The Use of File Description Languages for File Format Identification and Validation

Matthew Dunckley[(1)], Stephen Rankin [(1)]

and

Esther Conway [(1)], David Giaretta [(1)]

[(1)] Rutherford Appleton Laboratory (RAL), Science & Technology Facilities Council (STFC)
Chilton, Didcot, Oxon OX11 0QX, UK
EMail: m.j.dunckley@rl.ac.uk, s.e.rankin@rl.ac.uk

ABSTRACT

If an archive is to digitally preserve scientific data it is vitally important that the archive also curates the OAIS [1] representation information for that data. The representation information for the data can provide the archive with many additional capabilities, one of which is the ability to identify the data file type received from a data producer and another is validating its structure.

There are a range of file identification and structural verification mechanisms available to an archive to verify that the data they are receiving from a data producer is what they are expecting. From simple file extension checking, to the use of command line tools such as the BSD UNIX *file* command or the National Archives Droid tool [4] in conjunction with file format signature registries such as PRONOM [4] or the Global Digital Format Registry [12], through to the use of sophisticated data description languages such as EAST, DRB [8], XML Schema or DFDL [9] as used in the CASPAR project.

The use of file extensions for identification suffers well known disadvantages. Using file signature checking is useful but is not totally reliable due to the lack of granularity identified from the signature and the possibility of false identification through coincidentally equal signature formats, this is especially a problem with bespoke data formats where identification through signatures was not a concern at the time of conception. Using file signatures does not provide any data structure validation. The use of data description languages to describe the internal structure of the data right down to the bit level will provide a more holistic solution allowing a full and reliable identification of a file format as well as validating its structure.

Tools and generic software APIs that could solve the problems of file identification and structure validation by using a combination of file format signatures and data descriptions stored in a OAIS representation information registry will be presented and discussed.

INTRODUCTION - THE CHALLENGE OF FILE FORMAT IDENTIFICATION

Automated file format identification and validation is critically important [2] and becoming a necessary feature for an archive's ingestion mechanism of digital objects such as data files, it would be impossible for a human to work through, identifying files and checking the huge volumes of data archives are receiving. When digital objects are received, whether these are data

products or representation information or preservation description information, they will need to be accurately identified so they can be appropriately handled. A digital object is interpreted by it's representation information, OAIS defines representation information as either structural or semantic, this paper is dealing mainly with structure representation information however how one relates structural to the semantic information is discussed.
Digital objects such as data files will be in either standardised or specialised *ad hoc* data formats. Dealing with well-formed, widely used standardised data formats such as XML is very convenient due to the availability of parsers and query languages. However, on the whole the vast majority of data products are in *ad hoc* and specialised formats. *Ad hoc* data formats are a particular problem because there is little knowledge about them or available documentation [3]. These *ad hoc* data format structures commonly suffer from lack of support; the original creator may have moved on; the data format may be so old the person responsible for the format or the source of information about the format may have disappeared from the designated community long ago.
It is also common to find inconsistencies between a format specification, the actual structure of the data format and its use. Most *ad hoc* data formats used in many industries have not been designed with identification by a file signature in mind. It is quite usual for a data format, especially older formats, to consist merely of ASCII or binary tabular data with little or no metadata contained within, describing the data. These formats may not have any pre-build querying, parsing or transformation tools. There may be little or no documentation about the data format. Data products may be inconsistent and may not match the documented format. Data may be missing and errors may have been introduced for a variety of reasons, such as human data entry errors or file corruption over time. For problems such as these, identifying errors and then recovering the recoverable data can be a lengthy process. Together these problems make format identification a challenging area. Overcoming these challenges is part of the objectives of CASPAR - **Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval** is an Integrated Project co-financed by the European Union within the Sixth Framework Program (Priority IST-2005-2.5.10, "Access to and preservation of cultural and scientific resources"), that started on 1 April 2006. CASPAR will research, implement, and disseminate innovative solutions for digital preservation based on the OAIS reference model (ISO:14721:2002).
Scientific data, especially legacy data is often *ad hoc* or bespoke, but usually a scientific archive will know through negotiation and agreement what it will be receiving from a producer so identification is not key, however validation maybe essential. For archives which have not made an agreement with their producers over what they will receive, this may be for document collection or web archiving, identification and validation will be important.

CURRENT FILE IDENTIFICATION SOLUTIONS

File Extension

One current solution for file format identification is simply to look at the file extension (the series of characters following the dot at the end of a file name). This is an easy but very unreliable method for identification because not all file types have an extension and even if there is one, it is possible that the extension is somehow inaccurate due to error or corruption. There are also many cases where different file types have the same extension. Looking at the file extension alone is also unlikely to provide information such as file versioning. Another solution is to use metadata. It is common within many data formats to have a header section with in the file, containing metadata about the file, these often need to be read by software specifically written for the job, if

the format changes, the software will also need to change to keep up, which can be costly and doesn't always happen in practise.

File Signature

Another more favoured solution is to look for a sequence of recognisable bytes within the file, e.g. from the beginning of the file or offset from the beginning of the file. This is the technique employed by tools such as Droid [4], developed by The National Archives which will batch identify files from byte signatures stored in an XML file and links the identification to the PRONOM [4] file format registry allowing access to representation information describing the data format. The file in question is matched against every signature until one is found. This is the same method employed by the UNIX *file* command, here the byte signatures are stored in the lines of a text file, each signature is matched against the file in question. The UNIX *file* command currently differs from Droid by having a substantially greater number of signatures and allows for more complex matching but this may change as Droid evolves. This concept is fairly effective for identifying widely used standardised formats but for *ad hoc*, bespoke and specialised data files there is a much greater chance of inaccurate identification since there might be a coincidental sequence match. Identification will also fail if there is no knowledge of the file format in a signature file.

Identification of a standardised format may be correct to a certain level - for instance one may identify the kind of a file correctly but need more information to accurately determine what version of the file type is actually being dealt with. For instance looking at a ISO standardised MPEG audio format (commonly known as MP3) one may accurately identify this file as an MP3 from its file signature (454549) and ID version number but to understand which MPEG Standard version (MPEG Version 1 - ISO/IEC 11172-3, MPEG Version 2 - ISO/IEC 13818-3) and which Layer system (I, II, III) the file uses means reading parts of the file header, then determining by calculation on the file Header length the position of the Frame header, which will then allow one to identify the MPEG version and the layer. Once this information is acquired one will then need to use look up tables to determine information such as bitrate, frequency, etc, this is more commonly known as feature extraction or characterisation. Identification to this level would seem nearly impossible by using file signatures alone but to accurately identify, validate its structure and handle it correctly an OAIS may well need to acquire all of this information. This is often the case with many scientific file formats that a lookup table is used to interpret a field of data within a file.

When complete identification may depend on cross referencing information from a lookup table and further post ingestion processing may depend on complete identification this will create a problem. To follow on from this paper we plan to answer the questions: How does one provide this information to an identification tool? What format should the lookup tables be stored in? What representation information do these look up tables themselves require?

The lookup tables can be considered as representation information about the data format and could themselves have representation information about their structure and semantic information describing what they define. One solution is to have this information stored and accessible from a representation information repository related with the file signatures by the representation information network.

Another example of the difficult challenge of identification is the NASA-AMES data standard. NASA-AMES data files are ASCII files of tabular data with a metadata header. The metadata specifies information about the data contained within the file such as the length of the file and the number of data rows. In this case the ASCII data file simply starts with 2 integer values, the first denoting the NASA_AMES version, the second describing how many header lines follow before the data products start. Although it is a fairly widely used scientific standard identifying this kind of data file accurately using file signatures alone is likely to be very inaccurate or impossible.

Data Structure

In order to meet these challenges data description languages are now emerging [5], for example DRB, DFDL and PADS/ML [6]. There are a few older more established languages such as EAST. These data description languages share the common principles of providing a mechanism for describing a data structure or data format to a low level using predefined and user enumerated types. When describing data there is an intrinsic need to specify the representation of the interchanged data, including the logical structure of the data and the physical representation of the individual data items. That is the way the information is structured within the format and the information types the data is composed of. When allowing for the wide diversity of variables such as the operating systems and the machine representations for numerics, a full understanding of data can only be reached by using a rigorous notation/language that provides a complete, non-ambiguous logical and physical description.

By design a good Data Format Description Language should support the preservation process by provided key features, ideally a full formal logical and physical description of the format down to the bit level coupled with semantics information, for instance the logical description needs to record the structure of all the component parts of the format, there size in bits and their location within the description of data. The physical description needs to record the file encoding (ASCII or BINARY), the representation of the data in base data types (such as 32-bit integer) and the Endianness (Most/Least Significant bit ordering). The Data Format Description Language should also allow capturing of semantic information, the meaning of the data fields, their units and any other relevant information. Only with a full format description and the combination of these features should it be possible to fully identify the file. When validating a file from its format description the validation should be able to answer fundamental questions such as, are all of the data fields present in the data set? Which data fields are missing? Do the fields match their described representation? Has part of the file been corrupted?

Usually applications and their associated data formats are written without any advance understanding of how important they may become and when a format does become widely used it is often too costly or too difficult to change the data format later on into something more easily interchanged, data description languages also have the benefit of allowing these legacy data formats to be interpreted in a way easily interchanged and transferred between heterogeneous data systems. With more applications supporting web-service technologies there is a greater need now to describe these legacy formats for the purpose of interchange. In 1969 Mooers stated “A good data descriptive language should be congenial to use, general, transformable, and independent of supporting hardware or software” [7], this is especially true in the field of digital preservation where we can not rely on today’s hardware and software being available in the long term.

With these issues in mind it follows that a full format description should also be able to support data Interchange and interoperability through transformation, for example once we have an EAST or a DRB format description of a binary file it is possible to access and query the binary file as though it is an XML file or transform the file to an XML format and from there it is possible to transform the file further to other formats using XSLT therefore supporting data migration.

CURRENT DATA FORMAT DESCRIPTION LANGUAGES

EAST

EAST, developed through work at The Consultative Committee for Space Data Systems (CCSDS) is an ISO standard data description language with the purpose of supplying complete and non-ambiguous information about the format of some described digital object. The EAST description is used to interpret and gain access to the data entity. EAST has been developed with data description capabilities, human readability, and computer interpretability in mind and where these descriptions are required can provide a complete, precise, and unambiguous description of the file format structure down to the bit level. EAST can be used in conjuncture with the Data Entity Description Specification Language (DEDSL) a standard for recording semantic information developed by CCSDS. EAST has an established user base with the space community and there are several tool kits available for creating descriptions. A reported problem with EAST is a lack of support for describing file structures where the position of an data field needs to be calculated from other data fields.

DRB

Developed by Gael Consultant, DRB [8] (an expansion on EAST) allows the physical and logical description of data structures and file formats through the use of XML schema in combination with SDF (Structured Data File) description tags which can describe a data format down to the bit level. The DRB processor utilises XQuery to provide access, extraction, validation and transformation of data from multiple sources. By using DRB to describe a data format as a set of tree nodes with each node being given an identifier, the tree data structure can be accessed and traversed using XPath expressions. Semantic information can be informally recorded in the description by using the XML schema documentation tag. It is also possible to perform calculations using XQuery from within the data description allowing full description of files where the locations of data fields within a file must be calculated from other data fields. However calculations must be described in XQuery and can lead to increased complexity and a decrease in human readability. Due to DRB using XML schema for the description language it is possible to use standard XML software packages such as XMLSpy to create the schemas.

PADS/ML

PADS/ML is a Data Description Language developed at Princeton University and supported by AT&T, it is itself a functional computer language to formally specify the logical and physical structure. As with other description languages, the description also acts as formally specified documentation about the data format. The PADS/ML compiler, compiles the description into a suit of data processing tools which can be used to access, manipulate and transform the data into other formats. Writing a PADS/ML description is effectively programming in the ML language which means data producers would have to learn or employ people who could produce these descriptions. From a digital preservation point of view PADS/ML does not seem to be a good choice because the ML functional language and software tools may become unsupported, or suffer from lack of standards and a wide user base. PADS/ML does not appear to have good support for describing semantic information.

DFDL

"DFDL is a data description language allowing data to be accessed from its native format and be presented as a logical XML model or indeed converted to the corresponding XML document. DFDL also allows data to be taken from an XML model and written out to its native format."

DFDL [9] has been developed by the Data Format Description Language Work Group with data interchange and interoperability in mind and is a similar concept to DRB in that it enables data formats and structures to be described using XML Schema style notation with the addition of DFDL tags that describe the more low level structures down to the bit level. By having a data format described with a DFDL description which is accessible to multiple applications, one can provide a common interface to the data, therefore allowing data interchange to take place. DFDL is still a work in progress and there is currently a lack of support for it, IBM are working on an implementation as part of their Virtual XML [10] garden suite. DFDL will not support the capture of semantic information but can be used in conjunction with ontologies for this purpose.

SUGGESTED IDENTIFICATION AND VALIDATION STRATEGIES

1) Producer submits a Submission Information Package (SIP) containing data files and referencing a format description for the submitted data files.

A formal data description of the file derived from its standard specification (if it has been previously formally specified) or if this is unavailable, the structure can be inferred for the data by a data analyst who is responsible for it. The Producer constructs a SIP containing data files for Archive ingestion, the SIP logically contains or references a data description relating to the data file. On reception of the SIP the Archive un-packages it into the component data files and will either receive the data description or have to go to the referenced registry to retrieve it. Once retrieved the Archive can validate the data file against the data Description. Signature identification could also be performed but as the data file and data description are already related to each other by the SIP this may not be essential. This strategy is shown as a UML analysis diagram in Figure 1.

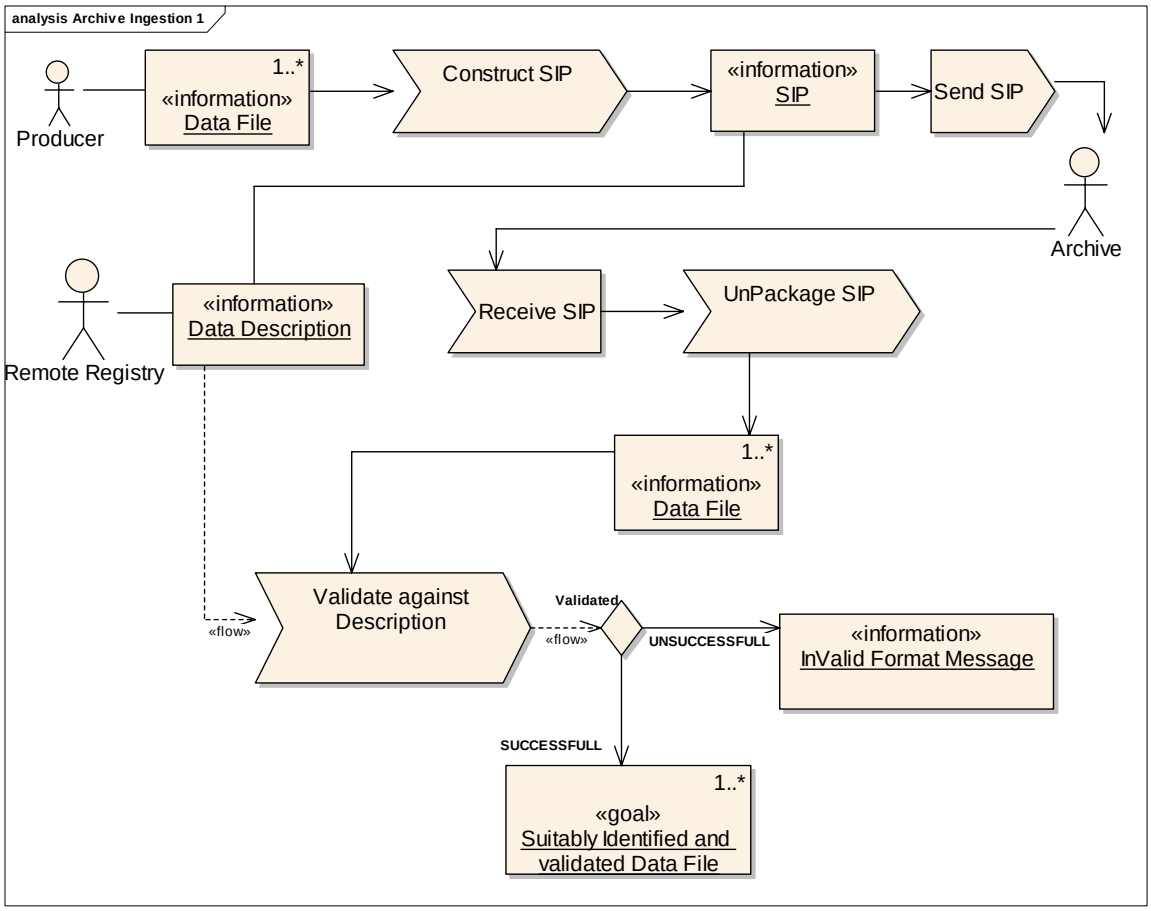

Figure 1 SIP Referenced Identification Strategy

2) Producer submits a set of data files as a SIP, the format is matched through representation information registry.

In many circumstances it will be too costly or time consuming for a producer to form the many relationships between the data files and their data descriptions, another option is to perform a search for a matching data description at the point of ingest. The data description would ideally be held in a representation information net accessible from a registry, ideally linking together all the representation information, semantic and structural needed to preserve the file format for the long term. The format description would be linked to a representation information Label (repInfo Label) that would link together as much representation information needed to keep the file format useful, this could be for instance one or more file signatures, semantics about the file format and technical documentation about the format. The registry itself should be an OAIS and support versioning of all representation information objects.
In this scenario the producer constructs a SIP from the data file to be sent to the Archive, the Archive receives the SIP and un-packages it into the unidentified data files. The Archive management system will access the registry and execute a match on all available signatures with the data file. On a successful signature match, all data descriptions associated to the signature via the repInfo label will be retrieved and matched against the data File. If there is a failure in matching the description against the actual structure within the file, which may result from information missing or the file having been corrupted, it would be ideal to know at which point

the file failed to conform to the description. This strategy is shown as a UML analysis diagram Figure 2 and is a planned implementation for the CASPAR project.

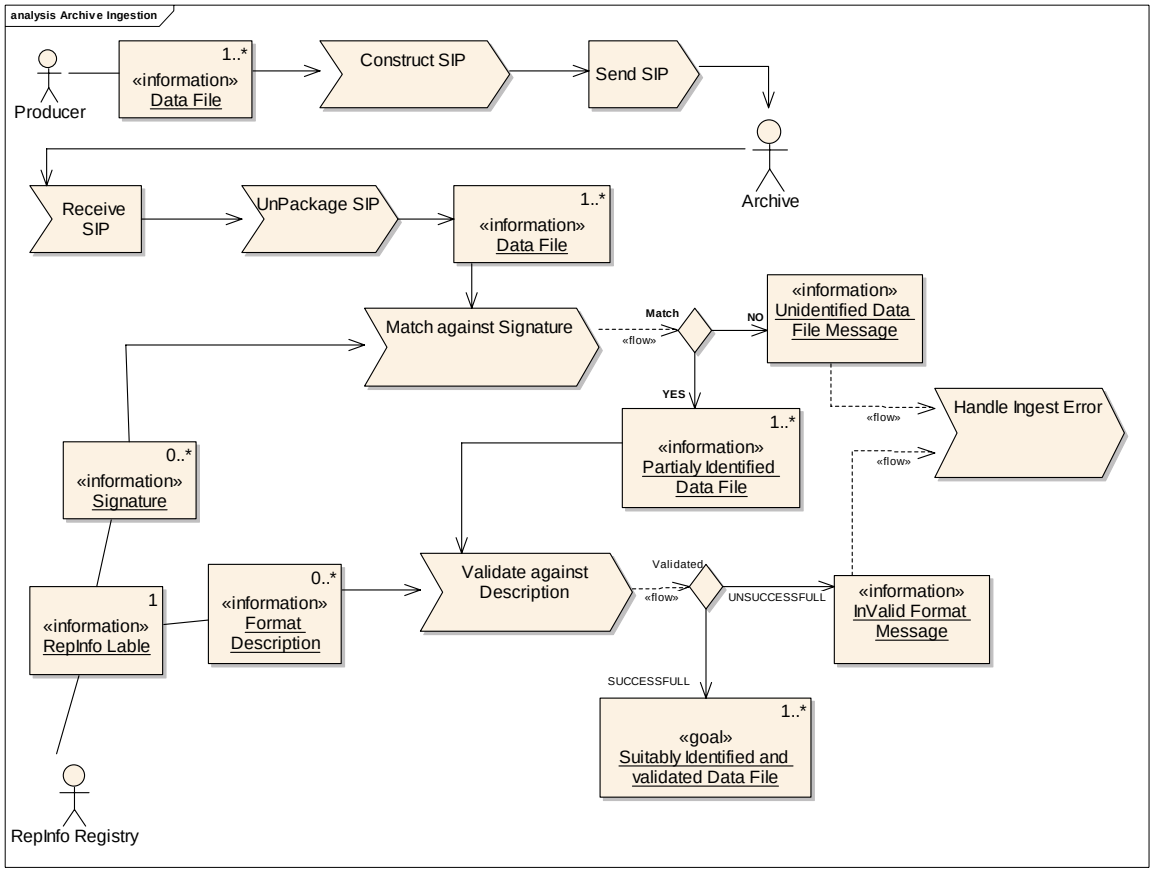

Figure 2: Registry Stored Description Identification Strategy

CONCLUSION

To summarize, at the critical point of ingest complete and robust format identification and validation can be achieved via the use of a formal file format or data description languages to describe data formats and provide structural representation information, descriptions can also provide added support for preservation services, such as feature extraction, characterization [11], the recording of semantic information and provide data access, transformation and interoperability services. A data description will also act as format documentation for the preserved digital object.

Ideally the archive should form an agreement with the producer in which case the archive should know the format of received files but these may still need to be validated. For the cases where there is limited or no agreement between producer and archive on what will be sent then both identification and validation is important.

If complete file format identification and validation is required, identification based on a files internal data types and structure would seem to be the most reliable and accurate method.

A representation information registry could contain file signatures related to file format descriptions and semantic representation information related to the format. Partial identification of a file type/format could be initially done by a file signature and then enhanced by checking

against all related file format descriptions. The CASPAR project will implement and test this planned strategy for structural identification and validation.

REFERENCES

[1] - D. Giaretta: Reference Model for an Open Archival Information System (OAIS). The Consultative Committee for Space Data Systems CCSDS (2002)

[2] - R. U. M. Borghoff, J. Scheffczyk, L. Schmitz: Preservation of Digital Publications, An OAIS Extension and Implementation. Institute for Software Technology, University of the Federal Armed Forces, Munich (2003)

[3] - A Stanescu: Assessing the Durability of Formats in a Digital Preservation Environment. Digital Collections and Preservation Services OCLC Online Computer Library (2005)

[4] - R. Lechich: File Format Identification and Validation Tools. 4th IAA International Symposium on Small Satellites for Earth Observation, Yale University (2007)

[5] - K. Fisher, Y. Mandelbaum, D. Walker: The Next 700 Data Description Languages. Prinston University (2006)

[6] - Y. Mandelbaum, K. Fisher, D. Walker, M. Fernandez, A Gleyzer: PADS/ML: A Functional Data Description Language. Princeton University (2006)

[7] - C. Mooers: Data Description Languages. (1969)

[8] - S. Mbaye: Baseband Data Archive InterChange Format. CEOS ICF, GAEL Consultant (2001)

[9] - M. Westhead: Data Format Description Language – XML Representation (DFDL WG), EPCC, University of Edinburgh (2003)

[10] - K. H. Rose, S. Malaika, R. J. Schloss: Virtual XML: A toolbox and use cases for the XML world view, IBM Systems Journal (2006)

[11] - S. Abrams: Knowing What You've Got, Format Identification, validation and Characterization. DCC/LUCAS joint workshop (2006)

[12] - S. Abrams: Global Digital Format Registry, An Interim Status Report. iPress (2006)

BIOGRAPHY

Matthew Dunckley and Stephen Rankin are information systems developers working within the Space Science and Technology Division for the UKs Science and Technology Facilities Council. They are both actively working and participating in The CASPAR digital preservation research project. Matthew has studied Computer Science and Electronic Engineering at Aston University, UK. Stephen has a degree in Applied Physics and a Ph.D in Physics from Manchester University, UK.

ACKNOWLEDGEMENTS

The authors want to acknowledge all those who are participating on the CASPAR project.