

摘要: 给出了金融服务信息系统开发的一般过程,并以实例的方式说明了金融服务信息系统的设计与规划以及开发环境的选择,为快速开发与部署金融服务信息系统提供的指导。

关键词: 金融服务信息系统
MatLab 开发与部署

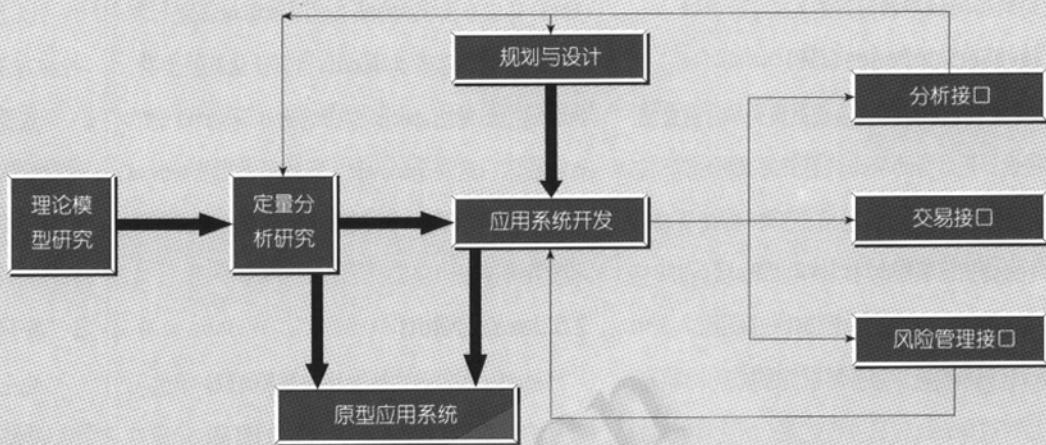


图 1 金融服务管理信息系统的开发过程

1 引言

现代金融服务公司,如银行、投资管理公司及保险公司等,为了保持竞争优势,适应环境的多变性及用户多样性的要求,必须具有能快速开发与部署为其业务服务的金融管理信息系统的能力。本文将结合具体的实例,说明在典型金融应用的系统规划、开发与部署的全过程中,如何通过选择合适的开发环境,保证金融管理信息系统的快速开发与部署。在本论文中,将主要讨论金融管理系统的项目准备与项目规划,其中主要包括:

- 金融服务管理信息系统的开发过程
- 金融服务应用系统规划过程

开发环境的选择。

对于在选定的开发环境中,金融服务信息系统的快速开发与部署将另文讨论。

2 金融服务管理信息系统的开发过程

金融服务信息系统的开发过程是一个多领域专家共同努力的过程,其中至少包括了金融理论研究专家、定量分析专家、应用系统开发人员、系统集成人员以及最终用户(包括分析人员、交易人员及普通客户等),而且还是一个带反馈的不断完善的过程。

金融理论研究专家创建用于固定收入证券、衍生证券、股票、外

汇、资产配置、资产定价及风险管理等的理论模型;而定量分析专家则用一些经验数据验证模型的正确性,对模型进行必要的修正,为模型的最后实施提供原型,并为系统的运行与测试提供必要的测试数据;而应用系统的开发人员及系统集成人员则在原型系统的基础上,通过与专家及最终用户的多次交流,直到项目最后交付用户使用。整个开发过程如图1所示。

3 金融服务应用系统规划

系统规划是金融服务信息系统开发过程中最重要的一个环节,在该环节中所作的决策决定了应用系统的内容与部署。例如,对最

终用户的使用平台是一个标准的桌面应用,还是采用客户/服务器模式,或是基于Web的浏览器应用,必然会影响到最终应用系统的开发与部署过程。

为了说明金融应用系统的规划过程,我们以建立一个最基本的用于股票期权定价的应用为实例进行说明。

3.1 定义需求

可以将该实例的要求归纳为一张用户需求表的形式,如表1所示:

3.2 系统规划

在系统需求分析表经过多次修正后,就基本上形成了对系统的主要功能、系统的运行环境及

表 1 期权定价系统的需求分析表

应用的目标是什么?	主要用于期权定价。在该实例应用中,将使用Black-Scholes 期权定价模型。
一个系统的用户及开发的用户数?	2000个交易者与分析人员,开发人员不超过20人。
用户界面的要求是什么?	图形用户界面
需要连接什么样数据?	RDBMS 及 Excel
需要什么样的报表功能?	报表要能打印、保存成外部文件及HTML发布
将要部署的平台是什么?	Windows、Unix 或 Internet
处理时间是否很关键?	需要
应用所基于的模型或分析工具是否需要经常修改?	需要
应用建设与部署时是否需要特别考虑费用因素?	需要,但需要针对性能/价格比进行综合分析。

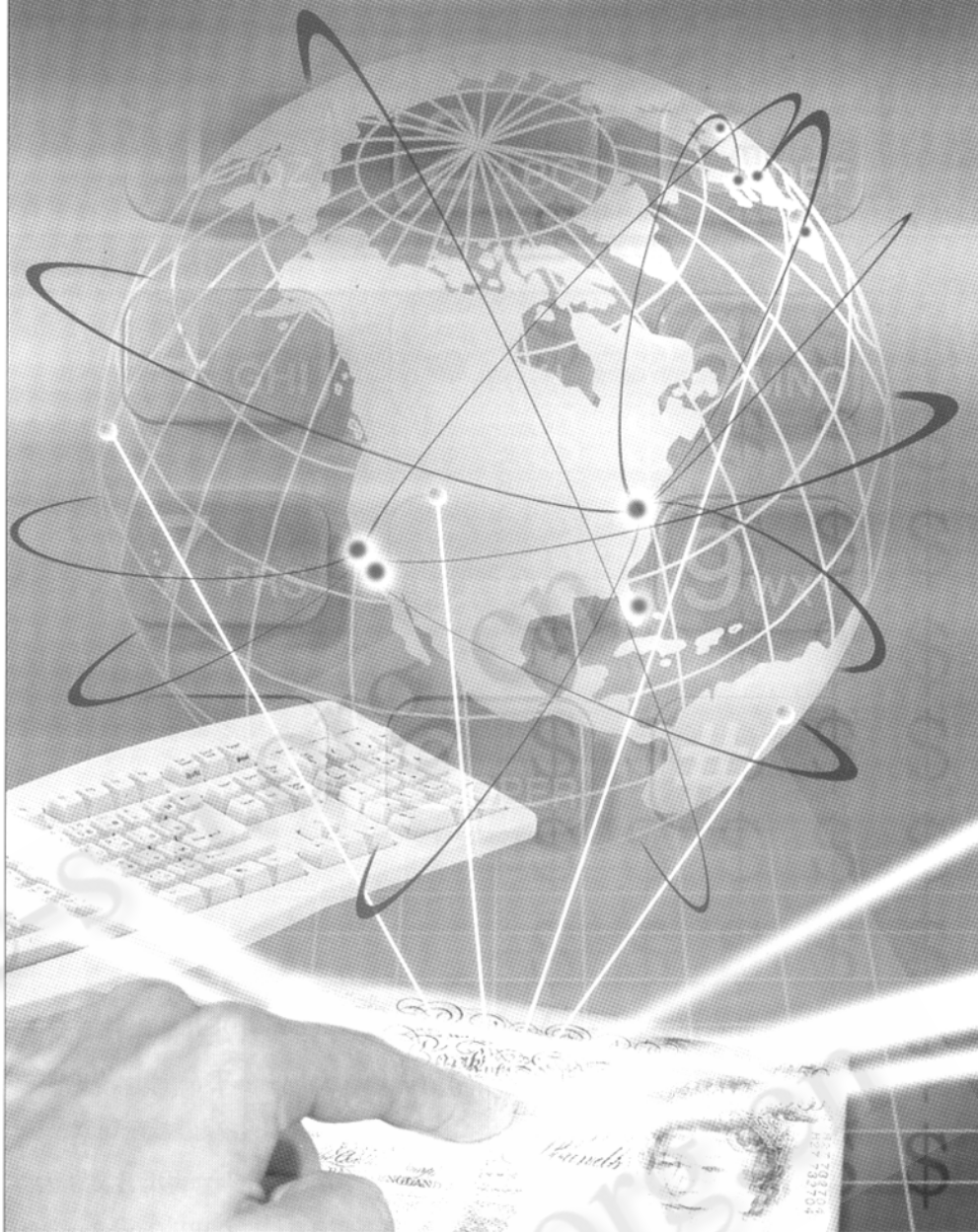
系统开发所需采用的技术的比较完整的结论。在此基础上,就可着手系统原理的分析及系统框架的设计,并最终得到完整的设计方案,这一点与一般的管理信息系统的开发没有什么区别。但在进一步详细设计与实现前,必须解决另一个重要的问题,即选择什么样的开发环境来实现金融管理信息系统所需的功能,而且实施起来方便快捷。

4 开发环境的选择

有许多集成的开发环境 IDEs (Integrated Development Environments) 可用于金融信息系统项目的开发。与每一个集成开发环境相对应,一般有一种或多种程序设计语言可供使用,如 Microsoft 的 Visual Studio 集成开发环境中包括了 Visual C/C++, Visual J++ 及 Visual Basic 等,而 Visual Cafe 提供了最稳定的 Java 开发环境, MathWorks 提供的 MatLab 则提供了一个使用 MatLab 语言开发与部署应用系统的综合环境。还有一些其他公司则以专业提供金融服务软件而闻名,如 MathSoft 的 SPlus, 它们一个仅提供给最终使用者一个操作黑箱,不便于扩展,一般仅适合于某些专用的领域,在这里,它们不属于于本文要讨论的开发环境。

为了讨论的方便,下面我们将对比现在最流行的四种集成开发环境,即 C/C++, Visual Basic, Java 与 MatLab, 而且各个集成环境的在本文中有时与它们所使用的程序语言不加区别。

Java IDE 的独立于平台、Web 部署功能,使得在开发需要既在 PC 机上运行,又能运行于 UNIX 工作站与 Internet 上的金融服务应用系统时是一个不错的选择。C/C++ 语言是应用最广泛的开发工具,而且与 Java 相比,因为是彻底的编译语言,因而一般具有较快的运行速度,而且有无数的数学分析与金融函数库可供使用。在大型金融服务项目的开发中,Java 与 C/C++ 集成开发环境是一个较好的选择,但这



两个平台也有一些缺点,金融市场需要金融应用系统的开发周期短,而且系统中所包含的模型与分析工具可能需要经常修改,因为这两个 IDE 对应的语言都是编译语言,将使系统开发与测试周期增长。

Visual Basic 是当今银行与投资管理公司使用最普遍的集成开发环境之一,这主要是因为它简单易用。一方面,金融服务项目因为需要大量的金融专业知识,所以它的开发者大多数并不是专业的 IT 人员;另一方面, Visual Basic 既是编译语言,又可解析执行,简化了系统的测试与调试过程,但是 Visual Basic 集成开发环境因为缺乏内建的数学/金融函数,而且不能直接移植到 UNIX 环境中去限制了它在大规模的需要多平台运行的金融服务应用项目中应用的可能性。

MatLab 正逐步成为开发金融服务应用项目

的受欢迎的 IDE。开发者能使用它自动产生 C/C++ 代码,能在 MatLab 程序中包含 C/C++ 及 Java 子程序,而且因为在 MatLab 中具有大量内建的金融函数,再加上它的程序设计语言既可编译执行,也可解析执行,而且独立于平台,使得基于该 IDE 可开发具有广泛适应性的金融服务信息系统。MatLab 提供的一整套适用于系统原型创建、系统开发与测试的工具,再加上基于其核心函数建立的大量金融与数学函数及数据库工具,为在该环境中快速开发复杂的金融应用系统打下了良好的基础。

因为每类 IDE 都有各自的长处与缺点,所以有时选择 IDE 也是一件非常困难的事件,针对一个特定的开发项目,为了决定最终该选择哪个 IDE,还必须回到上一节有关项目的需求分析的讨论,将需求进一步分为四大类:

· 内建的函数功能

表 2 四个 IDE 的综合比较

需求要素	Visual Basic	C/C++	Java	MatLab
基于 Black-Scholes 模型的期权定价函数	第三方提供	第三方提供	第三方提供, 而且主要需要用户自行开发	系统内建的函数功能
数据库连接性	ODBC 接口	ODBC 接口	ODBC 或 JDBC 接口	直接接口或通过 ODBC 或 JDBC 接口
工作表的连接性	通过 ActiveX	通过 ActiveX	通过 ActiveX	直接或通过 ActiveX
交叉平台配置	一般不可行	需要重新编译	需要相应的 JRE	可行
报表功能	使用第三方控件	使用第三方控件	使用第三方控件	本身具有
Web 部署	需要附加工作量	需要附加工作量	不需要	不需要
编译运行功能	有	有	有	有
用户数增大时, 作为标准应用部署时工作量	增加	增加	增加	通过使用 MatLab 的运行服务器, 可以大大减少部署的工作量
用户数增大时, 作为 Web 应用部署时工作量	变化不大	变化不大	变化不大	变化不大

- 连接数据库的能力及报表功能
- 多平台部署的能力
- 运行性能

下面将就这几个方面, 以上节讨论的简化的金融定价系统为任务, 讨论各 IDE 的特点。

4.1 开发工具中内建的函数

系统的目标是为最终用户提供基于 Black-Scholes 模型的期权定价系统, 这就首先需要各个 IDE 中使用相应的程序设计语言实现定价模型。在所讨论的四个 IDEs 中, 仅仅 MatLab 包含了模型实现的内建函数。当然, 在 C/C++ 及 Visual Basic 环境中, 可以使用第三方提供的函数库, 但是相应的 Java 函数库一般还不存在。虽然针对讨论的应用系统, 因为模型比较简单, 开发者完全可以使用 Java、C/C++ 或 Visual Basic 实现相应的模型, 但如果是大型金融服务项目, 则需要大量额外的时间。

4.2 连接数据库的能力及报表功能

C/C++ 及 Visual Basic 通过使用开放数据库连接 ODBC (Open Database Connectivity) 标准实现与数据库的连接, 而 Java 则主要通过 JDBC (Java Database Connectivity) 标准实现与数据库的连接, 而 MatLab 则不仅提供了由它本身提供的专用接口, 也可通过 ODBC 或 JDBC 连接数据库。

有多家第三方产品可用于使用 Java、C/C++

及 Visual Basic 开发的应用系统中, 为系统提供报表功能, 如 Crystal Report, 而 MatLab 则使用 MatLab Report 生成器实现报表功能。

4.3 部署能力

基于 Visual Basic IDE 开发的应用系统不能直接移植到 Unix 环境中去, 基于 Visual C/C++ IDE 开发的应用系统则需要在 Unix 平台中重新编译代码, 基于 Java IDE 开发的应用系统则需要在具有相应的 JRE (Java Runtime Environment) 环境中才能使用。因为 MatLab 是一个可跨平台使用的 IDE, 因而基于该 IDE 开发的应用系统可以直接运行于 Windows、Unix 或 Linux 环境中。

Visual C/C++ 及 Visual Basic 在进行 Web 部署时需要附加的配置; 而基于 Java 能开发能直接运行于 Web 页面的 Applet, 在进行 Web 部署时只有少量的附加工作量。MatLab 则通过 MatLab Web Server 实现 Web 部署功能, 开发者仅需要建立一些简单的 HTML 表格实现与 MatLab 应用程序交互的接口。

4.4 性能分析与评价

Visual C/C++ 与 Java 因是编译语言, 因而基于它们开发的应用系统将具有较快的运行速度; Visual Basic 与 MatLab, 因是解析执行, 相对而言运行速度会慢些。基于 Visual Basic 开发的应用系统能在 Windows 环境下编译成动态链

接库 (DLL) 及可执行文件的形式, 而 MatLab 则能在不同的平台下编译成 DLL 或标准的可执行程序, 而且可以由 MatLab 代码, 直接生成相应的 C/C++ 代码, 再使用相应的 C/C++ 编译器对这些代码进行编译。通过各种形式的编译, 可以提高基于 Visual Basic 及 MatLab IDE 开发的应用系统的执行速度。

4.5 结论

表 2 针对所要开发的应用系统的需求, 就四个平台能否达到需求进行了综合的比较, 由该表不难得出哪个平台是较好的选择。MatLab 是需要首先考虑的方案, 而 C/C++ 及 Java 也是可行的选择, 但 Visual Basic 则针对该应用系统的需求而言, 不是很合适。 ■

参考文献

- 1 薛定宇编著, 科学运算语言 MATLAB5.3 程序设计与应用, 北京清华大学出版社, 2000.10.
- 2 张志涌等编著, 精通 MatLab 5.3 版, 北京航空航天大学出版社, 2000.8.
- 3 彭江平等编著, Internet/Intranet 的开发与应用, 成都电子科技大学出版社, 1999.1.
- 4 彭江平等, 基于 FinancialCAD Developer 的 Web 金融风险管理系统, 微型机与应用, 2000.7.