Why Convert COBOL Programs To XML?

White Paper

March 15, 2006

By Avi Nudelman MSCS
Chief Technology Officer
Micro-Processor Services, Inc.
http://www.mpsinc.com
avin@mpsinc.com

Introduction

XML is a technology that allows the portability of data across platforms and between applications. The main reason to convert a program from an existing format (for instance, a COBOL program) into XML is to be able to search, index and transform these programs. The tools that currently exist to import and display XML data (for example Microsoft Internet Explorer) encode not only the program but also the data description and the hierarchal structure of the data. This description and structure data becomes very useful when converting between programming languages, as the encoding methodology allows automatic information extraction and data manipulation. Once a program is converted into XML, whether the original source was PL/I or COBOL or PASCAL, the format of XML allows the program and its structure to be analyzed more intelligently.

Why convert a COBOL program to XML?

- **Intelligent analysis of the COBOL program**

  Conversion of a COBOL program into XML format allows for automatic report generation about the business rules and data formats in the COBOL program, as well as, analysis of verbs used, DB2, SQL, and CICS statements and other code detail. In addition, XML allows for the visualization of the COBOL program using tools such as Internet Explorer, which allows a programmer to see the program's structure more easily. Finally, XML format of the COBOL program allows for automatic testing and validation for quality assurance purposes.

- **Intelligent transformation of the COBOL program**

  The XML format of the COBOL program and its SQL, DB2, CICS statements allow for easier analysis and transformation to object oriented format. In addition, XML allows CASE tools to recognize regular COBOL programs and allows for the transformation to UML for re-engineering.

- **Extract the business rules from the COBOL program**

  A legacy COBOL program normally includes business rules and other business information that is encoded within the program. In order to be able to use this information, we first need to decipher the information and present it to CASE tools in a format that is useable. This allows the use of the information and business rules in the following areas:

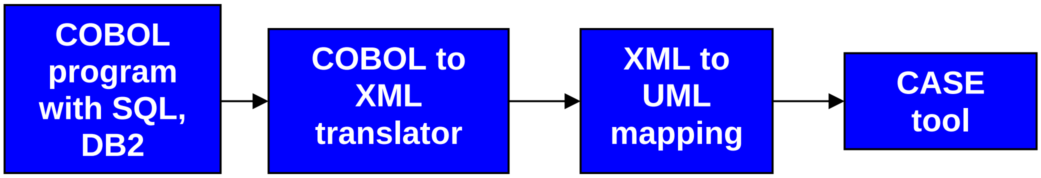
- Data division
- Procedure division
- SQL statements
- DB2 statements
- CICS statements

The data division describes the business data base. The procedure division describes the business rules that apply to the company data. The SQL, DB2, and CICS statements will describe both business rules and data. The XML format of the COBOL program and SQL, DB2, CICS statements will allow for automatic extraction of the business rules and data formats.

- **Encourage reuse of the legacy COBOL program**

  Conversion to XML allows the information embedded in the COBOL program to be used in a modern CASE tool (like **Together** by Borland). This will allow system architects to transform a legacy program to a modern program with on-line documentation and to re-engineer the program.

The Big Picture



Conclusion

Many companies have large systems that still have COBOL programs. Using the process described above, they can preserve most, if not all, of the business rules and data format from legacy undocumented COBOL programs and embedded in the SQL, DB2, CICS statements.

As we have shown, the translation of COBOL to XML can serve many purposes:

- Program analysis
- Program visualization
- Program documentation
- Program verification
- Intelligent transformations
- Extraction of business rules
- Extraction of data formats
- Re-engineering
- Reuse of legacy Cobol
- Reuse of SQL, DB2, CICS statements