**Spring**

2012

ARINC 838 Agile Gap Analysis

Team Information Overload V2

Mike Deats, Scott Griffin, Ryan Neal, Brandon Sutherlin, and Liron Yahdav

08

**Fall**

Team Information Overload 2.0 (IO2) implemented the ARINC 838 loadable software draft specification. Throughout the development process, the team identified certain gaps in the specification. The following report lists these gaps and their current resolution status. There are 20 gaps identified.

* 4 have been resolved during the project
* 3 have been partially resolved
* 2 have been possibly resolved
* 11 are unresolved

| Gap ID | Description | Specification Section | Description | Resolution Status |
| --- | --- | --- | --- | --- |
| G001 | Clarify Sections of SDF | 5.2.1.3 | Confirm that a Software Definition File can contain a collection of sections with each set of sections representing one LSP. (e.g. Software Definition File can contain a description for software-partnumber 'A' and a description for software-partnumber 'B' ...). | **Resolved** in spec 2012-02-22-V82 by removing sdf-sections section from XDF structure. |
| G002 | Sample file uses sdf namespace which is not defined in in the xsd. | Appendix D | We need to know if the XSD is incorrect, if the sample file is incorrect (our assumption), or a third option. | **Resolved** in spec 2012-02-22-V82 by specifying default namespace in XSD. |
| G003 | Difference between LSP integrity and SDF Integrity CRCs | Appendix N | It seems the SDF integrity is a checksum of the data files. LSP integrity is checksum of the entire LSP. In conjunction, they allow the confirmation that the LSP integrity is intact. If not intact, it indicates which data file has a corrupted CRC. Spec does not make it clear how to calculate these. | **Partially resolved** in spec 2012-02-22-V82. Still ambiguous how to exactly calculate these CRCs. Also, it seems that the LSP CRC should include the XDF in some aspect, but it doesn’t. |
| G004 | Should strings in binary be escaped as required in XML? | 5.1 | The spec requires that certain characters be escaped in XML strings. The same requirement is not made in the binary section. If escaping is not required in binary, we recommend that it be explicitly called out in the binary section. | **Resolved** in spec 2012-02-22-V82 by explicitly mentioning this escaping is XML specific. |
| G005 | BDF Target Definitions says UNIT32 length is "even" instead of "4" | 6.2 | In the Target Definitions section, the “length thw-positions” is defined as type UNIT32 and is of length "even." It seems this should say "4" (for 4 bytes in the UINT32). | **Unresolved**. |
| G006 | CRCs would be simpler to store as UINT32 instead of STR64K | 4.1 | It seems like it would make more sense to store CRC values as binary data (UINT32) instead of strings, as the spec describes. This would simplify the code and matches more typical CRC usage. | **Resolved** in spec 2012-02-22-V82 by introducing HEXBIN data type. |
| G007 | “Software Definition File” term confusing | 3.3.1.1 | The term “Software Definition File” is somewhat misleading. The use of the word file is leading us to believe that another actual file exists. Particularly the sentence, "A one-to-one relationship exists between the binary version (BDF) and XML version (XDF) of the Software Definition File (SDF)." A better term may be “Software Definition Format,” which is what we have been using in our heads. | **Partially resolved:** From Chris Ellison 2/24/12: Feedback sent to the spec authors. |
| G008 | Software part number check character algorithm is poorly written | Appendix G | The check character algorithm documented in Appendix G of the specification is very confusing. The core algorithm is to take each of the non-check/delimiter characters, XOR the binary representation of the characters together, and return the result in hexadecimal format as an upper-case string. However, the algorithm in the spec specified first converting them to hexadecimal, then to binary, then to "add the binary digits using mod 2 arithmetic." First of all, the conversion to hexadecimal step is not necessary. Second, the binary arithmetic statement is simply a binary XOR, but the way it is worded can easily lead an implementer down a path that is both costly and error-prone. | **Unresolved** as of 2012-02-22-V82. |
| G009 | Verify encoding of STR64K data type | 4.1 | XML is UTF-8, so it seems reasonable to use UTF-8 for STR64K in the binary files as well. | **Partially resolved** in spec 2012-02-22-V82. STR64K is defined as being ASCII, but there is no explicit mention of how it should be encoded in an XML, which is UTF-8. |
| G010 | First two bytes of the Hexbin32 and Hexbin64 | 6.1 | 1. On page 29 of the spec, HEXBIN32 is defined as an unsigned integer number. Page 33 indicates the length is 2 + the number of hex digits for the file integrity value. This is consistently true for the LSP integrity and SDF integrity values on page 33. What are the first two bytes? The length of the HEXBIN? 2. There is not a definition for Hexbin64 on page 29 of 81. | **Unresolved** as of 2012-02-22-V82. |
| G011 | Clarification on CRC types | Attachment 1 | The provided CRC\_Expected\_Values.xml does not have expected 64-bit CRC values for the files. The spec (ATTACHMENT 1) mentions that 64-bit CRCs are valid, so we are not sure why that data is missing. In addition, the spec (ATTACHMENT 1) does not specify that 8-bit CRCs are used, but they do exist in the CRC\_Expected\_Values.xml file. Can we get clarification on which CRC types should be supported and make sure we have expected values for all of those? | **Possibly resolved:** From Chris Ellison 3/1/12:I've fed this back to the spec authors already. |
| G012 | NULL used as indicator for last element. | 6.2 | In section 6.2, absolute pointers are defined for SEQUENCES to indicate the next element in the SEQUENCE. For example, in the file pointer sequence the next file pointer's value is described as "absolute pointer to next file definition, or NULL for last pointer." Should this be 0 for last pointer? Alternatively, does this mean an empty byte to represent NULL? If it should be 0, this needs to be corrected for each SEQUENCE element. | **Possibly resolved:** From Chris Ellison 4/21/12: I believe this to be a clarification issue only - ASCII NULL is 0x00, but ASCII NULL is not the same as "Null" as a concept. I have recommended clarifying the spec to state "a null character (0x00)" or similar. |
| G013 | Use of SEQUENCE in binary definition is not consistent. | 6.1, 6.2 | A SEQUENCE is defined as elements that must be placed in order. However, everything in the binary definition meets the criteria. SEQUENCE should either be used on everything, or eliminated as redundant. SEQUENCE is defined in section 6.1 and used in section 6.2. For example, the SEQUENCE is used on the File Definition. However, the pointer to the next file definition is outside of the SEQUENCE. This caused some confusion among team members. | **Unresolved** as of 2012-02-22-V82. |
| G014 | Spec doesn't specify where file definition files are located relative to XDF or BDF | Appendix D | The BDF/XDF files reference “file definition” files as file names only, without full paths (see sample XDF file in spec Appendix D). When we validate the CRCs on these files, we need to know exactly where they are located. For now, we will assume that those files are located in the same path as the BDF/XDF that was initially read in. It may be a good idea to clarify this in the spec. | **Unresolved** as of 2012-02-22-V82. |
| G015 | List of invalid characters for data file names is incomplete | 4.4 | The spec lists several invalid characters, however there are other characters that should be excluded from file names: !, ', &, ;, ^, `, @, +, %, /, and , | **Unresolved** as of 2012-02-22-V82. |
| G016 | Conflicting definition for target hardware definition list and position list. | 4.1 | Section 4.1 of the spec defines a LIST0 element as having 0 or more sub-elements. Section 4.1 also indicates that the only fields that are LIST0 is the target hardware definitions and target hardware positions. However, in Section 4.5 of the spec, the definition for these lists indicate that if the "LSP does not list the hardware that it could be loaded to then only 1 THW\_ID should be listed with a string 'ANY'." Similar text defines the same for the positions. This seems to indicate that the target hardware definition and positions lists will never be empty, thus nullifying the need for a LIST0 element. | **Unresolved** as of 2012-02-22-V82. |
| G017 | File name subset not defined. | 4.4 | In section 4.4 of the spec, the file-name field is defined as being restricted to a "subset of 7-bit US-ASCII." We do not find this subset defined in the spec. It is ambiguous if "subset of 7-bit US-ASCII" means 7-bit ASCII or some smaller subset. | **Unresolved** as of 2012-02-22-V82. |
| G018 | The spec lists 'Target Hardware Definitions' as both a LIST0 and LIST1 | 4.1, Appendix C | In section 4.1 “thw-definitions” is defined as a LIST0, but in the XSD schema, and binary definition it is a LIST1. | **Unresolved** as of 2012-02-22-V82. |
| G019 | Spec should indicate why standard XML boolean values are not used. | 5.1 | Section 5.1 indicates that lower-case “true” and “false” should be used for boolean values. The authors have a very good reason for doing so in that the CRC values would be different if all standard XML boolean values are used. They should provide the reasoning behind their decision. However, the CRC does not currently utilize the XML file, so the argument may be invalid. | **Unresolved** as of 2012-02-22-V82. |
| G020 | XML restrictions are onerous. | 5.1 | Section 5.1 of the spec goes into detail about whitespace and carriage returns for the XML file. This is quite onerous as most XML parsing tools can handle these issues. The rational given is it simplifies the comparison between binary and XML, however, we found this not to be the case. Because we use an XML library, there is no issue. This could be removed from the spec. However, if the library would need to be certified, one may consider writing a custom parser, and then this level of rigor would be appreciated. If the rigor is due to CRC requirements, again the XDF is not used in the CRC calculation, making these requirements irrelevant for that purpose. | **Unresolved** as of 2012-02-22-V82. |