PHYSICAL LAYER DESCRIPTION STANDARD FOR SOFTWARE DEFINED RADIO

Eugene Grayver
(The Aerospace Corporation, Los Angeles, CA; eugene.grayver@aero.org)

ABSTRACT

The main advantage of Software Defined Radio (SDR) is the ability to implement new waveforms without changing the hardware. The flexibility offered by SDR hardware is useless unless a designer can easily take advantage of it. Unfortunately, many of the currently-fielded SDR platforms use proprietary interfaces and have a high barrier to entry for 3rd party developers. Indeed, despite the programmability of the radios, the customer is often locked into the vendor for developing techniques and software for the radio. In this paper we define a simple standard for describing the physical layer of the waveforms implemented on an SDR. The Extensible Markup Language (XML) has been selected to represent the standard. The core of the standard has now been defined, covering hundreds of aspects from modulation to coding and framing. An interactive website has been created to allow multiple contributors to add and modify the standard and to allow collaboration with colleagues in industry and academia. This paper describes the philosophy behind the standard, introduces the XML notation and paradigms for both configuring a radio and querying its status and capabilities. The main goal of the paper is to solicit input from interested parties and to foster the acceptance of this open and free standard.

1. INTRODUCTION

Flexibility and complexity often go hand in hand. No manual is required to operate an old AM radio. An AM/FM radio is slightly more complex and a manual may be needed to locate the switch that selects the mode. Modern software defined radios can have hundreds of switches or an infinite number of different modes. The modes may be selected by the end user or by an automatic agent in a cognitive radio. The scenario that prompted the work reported in this paper involved a software defined basestation that had to communicate to dozens of different radios over the course of a day. Each of those radios would transmit/communicate at different times and use unique waveforms. The logistics and overhead of manually configuring the basestation for each radio became prohibitive. All the radios used the same network protocols and transmitted simple voice data. Only the physical layers were different. A standard way of describing the physical layer was therefore needed. Consider the simple task of specifying the signal carrier frequency – some refer to this parameter as f_c, others as *carrier*, and others as *freq*.

A common language for describing waveforms for wireless communication can:

- Facilitate reuse of software developed for different hardware platforms
- Reduce the cost and time to move between different vendors' solutions
- Stimulate research in software defined and cognitive radios by simplifying collaboration between different teams
- Reduce procurement costs by encouraging competition between vendors
- Help students and researchers see the commonalities between different waveforms rather than focusing on the unique details

Standardization of SDR has been proceeding for over 10 years. The US Army developed a very robust and capable standard – JTRS [1] – to get a handle on the capabilities of SDRs. NASA spearheaded an effort to standardize the SDRs used for deep space missions (STRS [2]). However, both of these efforts focus on the higher network layers and treat the physical layer as a 'black box.' Different research groups at universities such as Virginia Tech and Berkeley developed in-house standards to work with their SDR hardware [3]. Excellent progress is being made in standardizing the hardware interfaces to the analog and RF front ends [7], providing the lowest API layer. The standard proposed in this paper is neither unique nor groundbreaking, but it fills a critical need for a common open standard that is free of nondisclosure agreements (NDA) and patent protection.

The main goal of the paper is to solicit input from interested parties and to foster the acceptance of this open and free standard. The scope of the standard, hereafter known as SDRPHY, is limited – describing the physical layer properties of most waveforms. The physical layer covers just enough of the radio for the receiver to be able to

- acquire the signal
- synchronize to the frame structure
- demodulate, decode and decrypt the bits.

The hard decisions are then passed to the higher layers which are beyond the scope of SDRPHY. Thus, for example, modulation is covered, but packet verification based on cyclic redundancy check (CRC) is not.

Conceptually, SDRPHY can be used as follows:

- A description of the waveform based on SDRPHY is passed to an *interpreter*. The interpreter software is developed for a specific SDR implementation.
- It converts the description into a set of configuration commands, or creates a flowgraph, or even writes code to implement a radio satisfying the description. The interpreter may reside within the SDR hardware, or operate entirely off-line.

2. STANDARD PHILOSOPHY

The goal of this standard is to fill in the gap by providing a light-weight description language that covers the samples-to-bits part of the SDR. It is not meant to replace standards such as JTRS and STRS, but to augment them. The transmitter functionality starts with data bits and ends at the antenna, while the receiver starts at the antenna and ends at the decoded data bits. So far, we have only mentioned the problem of configuring a radio. However, just because a radio is software defined does not mean that it can support any waveform. The other part of the standard must therefore deal with describing the capabilities of a given radio.

The standard should ideally satisfy the following goals

- **Completeness**. Most practical waveforms should be describable using the standard. The goal is for 99% of all waveforms of interest to the user community to be supported. The remaining 1% of exotic waveforms would make the standard too complex.
- **Lightweight**. The interpreters for the standard are meant to be implemented in a wide range of devices – from satellites to handheld units. Therefore, low processing and memory requirements are desirable.
- **Consistency**. The standard should be self-consistent. Similar functionality should not require different descriptions. A high-level, abstract view of the waveform is adapted whenever possible. Conceptual commonalities should be exploited.
- **Compactness**. The same keywords should be used for describing the configuration and capabilities, whenever possible. In particular, no new keywords should be created if existing ones can be repurposed.
- **Accessibility**. There should be no barriers to entry for users wishing to participate in the creation or utilization of this standard.

The goal of this standard is to enable application developers to work with a wide range of practical flexible radios. The idea of a practical radio limits the scope of the standard; it will not attempt to describe an arbitrary radio. Some combinations of features may be physically possible, but are either not practical or highly unusual. For example:

> A waveform may use either convolutional (Viterbi) or LDPC forward error correction. It is possible to concatenate Viterbi and LDPC, but such a combination makes little sense from the communications architecture perspective (both are soft decoded). Likewise, it is possible to apply forward error correction to the chips in a spread-spectrum system instead of applying forward error correction (FEC) to the symbols. However, no practical radio uses this technique.

The standard can therefore rely on a large set of implied constraints, based on a canonical communications system. These constraints will be spelled out whenever appropriate.

The consistency and compactness goals are currently being debated. There is a trade-off between using the most abstract view of a parameter and ease of implementation and specification. Consider a Reed-Solomon (RS) code – it can be shown that RS is a special case of a BCH code [4]. Therefore, it is sufficient to define a keyword for BCH. On the other hand, many users of the standard may not be aware of the equivalency of RS and BCH, placing an undue burden on them. The current approach is to provide Reed-Solomon as a library entity, based on BCH. Similar trade-offs come up in many other cases.

The standard is processed by a software program called an *interpreter*. The interpreter can be considered an expert system (a PhD in a box) with extensive domain expertise in the field of communications. The implied constraints and the concept of a canonical system are an integral part of the interpreter. The expert system shifts some of the complexity of specifying a waveform from the application designer to the interpreter. A designer should be able to construct a description to configure a radio by reading a published standard.

There are at least two ways to describe a waveform:

- Defining constituent blocks and interconnections between them
- Describing the waveform by specifying the values for the different ‘knobs’

Neither approach is fully satisfactory. The block-based description comes close to specifying the implementation rather than the intent. Once the block-based paradigm is allowed, there's no logical reason why it cannot be extended to specifying every adder and multiplier in the system. Furthermore, it is not obvious how to go from the description of a waveform to a receiver that can process that waveform.

Describing the waveform by simply specifying the values for all variables (parameters) is problematic due to possible ambiguities. For example if an FEC encoder and an interleaver are both described, it is not clear which comes first.

This standard is based on the second approach of/to describing a waveform. Most of the interconnect is specified by the canonical communications system[1]. Note that some ambiguities remain and will have to be resolved as the standard matures.

> Consider a differential PSK modulation. It can be described as either:
>
> - an encoder followed by a memory-less linear modulation or
> - no encoder, followed by a modulator with each output symbol defined as a function of the current and *past* symbols.
>
> The two descriptions are equally valid, but it is very difficult to develop an interpreter that can identify the equivalency of the descriptions.

Unique aspects of some systems cannot be adequately described using the keywords defined in this standard. For example, a spreading sequence for a direct sequence spread spectrum system can be based on a proprietary cryptographic generator (e.g., GPS military codes). Describing such a generator is well beyond the scope of this standard. A foreign attribute is therefore defined to allow the user to supply a non-standard component. The foreign attribute is very problematic since it must be supported across different platforms – from software to hardware. The component interface must therefore be very well defined. The interface definition is beyond the scope of this document and will probably be based on existing industry standards (e.g., CORBA for software and OCP for hardware).

3. IMPLEMENTATION ISSUES

SDRPHY is built on top of XML. XML has a number of advantages, the main ones being widespread acceptance, availability of parsers and tools, and human readability. A radio configuration is encoded in XML, while radio capabilities are reported as an XML schema (XSD). The schema provides a standard mechanism to define parameter ranges and sets of options.

XML does come at a price since it is rather verbose, and therefore, requires more resources to generate, store, and parse than a custom binary format. However, this overhead is negligible, given the complexity of today's SDRs[2].

This paper makes no attempt to describe XML itself and does not necessarily adhere to XML 'best practices.' The final version of the standard will declare an XML namespace, *sdrphy*, to avoid name conflicts with other XML frameworks.

An XML description consists of a hierarchical tree of nodes. Each node represents a conceptual part of a waveform. The hierarchy order is not always obvious. For example:

> Does a radio consist of multiple transmitters and receivers, or does each transmitter and receiver consist of multiple channels?

In general, node B is a child of node A if it satisfies the following question: Is B a property of A?

The XML definitions of different modules (e.g., coding, modulation, etc.) can come from different vendors or designers. SDRPHY provides explicit support for libraries. A library contains one or more XML descriptions of a module, possibly using other libraries. A library is referenced using the *xlink:href* syntax:

```
<modulation xlink:href=
  "vendor_x_library.xml?custom_mod_1">
</modulation>
```

The file 'vendor_x_library.xml' contains[3] the definition of 'custom_mod_1.' For example:

```
<vendor_x_library>
  <modulation ID=”custom_mod_1”>
    <family>cpm</family>
    <constellation>
      <points>
        <set><e>-1</e><e>1</e></set>
```

---

[1] Connectivity can be specified when absolutely necessary. Any constituent block described in XML can be uniquely identified with an ID attribute. Connectivity is specified by adding a <input ID="unique_id"> node to the block following the uniquely identified block. This construct should be used sparingly.

[2] The XML text may be 'compiled' into a compact binary representation specific to a hardware platform [5].

[3] Note that the root node of the library file should be the same as the filename without the '.xml' extension.

```
      </points>
    </constellation>
  </modulation>
</vendor_x_library>
```

An included node can be modified by overriding some of the subnodes. For example:

```
<modulation xlink:href=
"vendor_x_library.xml?custom_mod_1">
  <constellation>
    <rotate>3.14</rotate>
  </constellation>
</modulation>
```

is translated to:

```
<modulation ID=”custom_mod_1”>
  <family>cpm</family>
  <constellation>
    <points>
      <set><e>-1</e><e>1</e></set>
    </points>
    <rotate>3.14</rotate>
  </constellation>
</modulation>
```

A set of standard libraries (similar to the standard functions that are a part of the ‘C’ language) will be defined. These libraries can be used to create much more compact XML descriptions for common waveforms. For example, canonical Gray-coded constellations such as QPSK and 16-QAM will be available in the standard library. A radio may be queried to list all the libraries that it already has stored. Some radios may be able to save the uploaded libraries so that they do not need to be re-sent with every XML file.

The order of XML tags is used to implicitly describe connectivity if different interpretations are possible. For example, a transmitter with an interleaver followed by a Reed-Solomon (special case of BCH) block is described as

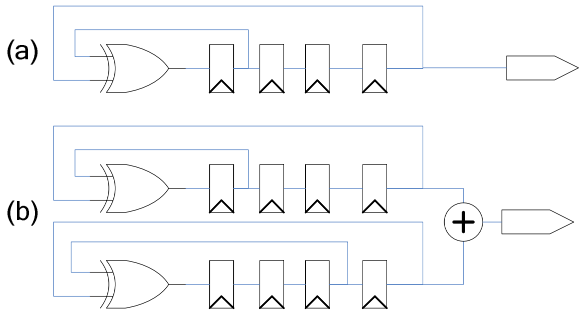
```
<coding>
  <interleaver> . . .</interleaver>
  <BCH>. . . </BCH>
</coding>
```

while a transmitter with a convolutional encoder followed by an interleaver is described as

```
<coding>
  <convolutional>. . .</convolutional>
  <interleaver> . . .</interleaver>
</coding>
```

This interpretation of XML is specified in the standard at the level of <coding>.

For a different example of the interpretation of XML, we look at specifying a pseudo-noise sequence based on a linear feedback shift register (LFSR). The outputs of two LFSRs may be added (modulo 2) to generate a Gold code. The block diagram and descriptions for the single LFSR and a Gold code are shown below. This interpretation of XML is specified in the standard at the level of <polynomial>.



(a)

```
<code>
  <polynomial>3</polynomial>
</code>
```

(b)

```
<code>
  <polynomial>3</polynomial>
  <polynomial>9</polynomial>
</code>
```

The implicit interpretation of XML in different contexts is rather straightforward for the description part of the standard. However, describing the radio capabilities is a lot more complicated. For example, how do we convey a radio that can support an interleaver before Reed-Solomon, but not after, versus one that can put the two blocks in any order? Likewise, how do we indicate that Gold codes are supported but single LFSR is not and vice versa? A solution for the second question is now part of the standard, while the first question is still being debated.

4. STANDARD CONTENTS

The field of wireless communications is very large. Hundreds of different waveforms have been developed over the past 60 years. The standard will initially cover only digital waveforms, but may be extended to include legacy analog waveforms as well. No single researcher, or indeed an organization, can claim to know all the variations and different attributes a waveform may possess. This consideration makes collaborative development of SDRPHY essential. The following section outlines some of the fundamental properties of waveforms and attempts to organize them.

- Frame structure
    - TDM[A] parameters are considered part of the frame structure.
- Carrier frequency and power
- Modulation – mapping of bits to symbols

- Coding – may include zero or more stages
- Cryptography
- Spread-spectrum (DSSS, hopping)
- Symbol rate

A description of the waveform itself may be sufficient to implement a transmitter, but is often not sufficient to implement a receiver. The operating environment may dictate different receiver architectures. For example, a BPSK receiver in a line-of-sight environment is much simpler than in a multipath environment. The receiver may also need the following parameters:

- Acquisition uncertainty (frequency, time)
- Dynamics – is it stationary or on a fighter jet?
- Environment – multipath, line of sight
- etc.

As an example, let us consider the description of the frame structure.

Packet and Frame Structure

Most practical communications systems impose a packet and/or frame structure on the transmitted symbols. The framing may be taken care of at the bit level (e.g., older SATCOM signals), or may have to be considered at the symbol level (e.g., DVB-S2). Some standards use different modulation and/or coding for different parts of the frame. The frame structure for these standards must be described before a transmitter or a receiver can be created. Any frame structure that can be handled at the user data bit level is beyond the scope of this standard.

A canonical wireless transmission frame can be described as a sequence of fixed-duration segments. Each segment may be described by a different waveform (e.g., a preamble may use a robust modulation such as BPSK, while the bulk data uses an advanced modulation). Most frames consist of just two segments—preamble and bulk data. The preamble is typically (but not always) a combination of a fixed pattern and a short user-defined sequence.

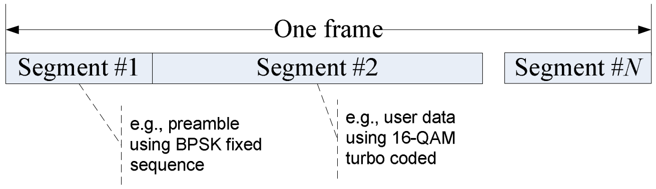


Figure 1. Canonical wireless data frame structure

A frame is defined in XML as a sequence of segments. Each segment is defined by

- <duration>. Set the time duration of the frame segment to be transmitted using the selected waveform. The duration is converted to symbols and/or chips based on the symbol rate set in the waveform.
- <data_source/sink>. The data source selects either an external source or fixed data sequence.
- <waveform>. As with any XML for this standard, the waveforms may be defined inside the segment definition, or reference previously defined waveforms.

The data source or sink provides an interface between the physical layer and upper layers. A preamble can be specified as a fixed data sequence. User data or variable header data are specified as *strings*. For example, a DVB-S2 frame consists of the following three segments:

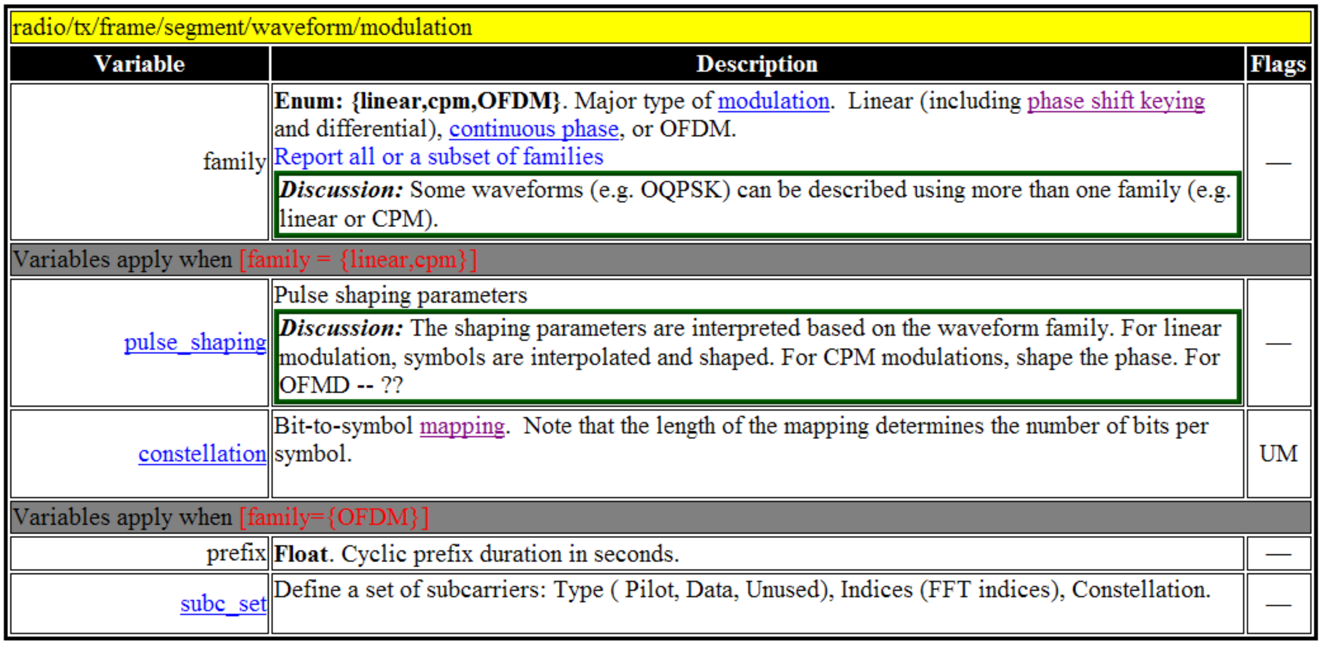
```
<frame>
  <segment> <!--Start of frame is 16
    <duration>1.6e-6</duration>
    <data_source><set><e>0</e><e>1</e> …
  </segment>
  <segment> <!--PLSCODE is 64 symbols-->
    <duration>6.4e-6</duration>
    <data_source>pls_code</data_source>
  </segment>
  <segment> <!--DATA is 16200 symbols-->
    <duration>1.62e-3</duration>
    <data_source>data</data_source>
  </segment>
</frame>
```

The first segment uses a fixed sequence, while the next two define interfaces to the upper layer. SDRPHY does not specify a mechanism for the upper layers to read/write to the named data sources.

The frame may also be characterized by the TDM structure in which it must operate. The TDM structure determines the periodicity of frames and the absolute timing.

5. COLLABORATIVE DEVELOPMENT

Collaborative development of the standard is key to its success. An interactive website has been created to allow multiple contributors to add and modify the standard and to allow collaboration with colleagues in industry and academia. The website also provides a means of distributing the current version of the standard. The site, http://SDRPHY.org, is owned and operated by The Aerospace Corporation. However, the standard itself is released into public domain. Access to viewing and editing is access controlled. A simple one-page ‘terms of participation’ document has to be executed with The Aerospace Corporation to obtain an account. <u>This agreement is not an NDA</u>. A representative screenshot of the viewing interface provided by the website is shown in Figure 2.



radio/tx/frame/segment/waveform/modulation		
Variable	**Description**	**Flags**
family	**Enum: {linear,cpm,OFDM}**. Major type of modulation. Linear (including phase shift keying and differential), continuous phase, or OFDM. Report all or a subset of families ***Discussion:*** Some waveforms (e.g. OQPSK) can be described using more than one family (e.g. linear or CPM).	—
Variables apply when [family = {linear,cpm}]		
pulse_shaping	Pulse shaping parameters ***Discussion:*** The shaping parameters are interpreted based on the waveform family. For linear modulation, symbols are interpolated and shaped. For CPM modulations, shape the phase. For OFMD -- ??	—
constellation	Bit-to-symbol mapping. Note that the length of the mapping determines the number of bits per symbol.	UM
Variables apply when [family={OFDM}]		
prefix	**Float**. Cyclic prefix duration in seconds.	—
subc_set	Define a set of subcarriers: Type (Pilot, Data, Unused), Indices (FFT indices), Constellation.	—

Figure 2. Representative screenshot of the standard *viewing* interface

6. CONCLUSION AND FUTURE WORK

The need for a standard to describe the physical layer for different radios is clear. This paper does not propose any groundbreaking ideas or present new results. Instead, we expect the new standard to foster better collaboration between research groups in industry, government, and academia. Novel results can be attained by leveraging the strengths of each group (hardware, system, algorithms). The main goal of this paper is to solicit input and contributions to the nascent standard. The standard is and will remain free of NDAs. The Aerospace Corporation is funding this effort until the time comes to hand it over to a true standards body such as The SDR Forum or IEEE. The standard proposed in this paper attempts to capture the intent of the waveform rather than the implementation. This approach is certain to be controversial and does limit the application of the standard to *reasonable* waveforms.

At this point, the SDRPHY can be considered a skeleton, with just enough tags to configure simple systems. The capabilities description part of the standard is even less mature. How would one describe the capabilities of a fully software-based radio – after all, *any* waveform can be implemented given enough code? Capabilities are also frequently determined as a trade-off (e.g., higher data rates are supported with simpler error correction).

Another challenge is determining a common set of waveforms supported by two or more radios by computing the intersection of the capabilities reports.

We are currently working on developing three interpreters:

- GNURadio [6] via the GNURadio companion
- A commercially available DVB-S2 modem
- An in-house-developed FPGA-based software defined radio.

We plan to demonstrate interoperability between the different platforms configured with the same XML file.

10. REFERENCES

[1] JTRS Software Communications Architecture http://sca.jpeojtrs.mil/

[2] M.C. Scardelletti, R.C. Reinhart, M. Andro, J.H. Glenn, "Software Defined Radio Architecture for NASA's Space Communications," *High Frequency Electronics*, July, 2007

[3] D. Scaperoth, B. Le, T. Rondeau, D. Maldonado, C.W. Bostian, S. Harrison, "Cognitive Radio Platform Development for Interoperability,"

[4] http://en.wikipedia.org/wiki/Reed-Solomon_code#Reed-Solomon_codes_as_BCH_codes

[5] M. Bayer, "Analysis Of Binary XML Suitability For NATO Tactical Messaging," http://theses.nps.navy.mil/05Sep_Bayer.pdf

[6] GNU Radio, www.gnuradio.org

[7] E. Nicollet and L. Pucker, "Standardizing Transceiver APIs for Software Defined and Cognitive Radio," RF Design Magazine, Feb 2008