

A WAVEFORM DESCRIPTION LANGUAGE FOR SOFTWARE DEFINED RADIO

Michael Gudaitis (L-3 Communications Analytics Corporation, Rome, NY, USA,
mike.gudaitis@L-3com.com)

ABSTRACT

Software Defined Radios shift the design emphasis of waveforms from hardware to predominantly software. Translating written waveform specifications to software source code is error prone and costly because design errors are undiscovered until the final stages of development when software is integrated and tested. To reduce errors, increase understandability, and simplify integration and testing, a Waveform Description Language (WDL) is needed to unambiguously describe waveform requirements, capture design information, and aid engineers to determine at the system level whether capabilities are best-suited to being implemented in hardware or software. A WDL would be used to describe complicated software and hardware design in graphical notation to quickly convey system concepts and relationships. A WDL would capture and specify the radio waveform information in a form that could be interpreted to convey the design visually, simulated to test the design functionally, and compiled to implement the design correctly on target hardware. This paper presents a WDL approach based on a combined subset of UML and MATLAB using the Future Multiband Multiwaveform Modular Tactical Radio (FM3TR) waveform as an example. There are various tools that support UML or MATLAB, but not both. This paper will also discuss ongoing research for automated conversion between UML and MATLAB to bridge the gap between these two languages for use as a WDL.

1. INTRODUCTION

As adopted by the SDR Forum [1], the term software defined radios (SDRs) is used to describe radios that provide software control of radio functions and waveform requirements. In other words, the radio waveform is defined by the software that controls it. SDRs shift the design emphasis of waveforms from hardware to predominantly software. Designing and implementing waveforms in software can be a difficult process. There is no systematic standardized process for translating written

waveform specifications to software source code. A WDL is needed to standardize waveform specifications to reduce design errors and ambiguity, improve clarity and understandability, enhance interoperability, and thereby decrease development costs and schedules. A WDL would be used to describe complicated software and hardware design in graphical notation to quickly convey system concepts and relationships. The WDL concept is further described in [2], [3], and [4]. This paper describes a WDL concept based on the Unified Modeling Language (UML) from the Object Management Group (OMG) [5], and MATLAB/Simulink from Mathworks [6], with eXtensible Markup Language (XML) [7] dialects for an intermediate data exchange format. The Future Multiband Multiwaveform Modular Tactical Radio (FM3TR) waveform [8] is used as an example to illustrate the WDL concept. MATLAB tools from Mathworks do not support UML or XML, so the conversion of model information to/from UML and MATLAB must be done manually. This paper will also discuss ongoing research for automated conversion between UML and MATLAB to bridge the gap between these two languages for use as a WDL.

2. BACKGROUND

The WDL concept is a model-based approach employing graphical models which allow developers to use pictures in standardized notation to convey complex relationships. The increased complexities of systems demand a graphical, unambiguous, and intuitive approach to waveform design. The goal of a WDL is to standardize the waveform pictures with a formal graphical and textual syntax to provide reusable specifications for waveform development. A WDL would employ the appropriate subset and mix of standard languages for intuitive and unambiguous waveform specification and design. A WDL will serve three objectives:

1. To describe waveforms using largely visual or graphical models/diagrams.
2. To test a design, either by applying formal constraints or by using simulation.
3. To assist in the implementation of design through automatic code generation.

UML was selected for this WDL concept because it is used extensively in the Joint Tactical Radio System (JTRS) Software Communications Architecture (SCA) [9]. It is also the most widely used software modeling language, and it is the basis of the OMG's Model Driven Architecture (MDA) [10]. UML has eight types of diagrams to describe software architecture, structure, and behavior, but it does not have features to model mathematical algorithms. For algorithm development, MATLAB is a better choice. MATLAB is the most popular math language and tool used to model mathematical algorithms such as those that describe the functions of a communications waveform. Simulink extends the basic capabilities of MATLAB by adding a block diagram tool and simulation capabilities. The UK Programmable Digital Radio (PDR) program [3] has selected MATLAB as its WDL. However, MATLAB doesn't model software system architecture very well. Therefore, to describe the software architecture and the algorithms of communications waveforms, the WDL concept proposed in this paper uses both UML and MATLAB to present a more complete model than could be obtained by either language individually.

In addition to UML and MATLAB, several dialects of XML are also needed for standardizing the file format and data interchange between tools. XML Metadata Interchange (XMI) is the OMG standard for exchanging UML diagram information between tools [11]. Math Markup Language (MathML) is a standard for format and content of mathematical equations [12]. Timing Diagram Markup Language (TDML) is a standard for timing diagrams [13]. Together, these languages form the basic components of a WDL as shown in Figure 1.

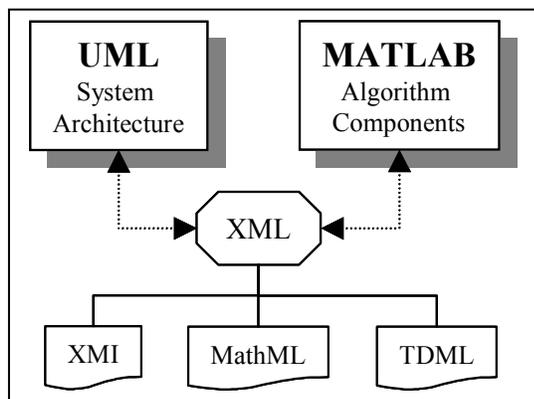


Figure 1. Language Components of Proposed WDL

The dashed line represents a conversion from the modeling languages (UML, MATLAB) to/from XML. The model information is saved in XML format by applying a set of standard rules associated with the XML dialect (XMI, MathML, TDML). The XML provides a standard intermediate representation that an interpreter

can use to build the waveform model. At this time, automated XML conversion is only partially supported by various tools. For example, many UML tools support XMI, but they do not support MathML or TDML. MATLAB doesn't support any of the XML dialects shown, but uses its own textual file format. However, there is ongoing research to bridge the translation gap between UML and MATLAB. In particular, Vanderbilt University has developed a tool to convert Visio UML diagrams to XML, and convert MATLAB/Simulink information to XML format [14]. This is described in the next section.

2. VANDERBILT UDM MATLAB TOOL

Vanderbilt University created a tool that uses XML as an intermediate representation of model information from MATLAB and UML models. The research was funded by DARPA under the Model Based Integration of Embedded Systems (MoBIES) program [16]. Vanderbilt created a metamodel of the data used in Simulink files. This MATLAB metamodel forms the basis of their UDM tool [17], which extracts model information from Simulink files and Visio UML files. This tool is useful as a starting point for creating an XML translator to/from MATLAB and UML. However, in its current form, the UDM tool doesn't follow the syntax of the XML dialects shown in Figure 1. Incorporating the syntax rules for the XML dialects of the proposed WDL is a subject of further research.

3. FM3TR

The FM3TR waveform specification [8] is typical of waveform specifications, and it includes informal textual descriptions and figures, mathematical equations, and timing diagrams. These can all be represented more formally using the standard language components of the WDL proposed herein. The remainder of this paper will focus on the MATLAB model for FM3TR.

3.1. Timing Diagram

FM3TR is a frequency hopping waveform, and the FM3TR specification contains timing diagrams for describing the timing for frequency hopping. The hop representation for the TW1a Waveform is shown in Figure 2. The timing diagram follows the TDML style. A Graphical User Interface (GUI) was developed using the MATLAB tool to easily create timing diagrams based on FM3TR timing parameters. N_{rt} is the rise and release time of the hop. N_{inf} is the dwell time for valid information symbols. N_{tt} is the tuning time between hops. N_{dot} is the dwell off time; $N_{dot} = 2N_{rt} + N_{tt}$,

i.e., the time between the last valid information symbol of the previous hop, and the first valid information symbol of the next hop. The MATLAB GUI captures the timing parameters for use in the FM3TR waveform model simulator described in the next section.

3.2. FM3TR Simulator

A simulation of the FM3TR waveform was developed using Simulink. Rather than starting from scratch, a Simulink model of the Bluetooth HV3 waveform was used as a starting point [15], [16]. This demonstrates the concept of model reuse for waveform development. The Bluetooth HV3 waveform is quite similar to FM3TR. Both employ frequency hopping, a Continuous Phase Frequency Shift Keying (CPFSK) modem, and a Continuously Variable Slope Delta (CVSD) audio codec. The primary difference between the two waveforms is the hop framing and synchronization. The Simulink model for the FM3TR transmitter is shown in Figure 3. This model was derived from the Simulink model for the Bluetooth HV3 waveform as shown in Figure 4.

As can be seen in the two figures, the basic structure of each waveform is quite similar. In the CVSD block of the Bluetooth model, the sample rate is reduced from 64 kbps to obtain the 16 kbps rate of the FM3TR waveform. In the modem block, the Gaussian filter is removed, the carrier frequency and hop rate is changed, and the channel bandwidth is reduced from 1 MHz to 25 KHz. The framing and synchronization structure is the only section of the model that requires a major redesign to account for the differences between the two waveforms. Bluetooth transmits one frame per hop in a time division duplex (TDD) protocol over timed channel slots. FM3TR transmits one frame per five hops using a conventional Push to Talk half duplex protocol. The Simulink tool provides the flexibility to experiment with different approaches for modeling the waveform framing and synchronization. This allows the waveform designer to quickly iterate and test several approaches in order to refine the design. The Simulink model then becomes the executable specification. The goal of the WDL is to create a standard approach and style guide for the naming conventions and graphics used in the Simulink model to facilitate automatic translation to/from UML using the XML dialects shown in Figure 1.

At present, there is additional work to be accomplished in order to integrate the three XML dialects into one tool that could translate between MATLAB and UML. Each of the XML dialects have their own tool support, and work is ongoing to merge this capability into a Waveform Development Environment that incorporates UML and MATLAB capabilities.

4. SUMMARY

This paper presents a WDL concept based on UML and MATLAB/Simulink to support waveform development for software defined radios (SDR). A WDL is needed to standardize waveform specifications to reduce design errors and ambiguity, enhance interoperability, and thereby decrease development costs and schedules. The WDL concept was illustrated using the FM3TR waveform as an example. Additional work is needed to integrate the various language components of the proposed WDL into a common Waveform Development Environment.

5. REFERENCES

- [1] "SDR Primer", Software Defined Radio Forum web site, http://www.sdrforum.org/sdr_primer.html.
- [2] M.S. Gudaitis, and R.D.Hinman, "Practical Considerations for a Waveform Development Environment," MILCOM 2001, IEEE, October 2001.
- [3] B. Prill, and M. Harrington, "The UK's Programmable Digital Radio Project" SMI's Second Annual Software Radio Conference. 17-18 June 2002, <http://www.smi-online.co.uk/media/docs/f91.pdf>
- [4] E. Willink's web page on Waveform Description Language, <http://www.computing.surrey.ac.uk/personal/pg/E.Willink/wdl/wdl.html>, accessed September 9, 2002.
- [5] Object Management Group (OMG), <http://www.omg.org>.
- [6] Mathworks, www.mathworks.com/products/simulink/.
- [7] XML Web site, <http://www.w3.org/XML/>
- [8] R. Smith, "Description of the FM3TR Proposed Reference Waveform," SDRF-01-I-0056-V0.00, 30 August 2001 http://www.sdrforum.org/MTGS/mtg_25_sep01/01_i_0056_v0_00_fm3tr_97_09_10_01.pdf
- [9] Software Communication Architecture (SCA) Specification with Appendices, MSRC-5000SCA V2.2, 17 November 2001, available at the Joint Tactical Radio System (JTRS) web site, <http://www.jtrs.saalt.army.mil/>
- [10] OMG's Model Driven Architecture, OMG document ormsc/2001-07-01. <http://www.omg.org/mda/>
- [11] XML Metadata Interchange (XMI) specification, <http://www.omg.org/technology/documents/formal/xmi.htm>
- [12] MathML web site <http://www.w3.org/TR/MathML2/>
- [13] TDML web site, <http://www.si2.org/tdml/>
- [14] Vanderbilt MATLAB UDM tool web site, <http://www.isis.vanderbilt.edu/Projects/mobies/MatlabUDM.html>
- [15] Bluetooth Simulink Model from MATLAB Central, <http://www.mathworks.com/matlabcentral/fileexchange/Files.jsp?type=category&id=&fileId=2283>
- [16] Specification of the Bluetooth System, Version 1.1, February 22, 2001, available from Bluetooth SIG, <http://www.bluetooth.com>.

This work is supported in part by Contract F30602-01-C-0205, Air Force Research Laboratory, Department of the Air Force. Points of view in this document are those of the author and do not necessarily represent the official position of the U.S. Department of the Air Force.

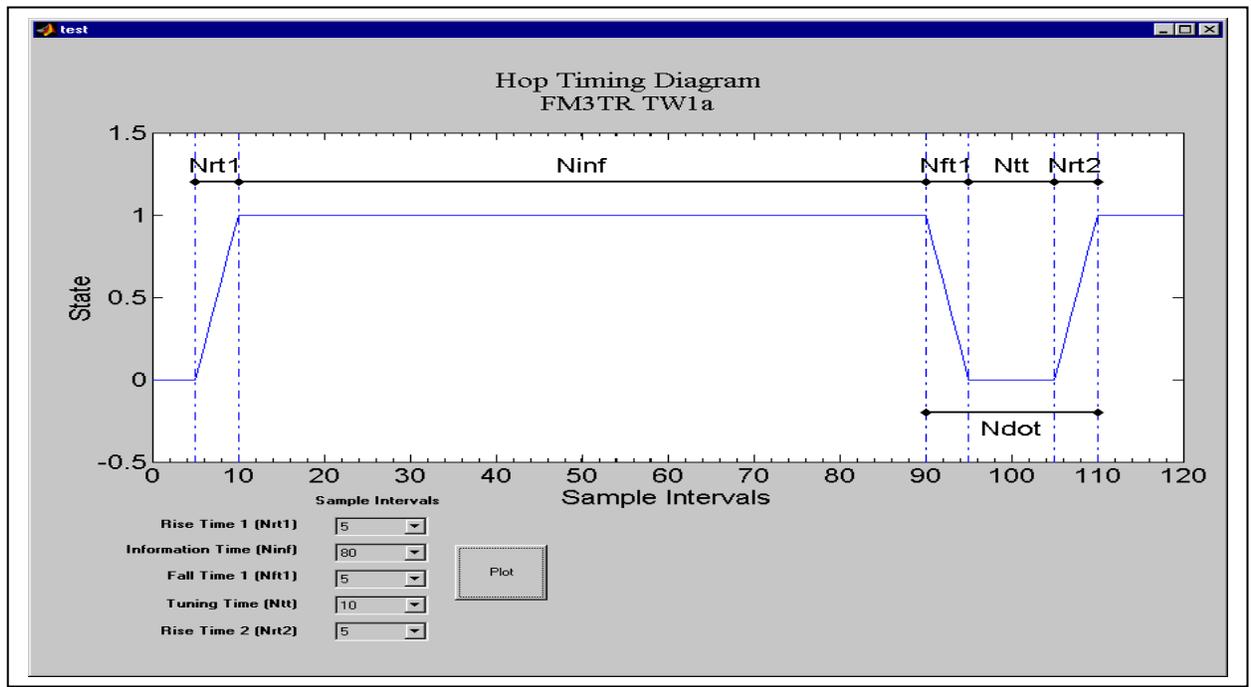


Figure 2. FM3TR Hop Timing Diagram drawn with MATLAB Tool.

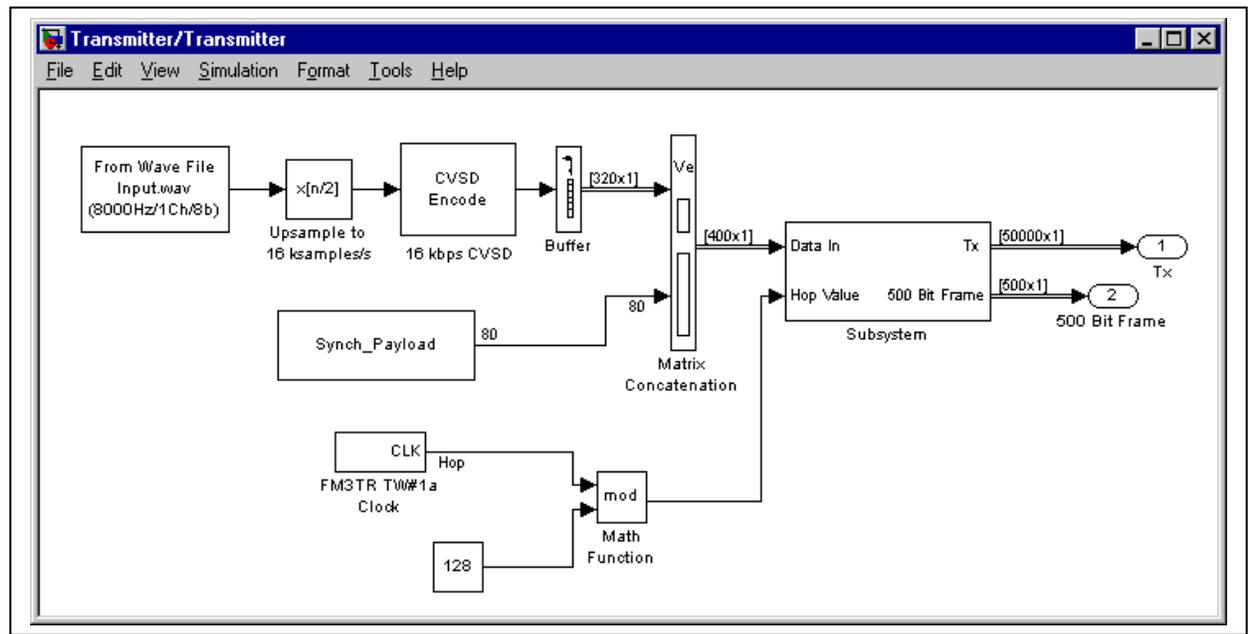


Figure 3. Simulink Model of the Transmitter for the FM3TR TW1a Waveform.

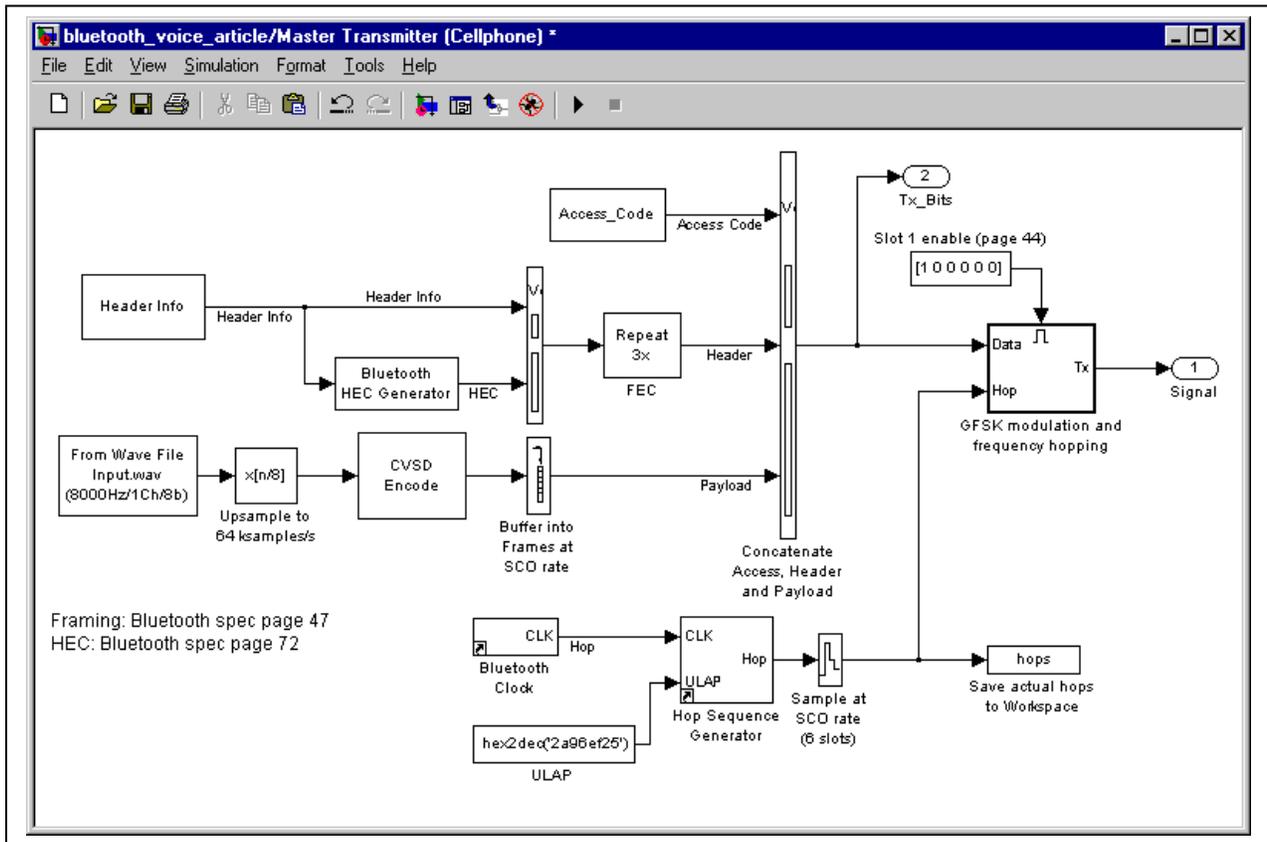


Figure 4. Simulink Model of Bluetooth Transmitter (from [16]).