作,采用传统的商用语言 COBOL。

2) 服务器 AIX (UNIX): 本系统的主体部分在 AIX 操作系统下,配 DB2 和应用服务器 Web Sphere,采用 JAVA 和 JSP 技术去编写网页和服务端程序,当需要向主机系统要求处理时,亦会通过 CICS-TO-CICS 方式,激活主机 CICS 的处理程序,在 AIX 平台运行 CICS,用 C 或 COBOL 语言实现。

3) 演示层 PC: 分两部分,剪裁系统及系统维护部分,用 JAVA 程序编写 GUI 接口程序去完成,在 Windows 平台上运行;应用操作,以浏览器为主体,用浏览服务器上的 JSP 网页去完成有关操作。

7 结语

本文给出了银行贸易结算系统的构件化设计与

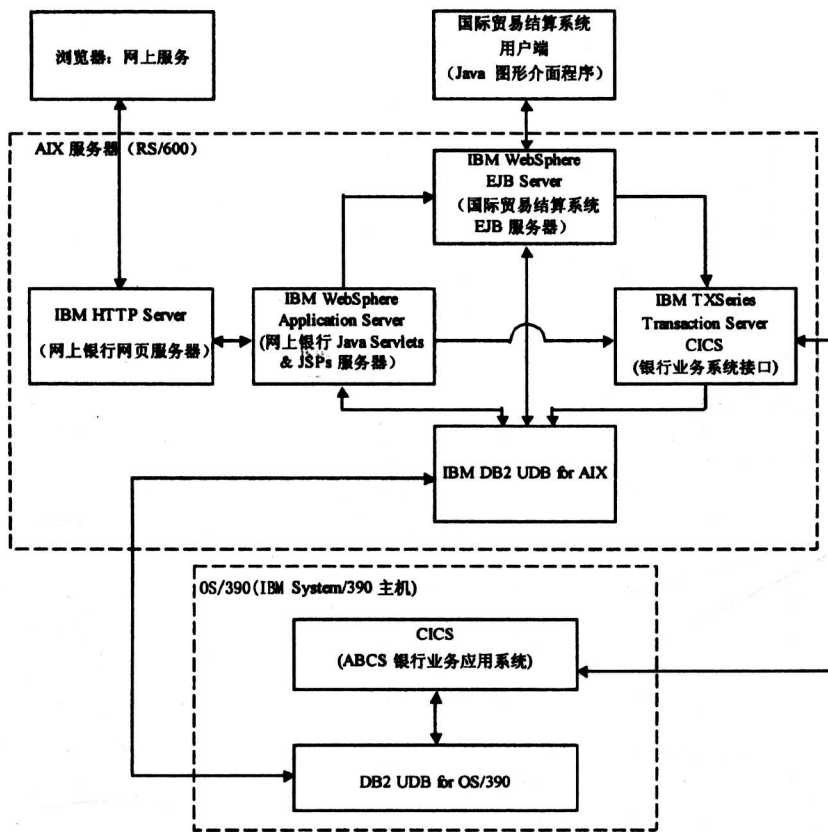


图 1 系统的部署和整体性框架

Fig.1 System deployment and overall framework

实现，它是在实际工作中根据业务需求进行的第三次转型工程。新一代系统的研制，即从开发人员的角度出发，也从应用人员的使用角度出发，结合了以往的开发实践、应用经验和当前的软件新技术。除了对领域的业务分析更加透彻，令系统的功能更加完备外，更重要的特点是采用半成品构件技术，能有效地解决因业务变化而带来的经常需要对系统进行修改的问题；应用信息工程的理论和方法以及以用户为中心的分析和设计技术，能为银行更好地发展业务打下基础。因此，虽然开发这样的系统的前期投入更大，对开发人员提出了更多的和更高的要求，但从全局和长远的观点看是完全值得的。

本文所述的系统，现阶段仍是以银行职员的操作为中心，采用的网络技术只是用于内联网上，但将来必须从两个方面去提升：1) 企业网络服务，即是将操作终端经网络引入客户“家里”；2) 电子商务企业中介服务。这些发展将会随着网络技术的发展，尤其是保密认证技术的成熟，但亦需要业务

和社会需要而配套发展。

参考文献

- [1] Ghose T K. The banking system of Hong Kong [M]. Hong Kong: Butterworths/Bank of China HK & Macau Group Training Center, 2001
- [2] 刘金宝. 互联网与银行电子化 [M]. 香港: 香港文汇出版社, 2000
- [3] 刘晓铭, 刘积仁, 李华天. 构件化领域框架设计与实现 [J]. 计算机研究与发展, 1999, 36 (2): 166~169
- [4] 何克清, 应 时, 田中茂, 等. 业务应用软件框架的一种分析方法 [J]. 软件学报, 2001, 12 (7)
- [5] Baster G, Konana P, Scoff J E. Business components: a case study of bankers trust [J]. Commu ACM, 2001, 44 (5)
- [6] Kozaczynski W. Architecture framework for business components [A]. In: Proc 5th Intl Conf Software Reuse[C]. Computer Society Press, 1998. 300~307

Review of Study of Water Mist Fire Protection Systems on Shipboard

Lu Qiang, Liao Guangxuan, Huang Xin, Wan Tianbin

(*State Key Laboratory of Fire Science, USTC, Hefei 230026, China*)

[**Abstract**] During the last decade, water mist fire protection technologies have always been the hotspot in world safety field. Being the large consumer of Halon1301 and Halon 1211 primarily, navies of most countries in the world felt more enthusiasm about it. US Navy, Royal Navy and other navies have undertaken an extensive study of water mist for fire protection applications aboard ship, especially in machinery spaces. And the results of the testing and theoretic study conducted by them are encouraging. Their study will be summarized in this paper, the purpose of this paper is twofold: to introduce their study to the fire protection workers in China and to provide a reference to China Navy in this field.

[**Key words**] water mist; pool fire; spray fire

(cont. from p.87)

Design and Implementation of Bank's International Trade Financing System Using Component Technology

Yu Jinshan¹, Chen Zhiqiang²

(1. *Department of Computer Science, Huaqiao University, Quanzhou, Fujian 362011, China*;

2. *Computer Division, Bank of China at Macau, Macau, China*)

[**Abstract**] While being an essential business in banking industry and its important role in banks getting more and more prominent with the coming of E-commerce era, the computerization of bank's international trade financing is considerably difficult. The difficulty comes from the complicated operation process, the variety, and dynamic requirement of flexibility of the business, which make it almost a mini-banking system itself. The appropriate and sophisticated software method and technology are needed to build a computer system of this kind. This paper presents a methodology based on semi-finished component model, and addresses a technique to design and implement the system. The technique is featured by building system on a standardized data bus interface, a system framework with menu/icon GUI for general controlling, and clusters of basic object components for elementary business. The basic object components can be grouped, wrapped, assembled into main frame and customized to produce a runnable system for a specific application or adapted to business changing. It also provides a flexible interface for integration with general banking systems and a sound foundation for distributed network banking systems.

[**Key words**] banking; international trade financing system; component; semi-finished system