



U.S. Army Develops High Quality, Extremely Low Cost Digital Message Parser

Edgar Dalrymple

U.S. Army Aviation and Missile Command

This article describes a software project managed by the U.S. Army Aviation and Missile Command, Software Engineering Directorate (SED). The project was nominated to CROSSTALK in its search for the top five software projects in the U.S. government. The project received high scores by the Software Technology Support Center review team for its description of the development of a highly automated process that the SED established to produce a digital message parser for the U.S. Army. The parser developed by the SED team, and their support contractor Science Applications International Corporation, has helped major weapon systems to demonstrate interoperability deemed critical to the future of the U.S. Army. The SED process has produced software of extremely high quality, and at the same time, extremely low cost. This article describes how these two characteristics, which may often seem mutually exclusive, were designed into the product by virtue of the development process that was established by the SED.

Battlefield information has always been an important component of war fighting. However, the U.S. Army's vision of the future is a fighting force linked as an integrated collection of interoperable systems. Loosely referred to as digitization, the addition of software-based information systems to weapon systems is the means to achieve this goal. Graphical maps populated with icons representing enemy and friendly forces are coupled with obstacles such as mine fields and air defense artillery to provide what is called situational awareness (SA) data. Satellites, aircraft, ground vehicles, and individual troops collect this data. It is distributed via a wireless computer network known as the Tactical Internet (TI). The data is used for mission planning and execution, and greatly enhances the effectiveness of every individual system, and the Army as a whole.

One of the key digitization challenges centers on the definition and implementation of a standard message protocol. Historically, different families of weapon systems have preferred different protocols, selecting them based on the needs and limitations of their individual systems. Since there was no requirement, or even any mechanism for them to interoperate, there was no reason for them to share common protocols. To obtain the desired interoperability, a standard message protocol for use by all nodes on the TI was needed.

Toward this end, a digital message protocol known as the Joint Variable Message Format (JVMF) has been selected for use by the Army. In fact, all Department of Defense (DoD) systems and our allied forces will eventually implement the protocol. The joint design

means that the protocol will be multi-service. It is the variable format nature of the protocol that adds a large amount of complexity and difficulty for the software developers who are faced with its implementation.

“The specification of the messages is nested up to six levels deep, and each level can have potentially thousands of data elements that may or may not be present.”

A variable format was selected in part to minimize input/output (I/O) loading on the very limited transmission bandwidth that is available with current Army wireless technology. Standard Army radios have been improved to transmit voice and data, but have a nominal throughput of about 9600 bits per second. This capacity is a significant bottleneck in the transmission of graphics and messages that can reach into the hundreds of megabytes in size.

Message Specifications

The specification for the JVMF message protocol is the Technical Instruction Design Plan that is maintained by the Army's Communication and Electronics Command (CECOM). The specification is effectively maintained as a database that is known as the Variable Message Format (VMF) Integrated Database (VID). The VID defines the possible

data fields and their associated parameters, structure, and the message cases and conditions. Cases and conditions are assertions about the consistency of the fields in the messages, and the parser must implement them in order to encode and decode a valid message.

CECOM produces a new database release when either new messages have been added, or existing ones have been changed. The specification of the messages is nested up to six levels deep, and each level can have potentially thousands of data elements that may or may not be present. The current version of the VID has 121 messages, with millions of fields possibly present. The information that could be contained in the full message set if all fields were populated would cause the storage size for the messages to be in the terabytes range.

By design, JVMF protocol eliminates the need to transmit empty, placeholder data packets that would be required in a fixed-format, character-based protocol. This means that only message fields that have valid user data – as signified by a bit known as the field presence indicator – get transmitted. This reduces the transmission load on the hardware. However, the complexity of the variable format places very stringent requirements on the message parsers that encode the user data into properly formatted binary data, and then subsequently decode the binary data back into user data on the receiving end. If one bit is set incorrectly, the entire message is unreadable by the recipient. Also, as the specification of the message types and field definitions is expected to continually evolve, some early studies have deemed the protocol unworkable based on the assertion that the message parsers would be too com-

plex to build, test, and maintain.

First to Face Digitization

The parser project was initially established in support of the Bradley A3 Fighting Vehicle, managed by the U.S. Army Tank and Automotive Command (TACOM). The A3 was one of the first major Army systems faced with digitization. While working with TACOM and its prime contractor, United Defense Limited Partnership, the Software Engineering Directorate (SED) anticipated and solved the problems associated with the development of the parsers by using sophisticated software engineering methodologies commonly used in compiler development.

The methodology for the development of the JVMF message parser was designed to solve the following problems:

- Produce software of exceptionally high quality.
- Maintain very short cycle times for product release.

The means by which these characteristics were achieved included the following:

- Maximum use of automation.
- Use of a Software Engineering Institute Capability Maturity Model[®] Level 3, and later Level 4, process.

The basic approach to the solution involved the following steps:

- Describe each of the messages in the VID in a formal Backus-Naur Form (BNF) grammar. This allowed for an unambiguous specification of the message structure.
- Develop automated software tools to read the VID and directly produce the BNF grammar.
- Develop automated software tools to read the BNF grammar and produce the Ada source code for the message encoder and decoder, i.e., the parser.
- Develop a test case generator to read the database and produce the suite of test cases to fully test all message fields for valid data, data ranges, and cases and conditions checks.

The SED defined a two-level BNF grammar. This allowed the entire (i.e. semantic and syntactic) behavior of the parser to be formally described. The fact that a two-level grammar was defined was essential to producing a parser that was capable of performing the cases and conditions checking. Not doing this would have made implementing the checking a manual activity. Development and maintenance of the code verifying cases and conditions by

manual means would have made achieving the quality and cycle time goals unobtainable. Failing to implement the cases and conditions checking at all would have produced an unacceptably inferior product.

To support the quality and automation goals, the team defined and developed software tools wherever possible to automate every feasible step of the process. This ensured a highly repeatable process was put in place. Once the tools were mature and debugged, it proved to be a process capable of producing extremely high levels of quality.

A brief summary of some of the tools and their functions follows. These tools are primarily written in the Ada programming language.

- **Data Extraction Tool:** Extracts the message specification data from the database supplied by CECOM to create text files used by other tools.
- **Grammar Generator Tool:** Reads the text files produced by the data extraction tool and produces the BNF grammar to represent the messages.
- **Parser Generator:** Reads the grammar generated by the grammar generator and produces the Ada source code files that represent the message parser for the messages.
- **Test Data Generator:** Generates more than 6,000 test cases to test the parser.

New tools are added as the customers' needs demand. For instance, a Symbol Table interface to the parser was created for the Program Executive Office of the Aviation Electronics Command. The symbol table parser was implemented in the Improved Data Modem (IDM). This was done to isolate software changes to the IDM, thus avoiding frequent changes to interfacing components in the avionics systems of helicopters. The flight certification concerns and associated costs due to changes to avionics software make this a very important benefit to IDM customers.

Product Performance

Currently the IDM is used in three helicopters. The ability to use the same parser software in three systems has saved an estimated \$1.2 to \$1.5 million dollars versus the cost of redeveloping similar code for each system. If used on an additional fourth aircraft as planned, an additional \$2 million dollars in savings is projected. There is also an unquantified cost savings due to the

quality of the product. Attempting to integrate unreliable, low quality software into an embedded, real-time system can have disastrous consequences on cost, schedule, and product quality. The users of this product have come to trust the SED development process to support their cost and schedule with some of the highest quality code in the Army.

The following metrics reflect the performance that has been achieved by the project. The Ada source code for the parser as counted by the terminal semicolon method consists of approximately 204 thousand lines of code. The duration required to fully generate, test, document, and deliver this code is three months from the receipt of a valid database. Engineering releases can be done in two to three days. The effort required is four person months, based on 152 person hours/person month. The post release defect density of the last three releases, as measured for six months after the commencement of operational use is 0.1 defects per 1,000 single line of code (SLOC). The current cost of the delivered product, ignoring the costs of the initial development of the tool-set, is approximately \$0.04 per SLOC.

The product is delivered with a full set of documentation, including requirements documents, test plan and procedures, application program interface specification, interface control document, version description document, Ada source code, and C language bindings. Users can compile and link only the encoder and decoder files that are needed by their application, thus avoiding the overhead of a solution that consists of a standard executable. The C language bindings enable the product to be used on virtually every hardware platform used by the Army.

As a final measure of product quality, the product is tested using the CECOM supplied VMF Test Tool (VTT). The VTT is the official test tool that provides certification of correct implementation of the message protocol. The IDM was independently tested by the CECOM labs and found to be in full compliance with the message standard.

Conclusion

This project provides a key technology to support one of the most significant organizational goals in the Army, interoperability via digitization. The increased SA data provides for a major increase in the effectiveness of Army systems. The highly automated develop-

ment process allows for extremely short cycle times in the generation and release of the product. This allows weapon systems to quickly update their ability to interoperate and provides the war fighting advantages of increased SA.

This would not be possible without the process the team has put in place to support this product. The methodology employed on this project was sophisticated and targeted to the risk associated with the application. It is a textbook example of how to craft a solution to a problem based on risk. The product speaks for itself. Hardly a week goes by that some other project fails to call and request a copy of the parser.

The fact that the technology is government owned allows for it to be distributed free of charge to DoD programs. The relatively small cost of maintaining this product is borne by the SED to the benefit of the entire Army. Custom enhancements, such as the symbol table interface, are funded by customers, notably IDM product office.

This project is an outstanding example of how the Army Life Cycle Software Engineering Centers can be used to benefit a wide range of Army systems in a cost effective and high

quality manner.

Key Players

The most important part of any project is its members. The author would like to specifically mention the following team members who have been key to this program:

- James Magnusson, Science Applications International Corporation (SAIC) assembled the technical team.
- Doris Chan, SAIC, has served as the parser technical lead for six years.
- John Shannon, SAIC, developed and defined the methodology to employ the formal BNF grammar.

All of these people have more than 20 years experience in the design, development, and management of embedded software systems for military applications:

- Larry Stanbery, SAIC, enhanced the automation of the testing process and has more than 10 years of software experience.
- Charles Hyder, SAIC, provided software quality assurance support.
- Debra Henry, EER Systems, provided configuration management support. ♦

About the Author



Edgar Dalrymple is a lead project engineer and division manager at the U.S. Army's Aviation and Missile Command, Software Engineering Directorate. For 15 years, he has supported major Army systems in the areas of software quality assurance, independent verification and validation, and software project management. In addition to the Joint Variable Message Format message parser, he currently supports the Bradley A3 modernization. He received a bachelor's degree in nuclear engineering from North Carolina State University in 1979.

**U.S. Army Avionics
and Missile Command
AMSAM-RD-BA-SQ
Redstone Arsenal, AL 35898
Phone: (256) 842-0187
E-mail: [edgar.dalrymple@
sed.redstone.army.mil](mailto:edgar.dalrymple@sed.redstone.army.mil)**

WEB SITES

Note: Newcomers to software process improvement or the Capability Maturity Model® IntegrationSM will benefit from these Web sites that offer information and training on all aspects of software process improvement.

Software Technology Support Center

www.stsc.hill.af.mil

The Software Technology Support Center (STSC) is an Air Force organization established to help other U.S. government organizations identify, evaluate, and adopt technologies to improve the quality of their software products, efficiency in producing them, and their ability to accurately predict the cost and schedule of their delivery. The STSC is offering a free workshop to U.S. government employees on Capability Maturity Model Integration Version 1.1 on May 21-23, 2002.

Software Technology Conference

www.stc-online.org

The Software Technology Conference (STC) is the Department of Defense's premier software conference. One tutorial and one track will be specifically aimed at addressing the Capability Maturity Model Integration Version 1.1. The CMMI tutorial can be found on Monday, April 29, 2002 Track 4 at 1:00 p.m.; the CMMI track can be found on Wednesday, May 1, 2002 Track 4 at 1:00 p.m.

Software Engineering Institute

www.sei.cmu.edu

The Software Engineering Institute (SEI) is a federally funded research and development center sponsored by the Department of Defense to provide leadership in advancing the state of the practice of software engineering to improve the quality of systems that depend on software. SEI helps organizations and individuals to improve their software engineering management practices. The site features complete information on models it is currently involved in developing, expanding, or maintaining, including the Capability Maturity Model Integration, Capability Maturity Model for Software, Software Acquisition Capability Maturity Model, Systems Engineering Capability Maturity Model, and more.

Software Productivity Consortium

www.software.org

The Software Productivity Consortium (SPC) is a nonprofit partnership of industry, government, and academia. The SPC develops processes, methods, tools, and supporting services to help members and affiliates build high-quality, component-based systems, and continuously advance their systems and software engineering maturity pursuant to the guidelines of all of the major process and quality frameworks. Membership is open to all U.S. or Canadian-based companies, government agencies, and academic organizations.