Smalltalk on the Mainframe

Introduction

This paper describes IBM′s* MVS* Smalltalk technology, a Smalltalk virtual machine and a set of classes and tools that are designed to support the development of mainframe applications. This paper describes a technology, as opposed to products currently available.[1]

The paper first gives an overview of the rationale for writing Smalltalk applications for the mainframe, then describes the features of IBM Smalltalk for MVS, and finally provides an overview of the recommended development process to follow.

IBM Smalltalk for MVS technology provides:

- A pure object-oriented solution for MVS programmers.
- A standard language fully supported across client and server platforms.
- A mission-critical server environment with object-oriented services under transaction-managed systems.
- An industrial-strength development environment that includes workstation-based MVS environment simulation.

With Smalltalk on MVS, a Smalltalk programmer can create an MVS application on the workstation, test the application on the workstation in a simulated runtime environment, and then deploy the tested application on MVS.

Why MVS Smalltalk?

Smalltalk is an exceptional business programming language with a reputation for rapid, efficient development. A rich set of highly interactive development tools together with the interpretive nature of the Smalltalk language provide the developer with immediate feedback during code development. A recent IDC report indicated that ″Smalltalk is the fastest growing OO programming language market.″[2] Used on MVS, Smalltalk can help programmers create quality mainframe applications rapidly.

Smalltalk enables programmers to build and maintain custom applications that are portable and interoperable in a heterogeneous, open environment. Smalltalk makes it easy to reuse code, enabling programmers to build applications out of existing, tested components and to modify or extend applications quickly. This type of reuse reduces the overall cost of computing, improves quality, and enhances programmer productivity.

Smalltalk, like other object-oriented languages, improves the communication among end users, systems analysts, and developers by providing a common language of objects, which helps bridge the gap between the real world problem domain and the software solution domain.

Bringing Smalltalk to the MVS platform extends object-oriented technology beyond the desktop and departmental level to include enterprise-wide solutions.

---

1 This paper represents IBM′s current technology direction and is subject to change or withdrawal.

2 *Object Tools: Smalltalk Market Accelerates*, Analyst: Steve McClure, International Data Corporation, March, 1995.

Features of Smalltalk on MVS

Smalltalk provides not only an easy-to-learn and powerful language, but also a fully integrated development environment. Smalltalk offers many features that facilitate object-oriented, mainframe application development. These features are listed here:

Mainframe architecture exploitation. Smalltalk applications can exploit the inherent strengths of large systems architecture: reliability, data integrity, and security. In addition, the Smalltalk runtime environment is designed to take advantage of the strengths and features present in the MVS environment, such as the link pack area (LPA), inter-system communication, etc.

Transaction management. Smalltalk runs in a transaction manager region (either IMS/ESA* or CICS/ESA*). By utilizing existing transaction managers, corporations are able to manage large client networks with a high degree of integrity and reliability while maintaining centralized control. Installations can leverage their investment in corporate data managed by existing database managers.

Data access. Smalltalk supports database access to IMS/DB, DB2 for MVS/ESA*, and VSAM through CICS file control. Smalltalk allows programmers to access legacy data using a set of classes and runtime features. Programmers can parse legacy data into a Smalltalk class and ″wrap″ the data into an object that Smalltalk can work with.

Inter-language communication. A Smalltalk application works with other programming languages, through LE/370* services. A Smalltalk application can call, and be called by, any other LE/370-supported language application. LE/370 supports many languages, including COBOL, PL/I, and C.

Smalltalk development environment. MVS Smalltalk provides a team development environment that enables programmers to collaboratively develop and manage software components.

In the Smalltalk development environment, a programmer spends the majority of time on the workstation, developing and testing an application. The remainder of the time is spent testing and debugging the application on the mainframe.

Smalltalk portability. The Smalltalk language together with the runtime simulators provides a high degree of portability between OS/2* and MVS during the development cycle. Smalltalk′s access classes are designed to execute both on the workstation and on the mainframe without the need to change code.

Smalltalk runtime environment. The Smalltalk runtime environment consists of a Smalltalk virtual machine and one or more user applications. The Smalltalk virtual machine is the reentrant runtime engine that interprets application code during execution.

Smalltalk performance. Smalltalk on MVS is designed to keep the cost per transaction and total system load to a minimum. Test cases are being developed in both Smalltalk and traditional compiled languages, such as COBOL, to compare performance. It is expected that the system resource utilization of a Smalltalk application will not significantly exceed that required by a compiled language program.

MVS Smalltalk′s Development Process

The following process describes the major steps for writing an MVS application using IBM Smalltalk. Other steps might be necessary depending on whether the application is written for execution on CICS/ESA or IMS/ESA and whether the application uses DB2 databases.

On the workstation, the development of an IBM Smalltalk for MVS application follows these steps:

1. **Design and develop the application.** Identify the business needs that must be accommodated by an application and design the application to best meet those needs.

   If you are developing an application for execution in CICS/ESA, the following services are supported through Smalltalk access classes:

   - The application can access files using the CICS file control functions.
   - The application can access CICS management functions through interval, task, program, storage, transient data, and temporary storage controls.
   - The application can access many other services, such as diagnostic and environment services.
   - The application can manage APPC conversations.

   If you are developing an application for execution in IMS/ESA, the following services of IMS/DB and IMS/TM are supported through Smalltalk access classes:

   - The application can retrieve, replace, delete, and add segments to a database.
   - The application can retrieve input messages, send output messages to the terminal, and send messages to other applications.
   - The application can request information and access the services of either the IMS/DB or the IMS/TM.

   Additionally, your CICS or IMS application can use DB2 databases, which allow you to perform static and dynamic SQL calls.

2. **Test your application on the workstation.** You can interactively test your application on the workstation by selecting and executing the code. For IMS, the IMS Option of the Micro Focus Workbench** product simulates an IMS/ESA environment. For CICS, the CICS for OS/2* platform simulates a CICS/ESA environment.

Once your application is completely tested on the workstation, you can deploy it in the mainframe target environment. Deploying an IBM Smalltalk for MVS application on the mainframe consists of the following steps:

1. **Package the application.** Porting an application from the workstation to the mainframe involves a process called packaging. Packaging creates an image from the Smalltalk application′s classes and methods.
2. **Build a load module.** Job control language (JCL) is then run to convert the image to an executable load module using the linkage editor.
3. **Execute the application.** In CICS, you need to define and install the module you want to execute. In IMS, you can execute the module either in batch or in a message processing region (MPR), either online message processing programs (MPP), interactive Fast Path programs

(IFP), or batch message processing programs (BMP).

4. **Debug the application.** If your application experiences an error on an MVS system, you can debug it using either the interactive debugger or the batch debugger. The interactive debugger allows you to make certain changes to an application while it is executing on MVS and continue processing. The batch debugger takes a snapshot of your Smalltalk application at the time an error occurs; you can download the snapshot to the workstation and debug it there.

Note: An alternative to steps 2 and 3 can be used during the development cycle to alleviate the need to produce a load module prior to execution. In this scenario, the image can simply be uploaded to the mainframe and run.

System Requirements

IBM is currently implementing Smalltalk on MVS in the CICS and IMS transaction environments. Future considerations include the addition of native MVS batch processing support. The transaction environments have the following requirements.

Mainframe Configuration

Your mainframe system must include:

- One of the following (or both, if you are using CICS and IMS):
  - IMS/ESA 4.1 or later
  - CICS/ESA 3.3 or later
- MVS/ESA* SP 4.2 or later
- DB2 for MVS/ESA 3.1 or later, if you plan to use DB2 databases
- LE/370 1.3 with Run-time Support for IBM C/C++ for MVS/ESA plus PTF UN76759 or later
- ISPF* 3.5 or later

Workstation Configuration

Your workstation must include:

- OS/2 3.0 Warp or later
- One of the following (or both if you are using CICS and IMS):
  - To test your IMS application on the workstation, Micro Focus Workbench with the 32-Bit IMS Option (EBCDIC version)
  - To test your CICS application on the workstation, CICS for OS/2 2.0.1 or later
- DB2 for OS/2* 1.2 or later, if you plan to use DB2 databases.
- To use APPC services or the interactive debugging capability, Communication Manager/2*.

Getting More Information

For more information about IBM Smalltalk on MVS, call 1-800-IBM-2279 or 919-254-4760.

For education offerings, call:

- (US) 1-800-IBM-TEAC
- (Canada) 1-800-661-2131

This document contains information about products that are not available and describes planned technology only. This information represents IBM′s current intent, is subject to change or withdrawal, and represents only goals and objectives. User management is responsible for the choice of appropriate products and services to meet its needs.

This publication could contain technical inaccuracies or typographical errors.

IBM and Object Technology International, Inc., of Ottawa, Ontario, Canada, have entered into a strategic relationship under which portions of this technology are licensed to IBM by OTI.