Project Number	IST-2006-033789
Project Title	Planets
Title of Deliverable	Basic design for the extensible characterisation languages
Deliverable Number	PC/2-D1, PC/2-D2
Contributing Sub-project and Work-package	SP/PC/2
Deliverable Dissemination Level	Internal public
Deliverable Nature	Report
Delivery Date	31th October 2006
Author(s)	Volker Heydegger, Johanna Neumann, Jan Schnasse, Manfred Thaller

Abstract

This document contains the first description of the overall architecture of the two XML based languages for describing and extracting characteristics of digital objects. It starts with some conceptual clarifications reflecting discussions at contact meetings with various partners during the first project months. The bulk of the document consists of XML schema definitions for the two languages and sample documents showing their practical application.

Contributors

Person	Role	Partner
Volker Heydegger	Author	University of Cologne
Johanna Neumann,	Author	University of Cologne
Jan Schnasse	Author	University of Cologne
Manfred Thaller	Author	University of Cologne

Table of contents

1. Overview

1. 1. Scope and purpose of this document

This document delivers the schema definitions for the two XML-based languages at the heart of the characterisation logic to be supported by the work packages PC/2 and PC/4 of Planets (and implicitly used in PP/5). It augments these definitions with a short discussion of the principles upon which they have been constructed.

The schemata are functionally complete in the sense they have been found sufficient in a proof of concept implementation of characterisation extraction software to process one class of files. This does not imply, that the schema definitions will not continually develop further during the Planets project. It *is* expected however, that the basic architectural design underlying the two languages, can support an extremely wide ranging extension of the scope of processable file formats.

This Overview – part 1 – of the document describes the abstract principles behind the design of the two languages. Part 2 describes the current technical documents of the XCEL, part 3 those of the current XCDL and part 4 the current technical definitions of abstract names used by the two languages. Parts 2 – 4 are mainly given in the form of the XML schemata used for the definition of the actual languages currently being processed and produced by the proof of concept implementation. They are not always explicitly linked to abstract concepts of part 1. (E.g.: While part 2 describes an XCEL description as a sequence of blocks, part 3 gives the definitions for the types of block which can actually occur within a file, without explicitly labelling them as “instances of a block”.) Part 5 gives the XCEL for the (almost complete) PNG format.

The two languages have been created at Cologne by the team of Volker Heydegger, Johanna Neumann, Jan Schnasse, based on an idea by Manfred Thaller. Jan Schnasse has in the current stage mainly worked on the implementation of proof of concept software to bring the two languages alive; in that role, he has contributed heavily to their emergence.

The main responsibility for the main parts is as follows: Part 1 M. Thaller. Part 2 J. Neumann,; Part 3 V. Heydegger, Part 4 V. Heydegger, J. Neumann, J. Schnasse, Part 5 J. Neumann, J. Schnasse.

1.2. The XCL rationale and vision

The extensible characterisation languages have been derived from a simple and straightforward vision.

A file exists at a given time in a “format” *oldFormat* (see below), which may pose difficulties for its long term preservation, e.g. by becoming rapidly obsolete, as more recent software ceases to use that format. One or more software systems exist, which can convert the data from this file format to another one, *newFormat*. We need the possibility to answer one or more of the questions:

1. Is all information contained within *oldFormat* also contained within *newFormat*?

2. Is all information, which is relevant for the usage of the information, within *oldFormat* also contained within *newFormat*?
3. Is the conversion process *a(oldFormat, newFormat)* better than *b(oldFormat, newFormat)* , i.e. does it preserve more of the information contained within *oldFormat?*

Two answer any of these questions, two prerequisites are obviously necessary:
(a) If we want to compare the information contained within two file formats, it must be possible to express it in a way, which is independent of these two file formats. Corollary: If we want to apply the procedure answering the question to *all* existing file formats, the way to express it must be independent of *all* existing formats.
(b) Question 2 above implies, that different chunks of information are of different importance. A simple and obvious example relates to the way in which information is compressed. If a file format *a* offers the (lossless) compression methods *m(a) = {p,q,r}* while a file format *b* offers the compression methods *m(b) = {s,t,u}* the information, that while encapsulated in a file of format *a,* the information we want to preserve has been compressed using *p* is irrelevant for almost[1] all processing after that information has been transferred into a file of format *b* using compression method *t*.

This leads to the following requirements:

1. We need a medium in which to express in an abstract way the information contained within any file. This is henceforth called an extensible characterisation *definition* language (XCDL), defining the information contained within a chunk of binary data.
   (Within Planets PC/2 and PC/4.)
2. We need an algorithm and an implementation of such, which allows to compare two of these descriptions of the information contained within chunks of binary data.
   (Within Planets PP/5.)

As the procedure described in requirement 1 above (expressing the information contained within a chunk of binary data) needs to be applied to thousands and millions of files, it is obviously necessary to automate it. If we would "express the information contained within a chunk of binary data" manually, we would need a humanly readable description of the way in which information is encoded within this binary data. Such humanly readable documents are traditionally called file format descriptions. If we want to perform the same service automatically, we could decide to write a program for each existing format description which is able to translate files being encoded according to that format description into the XCDL postulated above.

A considerably more generic solution, however, is reached if we provide a language, which allows to express all existing humanly readable file format descriptions in a general machine interpretable language, which allows a general program to translate any chunk of binary data into an XCDL

[1] The exception being such processing as may be needed at a later stage to decide about the autheticiy of the information of information which has been transferred over time across many file formats.

description of the information it contains, if it is possible to express the rules according to which information has been encoded in that chunk of binary data in this machine interpretable language.

3. This requires the definition of an extensible characterisation *extraction* language (XCEL) which describes the way in which information is extracted from a given file.

Both languages together are either referred to as XCEL/XCDL or extensible characterisation languages (XCL).

The vision of both languages assumes, that they are general enough to describe any existing file format; and therefore to extract characteristics from any existing file.

1.3. File formats, file characteristics and the processing of XCL.

1.3.1 Introductory remarks.

To process chunks of binary data, it is necessary to understand, how they express information. The way in which a file expresses information can be considered a first definition of "file format". At this abstract level it is irrelevant, whether a specification of this file format exists as a printed document, as strategic knowledge of the programmer responsible for the software handling the specific collection of information, or as a set of lucky guesses of a hacker extracting information from a proprietary digital object, about which no processing instructions have been released. For the scope of the XCL it is irrelevant, in which of these ways the format of a file is brought to the attention of the person writing an XCEL specification of that format, which allows the processing of the files encoding information according to that format. If the format of a chunk of binary data in the sense just discussed is unknown, it cannot be processed. In that sense reverse engineering can be described as the discovery of the format of a file. There is no such thing as a file "without a format".

At the other end of formality, the term "file format", as used for PDFs, JPEGs or similar categories of files, stands for huge systems of rules for the encoding of information, which provide in some sense a language for the encoding of information. It is easily possible with many of the broader format specifications to use the same file format to express two files, which share almost none of the features used to encode them, as it is easily possible to write two English sentences sharing not a single verb.

In discussion of file formats for long term preservation it has recently been increasingly understood, that the simple decision to use "PDF" or "TIFF" does not mean all that much, as in both cases it is possible to create either files with very high as well as very low preservational value. In the case of extremely rich formats, like PDF, this has lead to the definition of subsets of the rules comprising the format, which are preservationally safe. This has furthermore lead to a tendency, to identify informal "subformats" of file formats (e.g. http://www.digitalpreservation.gov/formats/content/still_preferences.shtml).

While it is very wise to identify the properties of files, which introduce dangers for their preservability, we oppose the formalization of such informal groupings of files with specific characteristics into formal “format variations” or “classes of digital objects” described by observed properties. While many file formats restrict the way in which individual characteristics can be combined, these restriction follow rarely concerns about the preservability of the file, so the way in which the characteristics of a file are combined are almost as unpredictable as the vocabulary used for the expression of a sentence in English.

These considerations are important, as they form the background of what “characteristics” “characterize” a file and which ones have to be extracted. A good case in point is that of a lossless compression, as already discussed above. The compression algorithm used for the compression of a file is a characteristic of the file, which is very important for long term preservation, as it implies that the file can only be preserved, as long as the specific algorithm used for that compression remains documented, more realistically: as long as an implementation for it exists within the overall preservation system. In this sense, it is an extremely important preservation characteristics. If we evaluate, on the other hand, whether this is an important characteristic, if we evaluate, whether a file contains the same information after and before a specific migration, it is totally irrelevant technically: Once the software which converts the file during a migration into a new format has done its job and the uncompressed byte streams which can be extracted from the original and the migrated version of a file are identical, for technical preservation it is absolutely irrelevant, which compression algorithm has been used in the first generation of the file. Taking another turn, we could argue, however, that knowing which compression has been used originally, can become a major characteristic for clarification of the authenticity of a file. (Indeed, one could even argue for a requirement, by which migrated files are augmented by a “technical trail”, which documents which characteristics of that type have been lost within each step of migration.)

As, therefore, the relevancy of characteristics of a file for preservation purposes is heavily dependent on the context within which the extracted characteristics are to be used, is a moving target in other words, it seemed to be implausible to start the design of XCL with a fixed set of “characteristics relevant for preservation”.

1.3.2 Basic assumptions I: f*ile formats*

The following assumptions are underlying all further architectural considerations for the XCL and the software which processes it:

Assumption 1: A *file format* is a set of rules which formalize *all* knowledge needed to process the binary information contained within a distinct and complete block of binary information, traditionally called a file.

Assumption 2: The extensible characterisation *extraction* language is designed to be able to express *all* such rules within a given file format. The extensible characterisation *definition* language is designed to be able to describe *all* the information contained within a file the format of which is described by a valid XCEL description.

Assumption 3: A specific XCEL description is *not required* to express all the rules within a specific file format. A XCDL derived from such a partial XCEL will, therefore, potentially also contain only *part* of the information of a file encoded in that format.

We differentiate here therefore, between the power of the mechanism to be designed, which has to be universal, and the applications which are realized by using that mechanism.

That difference, and the importance of that difference, becomes presumably immediately clear, if we look at a practical example:

Conceptually the XCEL shall be able to express *any* possibility of a PDF-file. For most applications, there will only be a pretty small amount of these possibilities, however, which are actually relevant to specify the preservation characteristics of a specific file, program or process. (The precise positioning of a footnote will in most, though not all cases, be rather irrelevant.) Given the complexity of the PDF specification, it will probably take a considerable time to translate it completely into XCEL.

A translation of the 50 or 100 features of information which are most relevant for the preservation of the content of a PDF, or for an evaluation of the risks inherent in preserving it unmigrated, into XCEL is, however considerably easier to achieve and will provide central services rather soon.

1.3.3 Basic assumptions II: *characteristics*

The rules which are used to structure information within a bytestream, the file format, as discussed so far, can in many ways be seen as the *syntax* of a language to encode a specific type of information.

The differences between file formats do not end there, however. A PNG and a PDF file do not only use different rules to signal, that a sequence of bytes signifies the “width” and the “height” of something, they also use different implicit assumptions about the meanings of these characteristics: The width and height of an image signalling the amount of pixels which can be extracted from the bytestream, the width and height of a printed page an (almost) arbitrary statement about the way in which its should be formatted.

An XCEL description must first of all describe the syntax of the way in which the information structured, basically: by which procedure do you extract a certain item of information from a bytestream. It is *also responsible*, however to map the description of the content of a specific item of information, the meaning of which is defined within the format document, unto a more general concept which is independent of the internal logic of a specific file format.

These more “general concepts” are the *characteristics* of a file to be extracted for preservational (and other) purposes. For the time being, we assume, that these characteristics are ordered best by assigning them to broad baseline categories of filetypes – images, text, sound etc. To be able to start working with the languages as soon as possible, the terminology for these characteristics is currently expressed in XML Schemata called name libraries. It is expected, that these definitions will later on be refined into an ontology of file characteristics.

Behind this concept is finally

Assumption 4: All XCDL definitions describe the *characteristics* extracted from files within a consistent abstract terminology. XCEL descriptions can be understood as a mapping between procedural descriptions of the location of chunks of information within a physical file and their abstract interpretation outside of the interpretations inherent in a file format specification.

1.4. Abstract basic language architecture

1.4.1.1. Abstract basic XCEL architecture

The extensible characterisation extraction language understands all existing physical files as a series of (potentially recursively) hierarchically ordered sets of blocks of information. Between individual blocks additional, non hierarchical, relationships are possible.

Each block of information can be expressed by:
a *start*.
a *length*.
an *interpretation*.

The *start* of an XCEL block of information can be (a) an absolute byte:bit offset from the beginning of the physical file, (b) a relative byte:bit offset from the beginning of a block or (c) the result of the execution of a procedure *p(begin, trigger)* at a specific position of the parsing process.

Start procedures are defined by two parameters: the *position* – usually the current position within a parsing process – at which the search for a block has to begin, and the *trigger* which indicates the start of the block sought for, expressed in the codeset of a file. A start procedure within an XML based file format would therefore typically be described as *p(current_Position, “<someTag>”).*

The *length* of an XCEL block of information can be (a) an absolute byte:bit number or (b) the result of the execution of a procedure *q(trigger,filter,implication)* .

Length procedures are defined by up to three parameters: the *trigger* which indicates the end of the block sought for, expressed in the codeset of a file, a *filter* procedure, which describes whether some segments of the byte stream have to be excluded from the analysis when looking for the *trigger* and a set of *implication* rules, describing implied ends of a block, where no *trigger* can be found. A length procedure within an XML based file format would therefore typically be described as *q(“</someTag>”,pair(“<[a-zA-Z0-9]*>”,”</&>”),signalXMLError),* while a *length procedure* within an XML based file format could frequently be described as *q(“</someTag>”,pair(“<[a-zA-Z0-9]*>”,”</&>”),implyBy(“</someOtherTag>”)).*

Note: The actual language implemented so far makes no provisions for the specification of start or length procedures. They have been added here to explain the strategy to expand the current specifications to such start / stop – symbol formats.

The *interpretation* of a block consists of all parts of the XCEL describing it as the expression for some characteristic of a file and / or as a container for other blocks.

1.4.1.2. Classes of XCEL symbols.

Any XCEL description consists of three types of symbols, which are described by XML elements.

Standard symbols describe the relationship between a chunk of information contained within a file and an abstract characteristic.
Implicit symbols describe a characteristic that is implied by the file format. (All TIFF files are image, not sound files.)
Transient or processing symbols describe information which is needed to process the content of the file, do not provide any information which can be meaningfully transferred into an abstract format. (E.g. the start of an internal symbol table within a file.)

1.4.1.2 Classes of XCEL symbols.

Any XCEL description consists of three types of symbols, which are described by XML elements.

Standard symbols describe the relationship between a chunk of information contained within a file and an abstract characteristic.
Implicit symbols describe a characteristic that is implied by the file format. (All TIFF files are image, not sound files.)
Transient or processing symbols describe information which is needed to process the content of the file, do not provide any information which can be meaningfully transferred into an abstract format. (E.g. the start of an internal symbol table within a file.)

1.4.2.1. Abstract basic XCDL architecture

The abstract architecture of the XCDL is in many ways even simpler than that of the XCEL.

A XCDL description consists of four types of XML elements:

Elements describing properties of a file as a whole. (Width of an image.)
Elements containing the basic content of a file. (The bytestream representing an image or a sound sequence; the encoded bytestream or characterstream representing a text.)
Elements describing properties applicable to segments of these bytestreams. (Bytes nnn – mmm of a specific bytestream use the font xxx, for which a definition is contained elsewhere in the document.)
Elements describing relationships between sets of content carrying bytestreams and their properties . (Essentially links between independent parts of the content of a file.)

While this architecture is simpler, than that of the XCEL, its is much more directly related to the details of its realisation in the language. (Cf. part 3.)

1.4.2.2. Note on the scope of the XCDL architecture

A *complete* description of the content of a file in a language, which is independent of a specific file format, is obviously eventually an extremely general file format in itself. It will also be extremely difficult to generate XCEL descriptions for all existing file formats, which would allow their transmission into XCDL representations of the files encoded according to such file formats. As stated above, it is therefore envisaged and planned, that in actual usage many XCEL descriptions would represent only part of a complex format document.

To support such a partial translation of file formats into XCEL, while at the same time retaining the potential of the XCDLs as an abstract format for the storage of data, the XCDL allows for the consistent duplication of the data extracted from a file. The <data> element of an XCDL always gives the uninterpreted version of a bytestream from a file. For a format, which is fully defined by the XCEL, this is obviously redundant, as the same information will be available in a much more usable form in other elements, typically the <normData> element. The retention of <data> elements, however, provides a mechanism, which allows to extract into an XCDL a “semiunderstood” version of a file. Characteristics, which are described by the XCEL available at the time of conversion into XCDL, can be processed; others can simply be preserved by copying them into the <data> section. (For a detailed treatment of the relationship between <data> and <normData> see section 3.1.2.3. <data>/<normData> below.)

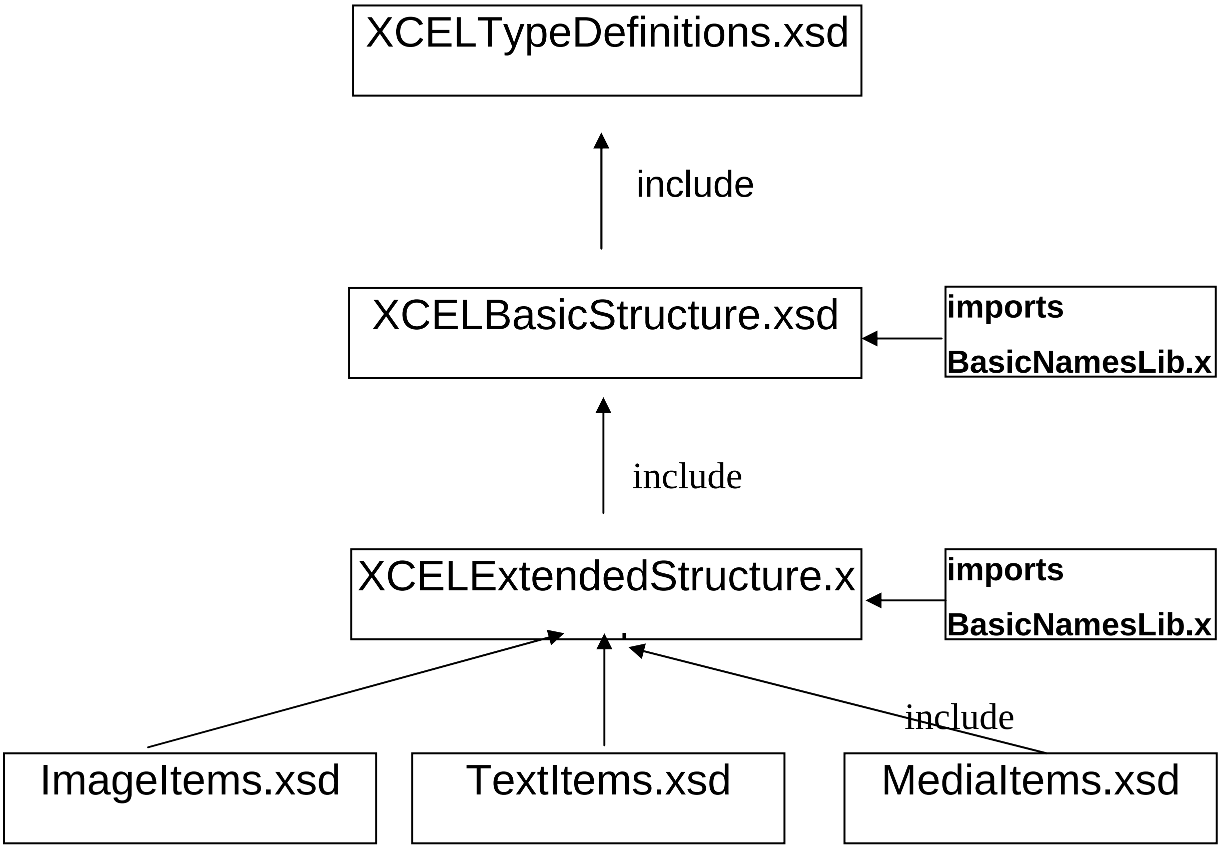
As this mechanism is central for possibilities to use XCDL descriptions for long term purposes themselves, it is discussed in detail in section 3.1. It is assumed that for all immediate purposes of the XCDL – extracting characteristics for the purpose of evaluating the information contained within a file – the <data> elements will typically not be created, leaving only the interpreted versions of the extracted information.

2. The eXtensible Characterisation Extraction Language – XCEL

2.1 Initial definitions: XCEL File Structure Documentation

<table>
<tr><th><em>FileName</em></th><th><em>Short Description</em></th></tr>
<tr><td>XCELTypeDefinitions.xsd</td><td>Defines complexType to express complex length or position statements.</td></tr>
<tr><td>XCELBasicStructure.xsd</td><td>Defines the structure of a XCEL Instance document.
Contains abstract Type definitions for items, symbols and properties.

XCELDocument
|
item (type=“itemType“)
|
subitem
|
item(type=“itemType“)
|
symbol(type=“symbolType“)
|
property (type=“propertyType“ derived from symbolType)</td></tr>
<tr><td>XCELExtendedStructure.xsd</td><td>defines extended definitions for items, symbols and properties</td></tr>
<tr><td>XCELImageItems.xsd</td><td>defines specific items, symbols and properties to describe image files.</td></tr>
</table>



File	***includes/imports***	***namespace***
XCELTypeDefinitions.xsd		td="http://www.planets-project.eu/xcl/schemata/xcelstructure“
XCELBasicStructure.xsd	<u>includes:</u> XCELTypeDefinitions.xsd <u>imports:</u> XCLBasicNamesLib.xsd	http://www.planets-project.eu/xcl/schemata/xcelstructure
XCELExtendedStructure.xsd	<u>includes:</u> XCELBasicStructure.xsd <u>imports:</u> XCLBasicNamesLib.xsd	http://www.planets-project.eu/xcl/schemata/xcelstructure
XCELImageItems.xsd	<u>includes:</u> XCELExtendedStructure.xsd <u>imports:</u> XCLImageNamesLib.xsd	http://www.planets-project.eu/xcl/schemata/xcelstructure

2.1.1. XCELTypeDefinitions.xsd

Defines complexTypes for XCEL-internal use, as for example lengthTypes or positionTypes.
Those can be derived to describe different situations found in various file formats.
At the moment there are three different positionTypes as well as two different lengthTypes employed while other types are outlined but not yet implemented.

1. positionTypes:
fixedPosition: The exact starting position of an item, given as a byte number in the file.
expectedPosition: The item is expected to start at the given byte number. If it is not found at the indicated position, the parser skips the item and will get back to it later on. The value of „expectedPosition“ is updated after each item the parser reads.
sequentialPosition: The item is expected to start at the current reading position of the parser.

2. lengthTypes:
fixedLength: The exact length of an item is given as a number of bytes.
referencedLength: The length of an item is given at another position in the file which is referenced by its ID.

2.1.2. XCELBasicStructure.xsd

Imports: NameLibs/XCLBasicNamesLib.xsd
Includes: XCELTypeDefinitions.xsd

1. item
Technical: Sequence of bytes.
Logical: a structuring unit, that splits a file into processing units and may have semantic meaning. The length of an item is either determined by the length of the incldued elements or by a range, that consists of a startposition and a length, both defined as one of the types mentioned above (needed for instance if the sub-item has the attribute multiple set to “true“ and the exact number of occurences is not known in advance).
An item might include the optional element override, which references one or more items, to be ignored and instead replaced by the information found at the current location.
Each item contains at least one sub-item, which is either a symbol, and therefore final or another item and as such divisible into smaller units.

2. symbol
Technical: Sequence of bytes.
Logical: Smallest unit for the reader to read.
It is always characterized by a range, indicating a position and length in the file and a series of attributes.
Position and Length correspond to the „types“ defined in TypeDefinitions.xsd.

A symbol might include the optional element override, which references one or more symbols, to be ignored and instead replaced by the information found at the current location.
Each symbol is of a certain symbolType, which can be derived in underlying schemas to attach format-specific properties to further describe its value.

3. property

A property is a specific kind of symbol, with the significant distinction that a property has a fixed value, which is the same for every file of the same file format.

4. attributes

attributes for items:

1. **ordered**: Determines wther the item's content is ordered, i.e. has to appear in the exact same order as defined in the XCEL or if the content within the item can follow in arbitrary sequence.
2. **identifier**: Unique Identifier, first item starting with "ID" the followings with a capital "I" for an item or „S“ for a symbol, followed by a number
3. **optional**: an optional item may or may not appear in the processed file. As a default every item is required, i.e. optional=false.
4. **multiple**: In case multiple is true, the symbol can reappear in other parts of the document. The default is false.
5. **byteOrder**: if not specified, the byteOrder is the same as in the enclosing item. has to be specified at least once in the first item.
6. **originalName**: name of the item or property in the file specification.

attributes for symbols

- **interpretation**: Instructions for the parser how to read the following n bytes, while 'n' is indicated by the element "length".
- **encoding**: The value's encoding given in the format specification. The default encoding is "decimal".

2.1.3. Extended Structure.xsd

Includes: XCELBasicStructure.xsd
Imports: NameLibs/XCLBasicNamesLib.xsd

extended symbol definitions

1. **basicSymbol**
   A simple symbol, which has a name characterising the file property which is going to be extracted, and the value it takes on.
   Contains element name, which allows names from the nm:xclBasicNameDefinitions namespace

2. **selectiveSymbol**
   A selectiveSymbol defines key/value pairs which represent an interpretation table for the value. The value the reader has found is then labelled by the corresponding keyName.
3. **dependentSymbol**
   A dependent symbol is interpreted only if the value of the referred file property (referenceName) corresponds to the given referenceValue.

<u>extended item definitions</u>

basicItem
This is a pre-defined item for all items with names from the basicNameDefinition namespace.
Contains a name element, which allows name from the nm:xclBasicNameDefinitions namespace
Names for items are optional.

<u>extended property definitions</u>

basicProperty
This is a pre-defined property for all properties with names from the basicNameDefinition namespace.
Has a name element, which allows name from the nm:xclBasicNameDefinitions namespace

<u>specific item definitions</u>

1. **fileHeaderItem**
   pre-defined item, which designates the item as containing file header data. for parsing purposes only.
2. **fileDataItem**
   pre-defined item, which designates the item as a cohesive data block. for parsing purposes only.

<u>specific symbol definitions</u>

1. **dataLengthSymbol**
   pre-defined symbol, which indicates that the following value has to be interpreted as a length statement. for parsing purposes only.
2. **dataTypeSymbol**
   pre-defined symbol, which indicates that the following value has to be interpreted as a unique name defined in the specification. for parsing purposes only.

<u>specific property definitions</u>

1. **dataLengthProperty**
   pre-defined property, which indicates that the following value has to be interpreted as a length

statement. Differs from dataLengthSymbol only with respect to its fixed value. For parsing purposes only.

2. **dataTypeProperty**
   pre-defined property, which indicates that the following value has to be interpreted as a unique name defined in the specification. Differs from dataLengthSymbol only with respect to its fixed value. For parsing purposes only.

2.1.4. XCELImageItems.xsd

imports: "../NameLibs/XCLImageNamesLib.xsd"
includes: ../XCELExtendedStructure.xsd

<u>extended symbol definitions to describe image files</u>

1. **basicImageSymbol**
   A symbol which is derived from „symbolType“ and contains an element <name>, which allows names from the img:xclImageNameDefinitions namespace.
2. **selectiveImageSymbol**
   A symbol which is derived from „selectiveSymbol“.
   Contains an element <valueinterpretation> which allows names from the img:xclValueLabelsImage and an element <name>, which follows the name conventions of the img:xclImageNameDefinitions namespace.
3. **dependentImageSymbol**
   A symbol which is derived from „dependentSymbol“.
   Contains an element <referencename> and an element <name>, which both allow names of the img:xclImageNameDefinitions namespace.

<u>extended item definitions to describe image files</u>

imageItem
An imageItem has a name-element which can adapt any name from the nm:xclImageNameDefinitions namespace , which, in turn, can build (more complex) semantic units.

2.2. XCELTypeDefinitions.xsd

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xlink="http://www.w3.org/1999/xlink"
      targetNamespace="http://www.planets-
project.eu/xcl/schemata/xcelstructure"
      xmlns:td="http://www.planets-project.eu/xcl/schemata/xcelstructure"
      xmlns:nm="http://www.planets-project.eu/xcl/schemata/xclnames"
      elementFormDefault="qualified" version="0" xml:lang="en">

      <xsd:import schemaLocation="nameLibs/XCLImageNamesLib.xsd"
namespace="http://www.planets-project.eu/xcl/schemata/xclnames"/>


      <xsd:complexType name="positionType"
abstract="true"></xsd:complexType>
      <xsd:complexType name="lengthType" abstract="true"></xsd:complexType>
      <xsd:complexType name="valueType" abstract="true"></xsd:complexType>

      <xsd:complexType name="rangeType" >
            <xsd:sequence>
                  <xsd:element name="startposition"
type="td:positionType"></xsd:element>
                  <xsd:element name="length"
type="td:lengthType"></xsd:element>
            </xsd:sequence>
      </xsd:complexType>

      <xsd:element name="sum">
            <xsd:complexType>
                  <xsd:sequence>
                        <xsd:element name="operand"
maxOccurs="unbounded"></xsd:element>
                  </xsd:sequence>
            </xsd:complexType>
      </xsd:element>
      <xsd:element name="subtract">
            <xsd:complexType>
                  <xsd:sequence>
                        <xsd:element name="operand"
maxOccurs="unbounded"></xsd:element>
                  </xsd:sequence>
            </xsd:complexType>
      </xsd:element>

      <!-- **************************** position types
***************************** -->

      <xsd:complexType name="fixedPosition">
            <xsd:annotation>
                  <xsd:documentation>
                        The exact starting position of an item, given as a
byte number in the file.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent mixed="true">
                  <xsd:extension base="td:positionType">
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name="sequential">
            <xsd:annotation>
                  <xsd:documentation>
                        The item is expected to start at the current
reading position of the parser.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent mixed="true">
                  <xsd:extension base="td:positionType">
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name="expected">
            <xsd:annotation>
                  <xsd:documentation>
                        The item is expected to start at the given byte
number.
                        If it is not found at the indicated position, the
parser skips the item and will get back to it later on.
                        The value of â□□expectedPositionâ□œ is updated
after each item the parser reads.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent mixed="true">
                  <xsd:extension base="td:positionType">
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name="calcPosition">
            <xsd:complexContent mixed="true">
                  <xsd:extension base="td:positionType">
                        <xsd:choice>
                              <xsd:element ref="td:sum"/>
                              <xsd:element ref="td:subtract"/>
                        </xsd:choice>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
      <!-- ******************************************** length Types
****************************************************-->
      <xsd:complexType name="fixedLength">
            <xsd:annotation>
                  <xsd:documentation>
                        fixedLength: The exact length of an item is given
as a number of bytes.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent mixed="true">
                  <xsd:extension base="td:lengthType">
                        <xsd:sequence>
                              <xsd:element ref="td:xlink" minOccurs="0"/>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name="calcLength">
            <xsd:complexContent mixed="true">
                  <xsd:extension base="td:lengthType">
                        <xsd:choice>
                              <xsd:element ref="td:sum"/>
                              <xsd:element ref="td:subtract"/>
                        </xsd:choice>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name="relatedLength">
            <xsd:complexContent mixed="true">
                  <xsd:extension base="td:lengthType">
                        <xsd:sequence>
                              <xsd:element name="reference"
maxOccurs="unbounded" minOccurs="0">
                                    <xsd:complexType>
                                          <xsd:sequence>

                                                <xsd:element
name="value"></xsd:element>
                                                <xsd:element
name="referenceName" type="xsd:string"></xsd:element>
                                                <xsd:element
name="referenceValue"></xsd:element>
                                          </xsd:sequence>
                                    </xsd:complexType>
                              </xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name="undefinedLength" mixed="true">
            <xsd:complexContent mixed="true">
                  <xsd:extension base="td:lengthType">
                        <xsd:sequence>
                              <xsd:element name="minValue" minOccurs="0"/>
                              <xsd:element name="maxValue" minOccurs="0"/>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name="referencedLength" mixed="true">
            <xsd:annotation>
                  <xsd:documentation>
                        The length of an item is given at another position
in the file which is referenced by its ID.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent mixed="true">
                  <xsd:extension base="td:lengthType">
                        <xsd:attribute name="ref"
type="xsd:IDREF"></xsd:attribute>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>

      <!-- ******************************************** value Types
*****************************************************-->
      <xsd:complexType name="valueList">
            <xsd:annotation>
                  <xsd:documentation>
                        List of valid Values
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent mixed="true">
                  <xsd:extension base="td:valueType">
                        <xsd:sequence>
                        <xsd:element name="keyValue"
maxOccurs="unbounded"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>

      <xsd:complexType name="valueRange">
            <xsd:annotation>
                  <xsd:documentation>
                        List of valid Values
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent mixed="true">
                  <xsd:extension base="td:valueType">
                        <xsd:sequence>
                              <xsd:element name="startRange"
maxOccurs="unbounded"></xsd:element>
                              <xsd:element name="endRange"
maxOccurs="unbounded"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>

      <!-- property Names -->
      <xsd:complexType name="propertyNames"
abstract="true"></xsd:complexType>
      <xsd:complexType name="images">
            <xsd:annotation>
                  <xsd:documentation>
                        List of valid Values
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent mixed="true">
                  <xsd:extension base="td:propertyNames">
                        <xsd:sequence>
                              <xsd:element name="value"></xsd:element>
                              <xsd:element name="label"
type="nm:xclImageNamesDefinitions"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>

      <xsd:element name="xlink">
            <xsd:complexType>
                  <xsd:sequence>
                        <xsd:any namespace="http://www.w3.org/1999/xlink"
processContents="skip" maxOccurs="unbounded"/>
                  </xsd:sequence>
            </xsd:complexType>
      </xsd:element>

      <xsd:simpleType name="fileStructuringNames">
            <xsd:restriction base="xsd:string">
                  <xsd:enumeration value="dataBlock">
                        <xsd:annotation>
                              <xsd:documentation>A sequence of bytes that
uniquely identifies a certain fileformat</xsd:documentation>
                        </xsd:annotation>
                  </xsd:enumeration>
                  <xsd:enumeration value="fileHeader">
                        <xsd:annotation>
                              <xsd:documentation>A sequence of bytes that
uniquely identifies a certain fileformat</xsd:documentation>
```

```
                        </xsd:annotation>
                  </xsd:enumeration>
                  <xsd:enumeration value="dataBlockType">
                        <xsd:annotation>
                              <xsd:documentation>A sequence of bytes that
uniquely identifies a certain fileformat</xsd:documentation>
                        </xsd:annotation>
                  </xsd:enumeration>
                  <xsd:enumeration value="dataBlockLength">
                        <xsd:annotation>
                              <xsd:documentation>The length in bytes of the
complete file</xsd:documentation>
                        </xsd:annotation>
                  </xsd:enumeration>
            </xsd:restriction>
      </xsd:simpleType>

</xsd:schema>
```

2.3. XCELBasicStructure.xsd

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns="http://www.planets-project.eu/xcl/schemas/xcelstructure"
      xmlns:nm="http://www.planets-project.eu/xcl/schemas/xclnames"
      targetNamespace="http://www.planets-
project.eu/xcl/schemas/xcelstructure"
      elementFormDefault="qualified">

      <xsd:include schemaLocation="XCELTypeDefinitions.xsd"></xsd:include>
      <xsd:import schemaLocation="nameLibs/XCLBasicNamesLib.xsd"
namespace="http://www.planets-project.eu/xcl/schemas/xclnames"/>


      <xsd:element name="XCELDocument">
            <xsd:annotation>
                  <xsd:documentation>
                  Basic Structure for XML-Instances based on the eXtensible
Characterization Extraction Language.
                  Each XCEL-Instance is structured into a sequence of
items, which can be processed as containers of ordered or unordered
content. </xsd:documentation>
            </xsd:annotation>
            <xsd:complexType>
                  <xsd:sequence>
                        <xsd:element ref="item" maxOccurs="unbounded"/>
                  </xsd:sequence>
            </xsd:complexType>
      </xsd:element>

      <xsd:element name="item" type="itemType">
            <xsd:annotation>
                  <xsd:documentation>
                  Technical: Sequence of bytes.
                  Logical: a structuring unit, that splits a file into
processing units and may have semantic meaning.
                  The length of an item is determined by the length of the
included elements.
                  Each item contains at least one sub-item, which is either
a symbol, and therefore final or another item and as such divisible into
smaller units.
```

```
                  </xsd:documentation>
            </xsd:annotation>
      </xsd:element>
      <xsd:element name="symbol"  type="symbolType">
            <xsd:annotation>
                  <xsd:documentation>
                        Technical: Sequence of bytes.
                        Logical: Smallest unit for the reader to read.
                        It is always characterized by a range, indicating a
position and length in the file and a series of attributes.
                        Each symbol is of a certain symbolType, which can
be derived in underlying schemas to attach format-specific properties to
further describe its value.
                  </xsd:documentation>
            </xsd:annotation>
      </xsd:element>
      <xsd:element name="property"  type="propertyType">
            <xsd:annotation>
                  <xsd:documentation>
                        A property is a specific kind of symbol, with the
significant distinction that a porperty has a fixed value, which is the
same for every file of the same file format.
                  </xsd:documentation>
            </xsd:annotation>
      </xsd:element>

      <!--************************************************* SubItem Group
****************************************-->
      <xsd:group name="subItem">
            <xsd:choice>
                  <xsd:element ref="item" minOccurs="0"
maxOccurs="unbounded"/>
                  <xsd:element ref="symbol" minOccurs="0"
maxOccurs="unbounded"/>
                  <xsd:element ref="property"  minOccurs="0"
maxOccurs="unbounded"/>
            </xsd:choice>
      </xsd:group>

      <!--*************************************************Item Type
Definition ****************************************-->
      <xsd:complexType name="itemType" abstract="true">
            <xsd:sequence>
                  <xsd:element name="range" type="rangeType"
minOccurs="0"/>
                  <xsd:element name="override" maxOccurs="unbounded"
minOccurs="0">
                        <xsd:complexType>
                              <xsd:attribute name="ref"
type="xsd:IDREF"></xsd:attribute>
                        </xsd:complexType>
                  </xsd:element>
                  <xsd:group ref="subItem" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attributeGroup ref="itemAtts"></xsd:attributeGroup>
      </xsd:complexType>

      <!--*************************************** symbol Type Definition -
is abstract and has to be derived in underlying schemas
****************************************-->
      <xsd:complexType name="symbolType" abstract="true">
            <xsd:sequence>
```

```
                  <xsd:element name="range" type="rangeType"
minOccurs="1"/>
                  <xsd:element name="override" maxOccurs="1" minOccurs="0">
                        <xsd:complexType>
                              <xsd:sequence>
                                    <xsd:element name="itemRef"
maxOccurs="unbounded"/>
                              </xsd:sequence>
                        </xsd:complexType>
                  </xsd:element>
            </xsd:sequence>
            <xsd:attributeGroup ref="itemAtts"></xsd:attributeGroup>
            <xsd:attributeGroup ref="symbolAtts"></xsd:attributeGroup>
      </xsd:complexType>

      <!--*************************************** property - has a range
and a fixed value  ****************************************-->
      <xsd:complexType name="propertyType" abstract="true">
            <xsd:complexContent>
                  <xsd:extension base="symbolType">
                        <xsd:sequence>
                              <xsd:element name="value"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>

      <!-- ************************************************** attributes for
structure element 'item'
***************************************************************************
-->
      <xsd:attributeGroup name="itemAtts">
      <xsd:attribute name="ordered" type="xsd:boolean" use="optional"
default="true">
            <xsd:annotation>
                  <xsd:documentation>
                        Determines wther the item's content is ordered,
i.e. has to appear in the exact  same order as defined in the XCEL or if
the content within the item can follow in arbitrary sequence.
                        </xsd:documentation>
            </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="identifier" type="xsd:ID" use="required">
            <xsd:annotation>
                  <xsd:documentation>
                        Unique Identifier, first item starting with "ID"
the following with a capital "I" followed by a number</xsd:documentation>
            </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="optional" type="xsd:boolean" use="optional"
default="false">
            <xsd:annotation>
                  <xsd:documentation>
                        an optional item may or may not appear in the
processed file. As a default every item is required, i.e.
optional=false.</xsd:documentation>
            </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="multiple" type="xsd:boolean" use="optional"
default="false">
            <xsd:annotation>
                  <xsd:documentation>
```

```
                        In case multiple is true, the symbol can reappear
in other parts of the document. The default is false.</xsd:documentation>
            </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="implicit" type="xsd:boolean" use="optional"
default="false">
            <xsd:annotation>
                  <xsd:documentation>
                        In case multiple is true, the symbol can reappear
in other parts of the document. The default is false.
                  </xsd:documentation>
            </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="byteOrder" type="nm:byteOrder" use="optional">
            <xsd:annotation>
                  <xsd:documentation>
                        if not specified, the byteOrder is the same as in
the enclosing item. has to be specified at least once in the first item.
                  </xsd:documentation>
            </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="originalName" use="optional">
            <xsd:annotation>
                  <xsd:documentation>name of the property in the file
specification</xsd:documentation>
            </xsd:annotation>
      </xsd:attribute>
      </xsd:attributeGroup>

      <!-- ************************************************ attributes
that define a value
****************************************************************************
-->
      <xsd:attributeGroup name="symbolAtts">
            <xsd:annotation>
                  <xsd:documentation>attributes for the value element in
property or derived symbols</xsd:documentation>
            </xsd:annotation>
            <xsd:attribute name="interpretation"
type="nm:xclDataTypeDefinition" default="uint8">
                  <xsd:annotation>
                        <xsd:documentation>
                              Instructions for the parser how to read the
following n bytes, while 'n' is indicated by the element "length".
                        </xsd:documentation>
                  </xsd:annotation>
            </xsd:attribute>
            <xsd:attribute name="encoding" type="nm:xclDataTypeDefinition"
default="decimal">
                  <xsd:annotation>
                        <xsd:documentation>
                              The value's encoding given in the format
specification. The default encoding is "decimal".
                        </xsd:documentation>
                  </xsd:annotation>
            </xsd:attribute>
      </xsd:attributeGroup>
</xsd:schema>
```

2.4. XCELExtendedStructure.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns="http://www.planets-project.eu/xcl/schemata/xcelstructure"
      xmlns:nm="http://www.planets-project.eu/xcl/schemata/xclnames"
      targetNamespace="http://www.planets-
project.eu/xcl/schemata/xcelstructure"
      elementFormDefault="qualified">

      <xsd:include schemaLocation="XCELBasicStructure.xsd"></xsd:include>
      <xsd:import schemaLocation="nameLibs/XCLBasicNamesLib.xsd"
namespace="http://www.planets-project.eu/xcl/schemata/xclnames"/>

      <!-- ************************************************* extended
symbol definitions ************************************************** -->

      <xsd:complexType name="selectiveSymbol" abstract="true">
            <xsd:annotation>
                  <xsd:documentation>
                        A selectiveSymbol defines key/value pairs which
represent an interpretation table for the value. The value the reader has
found is then labelled by the corresponding keyName.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent>
                  <xsd:extension base="symbolType">
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>

      <xsd:complexType name="dependentSymbol" abstract="true">
            <xsd:annotation>
                  <xsd:documentation>
                        A dependent symbol is interpreted only if the value
of the referred file property (referenceName) corresponds to the given
referenceValue.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent>
                  <xsd:extension base="symbolType"></xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>

      <!-- ************************************************* specific
symbol definitions for basicTypes
************************************************** -->

      <xsd:complexType name="basicSymbol">
            <xsd:annotation>
                  <xsd:documentation>
                        A simple symbol, which has a name characterising
the file property which is going to be extracted, and the value it takes
on.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent>
                  <xsd:extension base="symbolType">
                        <xsd:sequence>
                              <xsd:element name="name"
type="nm:xclBasicNameDefinitions"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name="fileStructureSymbol">
            <xsd:complexContent>
                  <xsd:extension base="symbolType">
                        <xsd:sequence>
                              <xsd:element name="name"
type="fileStructuringNames"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>


      <!-- ************************************************* item
definitions ************************************************** -->

      <xsd:complexType name="basicItem">
            <xsd:annotation>
                  <xsd:documentation>
                        this is a pre-defined property for all properties
with names from the basicNameDefinition namespace.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent>
                  <xsd:extension base="itemType">
                        <xsd:sequence>
                              <xsd:element name="name"
type="nm:xclBasicNameDefinitions" minOccurs="0"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>

      <xsd:complexType name="fileStructureItem">
            <xsd:complexContent>
                  <xsd:extension base="itemType">
                        <xsd:sequence>
                              <xsd:element name="name"
type="fileStructuringNames"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>

      <!-- ************************************************* definition of
a basic property ************************************************** -->
      <xsd:complexType name="basicProperty">
            <xsd:annotation>
                  <xsd:documentation>
                        this is a pre-defined property for all properties
with names from the basicNameDefinition namespace.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent>
                  <xsd:extension base="propertyType">
                        <xsd:sequence>
                              <xsd:element name="name"
type="nm:xclBasicNameDefinitions"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name="fileStructureProperty">
            <xsd:complexContent>
                  <xsd:extension base="propertyType">
                        <xsd:sequence>
                              <xsd:element name="name"
type="fileStructuringNames"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>


      <xsd:complexType name="implicitItem">
            <xsd:complexContent>
                  <xsd:extension base="itemType">
                        <xsd:sequence>
                              <xsd:element name="name"
type="propertyNames"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
</xsd:schema>
```

2.5. XCELImageItames.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns="http://www.planets-project.eu/xcl/schemata/xcelstructure"
      xmlns:nm="http://www.planets-project.eu/xcl/schemata/xclnames"
      targetNamespace="http://www.planets-
project.eu/xcl/schemata/xcelstructure"
      elementFormDefault="qualified">


      <xsd:import schemaLocation="../nameLibs/XCLImageNamesLib.xsd"
namespace="http://www.planets-project.eu/xcl/schemata/xclnames"/>
      <xsd:include
schemaLocation="../XCELExtendedStructure.xsd"></xsd:include>

      <!-- ************************************************ specific
symbol definitions for imageTypes
************************************************* -->

      <xsd:complexType name="basicImageSymbol">
            <xsd:complexContent>
                  <xsd:extension base="symbolType">
                        <xsd:sequence>
                              <xsd:element name="name"
type="nm:xclImageNamesDefinitions">
                                    <xsd:annotation>
                                          <xsd:documentation>
                                                Name of the file property to
be extracted.
                                          </xsd:documentation>
                                    </xsd:annotation>
                              </xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>

      <xsd:complexType name="selectiveImageSymbol">
            <xsd:complexContent>
                  <xsd:extension base="selectiveSymbol">
                        <xsd:sequence>
                              <xsd:choice>
                                    <xsd:element name="validValues"
minOccurs="1" type="valueType"></xsd:element>
                                    <xsd:element name="valueInterpretation"
maxOccurs="1" minOccurs="1">
                                          <xsd:complexType>
                                                <xsd:sequence
maxOccurs="unbounded">
                                                      <xsd:element
name="valueLabel" type="nm:xclImageValueLabels">
                                                            <xsd:annotation>

      <xsd:documentation>
                                                                        The
label, the value found at keyValue is associated with.
                                                                        The
parser compares the value found for the current symbol with the one given
as keyVaue and choses the

      appropriate label for the file property.

      </xsd:documentation>

      </xsd:annotation>
                                                      </xsd:element>
                                                      <xsd:element
name="keyValue">
                                                            <xsd:annotation>

      <xsd:documentation>

      Encoded file property characteristic, which is interpretated by
keyName.

      </xsd:documentation>

      </xsd:annotation>
                                                      </xsd:element>
                                                </xsd:sequence>
                                          </xsd:complexType>
                                    </xsd:element>
                              </xsd:choice>
                              <xsd:element name="name"
type="nm:xclImageNamesDefinitions"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>

      <xsd:complexType name="dependentImageSymbol">
            <xsd:complexContent>
                  <xsd:extension base="dependentSymbol">
                        <xsd:sequence>
                              <xsd:element name="reference"
maxOccurs="unbounded" minOccurs="1">
                                    <xsd:complexType>
```

```
                                          <xsd:sequence>

                                                <xsd:element
name="referenceName" type="nm:xclImageNamesDefinitions">
                                                      <xsd:annotation>

      <xsd:documentation>
                                                                  The Name
of the file property, of which the value interpretation of the current file
property is dependent of.

      </xsd:documentation>
                                                      </xsd:annotation>
                                                </xsd:element>
                                                <xsd:element
name="referenceValue">
                                                      <xsd:annotation>

      <xsd:documentation>
                                                                  If the
referenceValue corresponds with the value for the file property named by
referenceName, the value read
                                                                  by the
parser is interpreted by this symbol. Otherwise the parser looks at the
next symbol for a correct interpretation.

      </xsd:documentation>
                                                      </xsd:annotation>
                                                </xsd:element>
                                          </xsd:sequence>
                                    </xsd:complexType>
                              </xsd:element>
                              <xsd:element name="name"
type="nm:xclImageNamesDefinitions"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>


      <!-- **************************************************  item
definitions  ************************************************** -->

      <xsd:complexType name="imageItem">
            <xsd:annotation>
                  <xsd:documentation>
                        this is a pre-defined property for all properties
with names from the basicNameDefinition namespace.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent>
                  <xsd:extension base="itemType">
                        <xsd:sequence>
                              <xsd:element name="name"
type="nm:xclImageNamesDefinitions" minOccurs="0"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>

      <xsd:complexType name="implicitImageItem">
            <xsd:annotation>
                  <xsd:documentation>
```

```
                        this is a pre-defined property for all properties
with names from the basicNameDefinition namespace.
                  </xsd:documentation>
            </xsd:annotation>
            <xsd:complexContent>
                  <xsd:extension base="itemType">
                        <xsd:sequence>
                              <xsd:element name="name"
type="nm:xclImageNamesDefinitions" minOccurs="0"></xsd:element>
                        </xsd:sequence>
                  </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>


</xsd:schema>
```

3. The eXtensible Characterisation Definition Language XCDL

3.1 Initial Definitions[2]

3.1.1. Introduction

The XCDL is designed with respect to the overall goal to provide a means for describing digital objects. The underlying technology is primarily XML and XML schema which are the technical backbone of the language. The realisation of the XCDL is the instantiation of the XCDL (the XCDL document) applied to a digital object.
Digital objects are characterised through certain attributes, the properties of the digital objects. A property has always a dedicated value. Each XCDL document therefore describes digital objects through the specification of its properties values.

According to its OAIS definition [OAIS02], a digital object is composed of a set of bit sequences. Three items are always appropriate for digital objects as well:
(1) The set of bit sequences is always manifest, i.e. physically located on a storage medium. Operating systems usually organize bit sequences as files on the storage medium.
(2) Bit sequences organized as files are subject to a specific formatting, determined by the file format specification (primary format). This is nothing but a set of interpretation instructions to be applied to the bit sequences. The set of properties for a digital object is also determined by the format specification.3
(3) Parts of the bit sequences interpreted by the constraints of the primary file format may be subject to additional interpretation rules (secondary format). This is always the case for embedded objects (i.e. object within an object).

A digital object described by a XCDL document shall be understood under these presumptions.

There a some consequences resulting from this concerning the structure of a digital object and its representation through the XCDL:
(1) A digital object can be physically distributed, i.e. the bit sequences not necessarily need to be organized in one single file, but may be located in separate files. This is solved through the <composition> element set.
(2) Every sequence of bits formatted by file format X can be captured as a holistic information unit. This is solved through the <object> element set.
(3) Sequences of bits can be subject to more than just one single format, i.e. underlie different format specifications (or more general: rules for interpretation). This is solved through the possibility to relate objects with each other using the <objRel> element.

[2] Compliance for section 3.: ‘shall’ and ‘must’ refer to mandatory requirements. ‘should’ refers to recommended items and ‘may’ refers to optional items.

[3] For the possibility to include properties which do not base on the format description or the accordant XCEL description see section 3.1.2.3. <property>

As a result of this, from the viewpoint of the XCDL, a digital object is a sequence of bits, subject to X sets of interpretation rules (commonly called ‘format’ or ‘formatting’) organized as one or many files.

For example, a sequence of bits can be stored in a single file, formatted as ‘text’ as its primary underlying file format. This digital object can be further decoded by applying another set of interpretation rules which could be defined through the WWW 3C XML standard. (for more details on relationships of objects see sections 3.1.2.2 and 3.1.2.3. <dataRef>)
Besides this, in practice certain digital objects can only be interpreted in a meaningful way if they are taken as an entity. Examples for this are the shape file format [ESRI98] for geographical data or the new office open XML format [MS05]. In such cases, several digital objects, distributed on several files, potentially with different primary formats, are seen as a single logical unit. The XCDL also provides a solution to cope with such digital entities through applying the <composition> element set.

3.1.2. XCDL Core Schema Elements

The core XCDL schema elements are included within the ‘XCDLCore.xsd’ file.

3.1.2.1. root element <xcdl>

Every XCDL document starts with the <xcdl> root element set that wraps optionally the <composition> child tag set and at least one <object> element set. A <xcdl> element shall have an identification number (attribute: id). The schema allows any string or integer number to be validated for this ID. It is recommended to use unique identifiers. At least the ID should be unique in a certain namespace to enable extensive use of XCDL functionalities (see 2.2).

3.1.2.2. element <composition> and child elements

<composition>, <xcdlRef>, <obj>

If there is a need to distribute digital objects on various XCDL documents (e.g., large objects may be split on various XCDL documents for the purpose of improved processing) the use of the <composition> element is indicated for consistent binding. Note that this element can be used unbounded in each XCDL document, depending on the number of digital objects it is part of.
Example: According to its specification, PNG [PNG03] files may contain an embedded ICC profile. Such a specific ICC profile can be described as a discrete digital object (using the <object> element). Every other digital object that embeds it too can then relate to this modularized object via the <composition> structure (given that the XCDL document provides an identification number that is unique in a certain namespace).
Within the <composition> tag set, there is only one child allowed. This is the <xcdlRef> element. It is created to wrap and combine the single partial objects which are referenced to by the <obj> element. It provides an attribute ‘id’ for inserting the appropriate objects reference ID. <xcdlRef> itself provides this attribute for referencing to the appropriate ID of the referenced XCDL document.

The <composition> element must be used if there is an actual relation of parts of a digital object on the data level (see section 3.1.2.3, <dataRef>) or between digital objects (e.g., embedded objects), otherwise such relations can not be retraced. In such a case, any XCDL builder shall create the element automatically.

The element is also designed to be used to describe logical units (see section 3.1.1). In this case, the attribute ‘use’ shall be used and set on value ‘logical’ to indicate such a relation. This should be done via an interface for presetting the XCDL output

```
<xcdl ... id=”A123456”>
      <composition>
            <xcdlRef id=”A123457”>
                  <obj id=”1”/>
            </xcdlRef>
            <xcdlRef id=”A123458”>
                  <obj id=”4”/>
            </xcdlRef>
      </composition>
      <object id=”1” ...>
      ...
</xcdl>

<xcdl ... id=”A123457”>
      <object id=”1”>...</object>
</xcdl>

<xcdl ... id=”A123458”>
      <object id=”4”>...</object>
</xcdl>
```

Ex. XCDL-1: Usageof the <composition> element

Example XCDL-1 shows a <composition> structure. The XCDL document with id=”A123456” uses two objects from other XCDL documents (“A123457” and “A123458”). The objects are referenced and identified through their ID provided within the <object> elements. The whole digital object is composed of three objects: The one (id=“1”) that is part of “A123456”and the others which are part of “A123457”, id=”1” and “A123458”, id=”4”. Note: Applying this structure enables to widely distribute digital objects and to describe them by redundancy. The single XCDL documents can be distributed not only locally within a single repository of an archive X or a library Y but also can be distributed on several repositories of different archives Z. As a consequence, digital objects can be described using references to redundant stored objects. This may be helpful to guarantee long-term access to digital objects For more details on the objects ID, see the next section.

3.1.2.3. element <object> and child elements

<object>

Every XCDL document shall include at least one digital object mapped to the <object> element. For all <object> elements an identification number (attribute “id”) is required. Note that this ID only shall be unique within the XCDL document if the wrapping <xcdl> element has a unique ID.

The <object> element has two child elements, <data> and <normData> for optional usage and a third one with required usage, the <property> element.

The attribute “natFormat” is dedicated to provide information about the native[4] format. The designation should be a discrete label from a predefined vocabulary. For this version of the XCDL, the PRONOM identifiers [PRO06] for file formats are recommended.

<data>

The <data> element wraps the full data of the source object. By default, character data shall be encoded in UTF-16, binary data shall be represented as hexadecimal encoded binary data. For relation and reference purposes, an identification number (attribute “id”) is required. Note that this ID shall only be unique within the XCDL document. If there are relations in terms of a <composition> structure, the IDs of the XCDL documents (<xcdl> element) should additionally be unique within a certain namespace.

Besides others[5], wrapping the full source objects data may be useful for data based references of properties (see this section, element <property>).

<normData>

This repeatable element wraps the source data in a context-free representation (normalised to a standard representation). All byte sequences which relate to specific properties or which appear in an encoded representation are removed or decoded to the standard UTF-16 representation (also required for <normData>). So normalised data is an abstraction of format internal representation of content data. The representation as normalised data is primarily important for an exact designation of property to data relations as references on normalised data (see this section, element <dataRef>).

The <normData> element provides two attributes. An identification number (attribute ‘id’) is required, again to realize relational functionality, i.e. to enable references of property values to parts of the data. Attribute ‘type’ is optional. If it is used it indicates the type of information the data represents. For details see section 3.1.3 on the basic XCDL types.

Note that if the normalised data is splitted on more than one <normData> element, the IDs shall reflect the original structure (i.e. order) of the data. Moreover, if there is a reference to normalised data, the entire data shall be normalised by the XCDL builder in order to guarantee the functionality of the referencing mechanism (see element <dataRef>for more details).

The normalisation is done by a designated tool (software module) , part of the entire extensible characterisation language processing software.

- This is a **text.**

```
<data id=”1”>
{\rtf1\ansi\ansicpg1252\deff0\deflang1031{\fonttbl{\f0\fswiss\fcharset0
Arial;}}\viewkind4\uc1\pard\f0\fs20 \bullet  This is a   \b text\b0 .\par}
</data>

<normData id=”1” type=”text”>
 • This is a text.
</normData>
```

[4] identical terms used in this section (3.) are: original, source.

[5] see section 1.4.2.2

Ex. XCDL-2: Data and normalised data compared

Example XCDL-2 presents the data from a source object wrapped as a whole in the <data> element and as normalised data in the <normData> element. The source object is formatted in rich-text (RTF, second representation in the example). RTF [RTF01] marks up text with control words which encode a specific property to be applied to a certain sequence of bytes (characters).

In this example, 'text' is formatted as bold, i.e. the rendered object should appear with the 'text' string put in bold letters (first representation). Additionally a bullet is encoded in RTF with control word '\bullet'.The normalised string representation abstracts from this internal format specific representation. Any other file that contains exactly the same string but may be described in a different format also uses different controllings for internal format-specific representation. This difference is compensated in the normalised representation. For the purpose of the exact correlation of properties to designated content representing strings (see <dataRef>) and for the purpose of comparing digital objects which come from diverse native formats, normalisation of data is extremely advantageous.

Note that neither <data> nor <normData> are defined as required elements.

<property>

Properties are the core unit through which a digital object can be characterised. The XCDL maps properties with the <property> element. Three attributes are assigned: Again, an identification number (attribute 'id') is required for relational ends. The attributes 'source' and 'cat' are intended for optional use.

'Source' indicates the source the property is derived from. In most cases this will be value 'raw', means derived from the source object data. Value 'implicit' means the property is not fixed to the source object data but derived from the source objects format specification. Attribute 'extended' is assigned if there is no derivation from the source object or the source objects specification. Finally 'added' is dedicated for properties which have been inserted at a later date, deviating from the date of initial creation. The "last" value could be useful in case of reconstructed objects.

Attribute 'cat' can optionally be activated. If so, it indicates the category of the property. In the current version there are four possible values:

'descr': descriptive property, i.e. occurrence of object describing property

'hist': history property, i.e. property that may appear in a different shape in the source object which may be resolved in the XCDL document (e.g., compressed data)

'cont': content property, i.e. relating directly to a byte sequence

'extern': property that refers to external item, i.e. not related to objects data, e.g., software and hardware used to create the object.

Note that through applying the last value it is also possible to designate and include properties which are not derived from the source objects data or format specification. In this case, attribute 'source' should be set to value 'external' (cp. first footnote in section 3.).

<name>

Every property has a unique name, an intellectual identifier. It assigns a specific meaning to the property. All unique property names are defined (i.e. the functionality of the property precisely maps to a unique name) in external schemas which are subsumed as so called ‘name libraries’ (NameLibs). These libraries are intended to be expanded over time according to the increase of available XCEL descriptions of file formats. Both languages, the XCDL as well as the XCEL share their definitions. (for more details on this see the section discussing the name libraries).

The attribute ‘alias’ can be activated to assign the original name of the property as specified in its format specification. In the current version, no defined vocabulary exists. The values are taken from the XCEL descriptions as they appear there.

```
<property id="2" source="raw" cat="descr">
      <name alias="length">height</name>
      ...
</property>
```

```
<!—the corresponding definition entry in ‘XCLImageNamesLib.xsd: -->
...
<xs:enumeration value="width">
      <xs:annotation>
            <xs:documentation>Width of an image. Corresponds
                     to
 the horizontal dimension(x-
 axis)
            </xs:documentation>
      </xs:annotation>
</xs:enumeration>
```

Ex. XCDL-3: Usage of <property> and <name> elements

Example XCDL-3 shows a XCDL structure describing an image property named ‘height’, indicating the vertical dimension of an image. It is derived from the original objects data and categorised as a descriptive property. The source format specification terms this property ‘length’. The definition (reflecting the meaning) of this property is part of the name library ‘XCLImageNamesLib.xsd’.

<valueSet>

Properties are primarily described by their assigned values. The XCDL provides two elements each created to take up different kinds of value representations: Raw values (<rawVal>) and labelled values (<labVal>). The element <valueSet> wraps these values.

Every value set shall have an identification number (attribute ‘id’). There also shall be at least one value set in presence. Besides these two data representing elements, three more childs are defined: <dataRef>, <propRel> and <objRel>. All of the five childs are defined for optional usage. Nevertheless, a XCDL document shall only be valid if at least one of it appears.

<rawVal>

A raw value of a property is the representation of the underlying data according to the representation within the source object. The only modifications this data can be subjected are the required representations as UTF-16 encoded data for character data or hexadecimal binary encoding for binary data.
The description of property values as raw data is an option. It may be useful for various tasks and according to the nature of properties.

```
... This is a \b text\b0 ...   <!-- original data -->

<property id=”5” source=”raw” cat=”descr”>
      <name>boldFace</name>
      <valueSet id=”1”>
            <rawVal>text</val>
      </valueSet>
      ...
</property>
```

Ex. XCDL-4: Usage of element <rawVal>

In example XCDL-4 the value of the property ‘boldFace’ is represented as a raw value. For the relation of the properties value ‘text’ to the specific location within the objects’ data see this section, element <dataRef>.

<labVal>
A labelled value of a property is the representation of the properties data in an interpreted form that goes beyond the simple representation of property values as raw data, which may only be modified by the constraints of the XCDL required encoding representations (UTF-16, hexadecimal binary, see above). The element <val> is created to pick up the distinct labelled value, element <type> indicates the type of labelling. Allowed values for the labelling types are defined in the associated high-level schema ‘XCLDataTypes.xsd’ which is part of the name bundle (see section about name libraries). A distinct labelled value can either be one of the predefined fixed labels for property values (using the simple type ‘fixLabelsType’ defined in ‘XCDLBasicTypes.xsd’ which is a union type of predefined values (‘value labels’) of the name libraries), an integer or decimal number or any string.
If the type of the labelled value is ‘fixLabel’ a XCDL validator shall only validate a XCDL document as correct if one of the predefined values is used in the <val> element.
The distinct labelled value given in <val> may optionally be further specified by two attributes, ‘unit’ and ‘base’, particularly in case of integer or decimal numbers. Both are derived from the XCDL basic type ‘measureType’ that is defined in the associated ‘XCDLBasicTypes.xsd’.

```
<property id="1" source="raw" cat="descr">
      <name>width</name>
      <valueSet id="1">
            <labVal>
                  <val unit="pixel">32</val>
                  <type>int</type>
            </labVal>
            ...
      </valueSet>
</property>
```

Ex. XCDL-5: Property with labelled value

Example XCDL-5 shows the representation of the image property ‘width’ as a labelled value. The original binary data representation is transformed to a labelled value, whereas the type of the labelled value is an integer number, the distinct value is the integer number ‘32’. The attribute ‘unit’ indicates the reference unit of the number.

Some properties distinct values may be expressed as a sequence of values which are repeatable groups. In such a case, the attribute ‘group’ is used, indicating the number of single values in sequence which form a logical unit. Example XCDL-6 is another one taken from the image domain:
A palette with entries for colours mixed in red, green and blue channels may be x times repeatable depending on the bit depth. In this case the 'group' attribute is set on value '3', telling that a meaningful unit consists of succeeding triplets.

```
<property id="10" source="raw" cat="descr">
      <name>rgbPalette</name>
      <valueSet id="1">
            <labVal>
                  <val>0 255 0 255 0 0 255 255 0 0 0 255</val>
                  <type group="3">int</type>
            </labVal>
            ...
      </valueSet>
</property>
```

Ex. XCDL-6: Property with labelled value using the ‘group’ attribute.

<dataRef>

Especially for the purpose of comparing digital objects described as XCDL documents it is necessary to indicate the distinct reference of property values to the location within the object, i.e. to the according byte sequences. This is done by using the <dataRef> element.
The element is optional but shall not appear more than exactly one time within a value set. The precise reference is indicated by the <ref> child element that can be used repeatable. Therefore the attributes ‘id’, ‘start’ and ‘end’ are provided. The first one is necessary for identifying the <data> or <normdata> element, depending on the data reference indicator ‘ind’ (see this section below). The others pick up the concrete location within the referenced byte sequence, starting with ‘start’ and ending indicated by ‘end’. The start and end values are counted in integer numbers, whereas one single character (for UTF-16 encoded character data) is exactly one number and one hexadecimal number (formed by two characters in the hexadecimal representation) as well.

The number of data references depends on the properties and object category. For example, a large text object may have many strings where the characters are formatted in boldface distributed over different locations within the object. In such a case the property ‘boldFace’ has an according number of value sets each with a specification of the exact reference to the location within the digital object, realised through a <dataRef> element (including child <ref>).

The attribute ‘ind’ (=indicator) is required. It indicates the type of data reference. Properties’ values can either be referenced to the overall data (element <data>) or to normalised data (element <normData>). The following values are predefined:

none= no reference to data
sourceAll= reference to total source data (<data>)
sourceSpecific= reference to specific part of source data
normAll= reference to all normalized data (<normData>)
normSpecific= reference to specific normalized data

If the value is one of the specific values, the <ref> element shall be included.

Example XCDL-7 shows a data reference structure within a XCDL document. The original text, formatted in RTF, is intended to be rendered with the words ‘Ashes’ and ‘more’ in boldface. To map this, two <valueSet> elements are created both containing a <dataRef> element with references (<ref>) to the exact location of the properties’ values within the normalised data.

Ashes to **Ashes** once **more.**

```
<data id=”1”>
{\rtf1\ansi\ansicpg1252\deff0\deflang1031{\fonttbl{\f0\fswiss\fcharset0
Arial;}}\viewkind4\uc1\pard\f0\fs20 \b Ashes\b0  to \b  Ashes\b0  once \b
more\b0.\par}
</data>

<normData id=”1” type=”text”>
 Ashes to Ashes once more.
</normData>

<property id=”5” source=”raw” cat=”descr”>
      <name>boldFace</name>
      <valueSet id=”1”>
            <rawVal>Ashes</val>
            <dataRef ind=”normSpecific”>
                  <ref id=”1” start=”0” end=”4”/>
                  <ref id=”1” start=”9” end=”13”/>
            </dataRef>
      </valueSet>
      <valueSet id=”2”>
            <rawVal>more</val>
            <dataRef ind=”normSpecific”>
                  <ref id=”1” start=”20” end=”23”/>
            </dataRef>
      </valueSet>
      ...
</property>
```

Ex. XCDL-7: Usage of <dataRef>

<propRel>

Properties can be related to each other. A property relation indicates that the values of the single properties involved can only be interpreted as reasonable if they are understood as an entire logical unit.

The attribute ‘id’ identifying the relation is required. The property relations are specified by the usage of the <rel> element. The attribute ‘propId’ indicates the identification number of the related property, ‘valueSetId’ the distinct value set within this property to which a relation exists.

Regard example XCDL-8 taken from the image domain. A property called ‘suggestedPaletteName’ with labelled value ‘PALETTENAME’ and data reference to the entire normalised data has two relationships. The first one with four properties identified through their IDs (27 to 30) whereas every single relation within the properties 27 to 30 relates to the properties’ value set #1 of every single property.

The second property relationship has almost the same structure, except in relation with property #28 where the related value set is value set #2.

Actually these relations simply tell us, that there is a suggested palette specified which name is ‘PALETTENAME’. This palette can have various characteristics. One of the variations concern property #28. The palette can have a suggested palette RGB sequence of triplets with value ‘1000’ and another with value ‘2000’.

```
<property id="26" source="raw" cat="descr">
      <name>suggestedPaletteName</name>
      <valueSet id="1">
      <labVal>
            <val>PALETTENAME</val>
            <type>string</type>
      </labVal>
      <dataRef ind="normAll"/>
      <propRel id="1">
            <rel propId="27" valSetId="1"/>
            <rel propId="28" valSetId="1"/>
            <rel propId="29" valSetId="1"/>
            <rel propId="30" valSetId="1"/>
      </propRel>
      <propRel id="2">
            <rel propId="27" valSetId="1"/>
            <rel propId="28" valSetId="2"/>
            <rel propId="29" valSetId="1"/>
            <rel propId="30" valSetId="1"/>
      </propRel>
</valueSet>
</property>
...
<property id="28" source="raw" cat="descr">
      <name>suggestedPaletteRGB</name>
      <valueSet id="1">
            <labVal>
                  <val>1000 1000 1000</val>
                  <type group="3">int</type>
            </labVal>
            <dataRef ind="normAll"/>
            <propRel id="1">
                  <rel propId="26" valSetId="1"/>
                  <rel propId="27" valSetId="1"/>
                  <rel propId="29" valSetId="1"/>
                  <rel propId="30" valSetId="1"/>
            </propRel>
      </valueSet>
      <valueSet id="2">
            <labVal>
```

```
                  <val>2000 2000 2000</val>
                  <type group="3">int</type>
            </labVal>
            <dataRef ind="normAll"/>
            <propRel id="1">
                  <rel propId="26" valSet="1"/>
                  <rel propId="27" valSet="1"/>
                  <rel propId="29" valSet="2"/>
                  <rel propId="30" valSet="2"/>
            </propRel>
      </valueSet>
...
```

Ex. XCDL-8: Usage of <propRel>

<objRel>

The <objRel> element is primarily used for relations to objects which occur in the context of embedded objects.

The attributes ‘xcdlId’ and ‘objId’ from the <rel> child element provide the identification numbers of the related object within a certain XCDL document. Example XCDL-9 shows the structure of a <property> element with an object relation. The digital object wrapped in the <object id=”1” ...> contains an embedded ICC profile. This embedded ICC profile is outlined as a property (name ‘ICCProfile’) of the object. An ICC profile is an object that can be described for itself. This is done by the second object within the XCDL document (object with id=”2”). The <objRel> element within the property structure relates to this embedded object via providing the xcdlId and the objID values. Note that in this example both objects are described within a single XCDL document. Nevertheless it is also possible to “outsource” the embedded object or to relate to an existing identical object.

```
<xcdl ... id=”A123456”>
      <object id=”1” ...>
      ...
            <property id="19" source="raw" cat="descr">
                  <name>ICCProfile</name>
                  <valueSet id="1">
                        <objRel xcdlId=”A123456” objId=”2”/>
                  </valueSet>
            </property>
      ...
      </object>
      <object id=”2” ...>
      ...
      </object>
</xcdl>
```

Ex. XCDL-9: Indication of an object relation within <property> element

3.1.3. XCDL Basic Types

The XCDL core uses a couple of schema types (simple types) which are fundamental to the language. The actual version of the XCDL core includes these types within a separate XML schema ‘XCDLBasicTypes.xsd’. It is likely that these basic types will be extended in future versions of the XCDL along with the increasing XCEL specifications available.

3.1.3.1 Union Types

Union types specify the allowed data type(s)) for the content of elements. Currently there are three union types predefined. For details see the schema.

3.1.3.2 Identification Number Types

These types are predefined to determine a vocabulary of allowed ID types. Currently there are two types predefined.

3.1.3.3 Information Types

Information types relate to the XCDL core element <normData>. Compare the accordant section for details.

3.1.3.4 Format Identifier Types

Format identifier types specify the vocabulary for unique format identifiers. Currently the use of the PRONOM format identifiers are recommended.

3.1.3.5 Measure Types

Measure types currently relates to the child element <val> of <labVal>. For details compare the corresponding section.

3.1.3.6 Name Type, Labelled Values Type and Fixed Labels Type

These three types are created to establish an allowed vocabulary for the element contents of <name> and the childs of <labVal> which are <val> and <type>.
They all are derived from the vocabulary defined within the name libraries. For more details consult the corresponding sections of the XCDL core specification and the name libraries section.

3.2. XCDLCore.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited by Volker Heydegger (PLANETS, PC2, University of Cologne, HKI)
-->
<!-- version 1.0, October 31, 2006 -->
<!-- change history: no changes -->
<xs:schema xmlns="http://www.planets-project.eu/xcl/schemata"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xcdl="http://www.planets-
project.eu/xcl/schemata/xcdl" targetNamespace="http://www.planets-
project.eu/xcl/schemata" elementFormDefault="qualified" version="1.0"
xml:lang="en">
      <xs:import namespace="http://www.planets-
project.eu/xcl/schemata/xcdl" schemaLocation="XCDLBasicTypes.xsd"/>
      <!-- ********************************** xcdl section  (root
element)********************************************* -->
      <xs:element name="xcdl">
            <xs:annotation>
                  <xs:documentation>eXtensible Characterisation Description
Language (XCDL)</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:complexContent>
                        <xs:extension base="xcdlType"/>
                  </xs:complexContent>
            </xs:complexType>
      </xs:element>
      <!-- ............complex type:  xcdlType ................. -->
      <xs:complexType name="xcdlType">
            <xs:annotation>
                  <xs:documentation>A XCDL document describes digital
objects. Every xcdl description shall have an identification number.
                </xs:documentation>
            </xs:annotation>
            <xs:sequence>
                  <xs:element ref="composition" minOccurs="0"
maxOccurs="unbounded"/>
                  <xs:element ref="object" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="id" type="xcdl:idType02Type"
use="required"/>
      </xs:complexType>
      <!-- *****************************   composition section (child of:
'xcdl')  ******************************* -->
      <xs:element name="composition">
            <xs:annotation>
                  <xs:documentation>Function: Digital objects may contain
digital objects or may be related among themselves. Such digital entities
can be wrapped by the 'composition' element. For indication of logical
relations, attribute 'use' can be set on value 'logical'
                  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="xcdlRef" maxOccurs="unbounded"/>
                  </xs:sequence>
                  <xs:attribute name="use" type="compositionType"
use="optional"/>
            </xs:complexType>
      </xs:element>
      <!-- ................composition type ......................... -->
      <xs:simpleType name="compositionType">
```

```
            <xs:annotation>
                  <xs:documentation>indicates the type of composition; set
value on 'logical' for logical units</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:token">
                  <xs:enumeration value="logical"/>
            </xs:restriction>
      </xs:simpleType>
      <!-- ....................  element: 'xcdlRef' .....................
-->
      <xs:element name="xcdlRef">
            <xs:annotation>
                  <xs:documentation>Function: A composition of digital
objects is always manifest in at least one xcdl description. The reference
ID to it (or them) shall be inserted here.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:sequence>
                        <xs:element name="obj" maxOccurs="unbounded">
                              <xs:annotation>
                                    <xs:documentation>Every xcdl description
is composed of x objects. All objects which refer to the same digital
entity shall be included by adding the ID of the according
object.</xs:documentation>
                              </xs:annotation>
                              <xs:complexType>
                                    <xs:attribute name="id"
type="xcdl:idType01Type" use="required"/>
                              </xs:complexType>
                        </xs:element>
                  </xs:sequence>
                  <xs:attribute name="id" type="xcdl:unionType02Type"
use="required"/>
            </xs:complexType>
      </xs:element>
      <!-- *****************************  object section (child of: 'xcdl')
*********************************************** -->
      <xs:element name="object">
            <xs:annotation>
                  <xs:documentation>Function: Wrapper element for objects
to be described through a xcdl. Every object shall have an identification
number. The native format of the object may be added.
                  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="data" minOccurs="0"/>
                        <xs:element ref="normData" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:element ref="property" maxOccurs="unbounded"/>
                  </xs:sequence>
                  <xs:attribute name="id" type="xcdl:idType01Type"
use="required"/>
                  <xs:attribute name="natFormat"
type="xcdl:formatIdentifierType" use="optional"/>
            </xs:complexType>
      </xs:element>
      <!-- ***************************  data section (child of: 'object')
**************************************** -->
      <xs:element name="data">
            <xs:annotation>
```

```
                  <xs:documentation> Function: Wraps the full source 
objects data. For relation and reference purposes, an identification number 
is required.
                  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:simpleContent>
                        <xs:extension base="xcdl:unionType01Type">
                              <xs:attribute name="id" 
type="xcdl:idType01Type" use="required"/>
                        </xs:extension>
                  </xs:simpleContent>
            </xs:complexType>
      </xs:element>
      <!-- *************************** normalized data section (child of: 
'object')  ***************************** -->
      <xs:element name="normData">
            <xs:annotation>
                  <xs:documentation>Function: Wraps normalized data.

                  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:simpleContent>
                        <xs:extension base="xcdl:unionType01Type">
                              <xs:attribute name="id" 
type="xcdl:idType01Type" use="required"/>
                              <xs:attribute name="type" 
type="xcdl:informType" use="optional"/>
                        </xs:extension>
                  </xs:simpleContent>
            </xs:complexType>
      </xs:element>
      <!-- ************************  Property section (child of: 'object') 
******************************* -->
      <xs:element name="property">
            <xs:annotation>
                  <xs:documentation>Function: Wraps the objects properties. 
A property shall have an identification number within the xcdl description.
                  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="name"/>
                        <xs:element ref="valueSet" maxOccurs="unbounded"/>
                  </xs:sequence>
                  <xs:attribute name="id" type="xcdl:idType01Type" 
use="required"/>
                  <xs:attribute name="source" type="sourceType" 
use="optional"/>
                  <xs:attribute name="cat" type="catType" use="optional"/>
            </xs:complexType>
      </xs:element>
      <!--  ...........simple type: sourceType............:  -->
      <xs:simpleType name="sourceType">
            <xs:annotation>
                  <xs:documentation> the source the property is derived 
from. 'raw' =derived from the source object; 'implicit'=property is not 
fixed to the source object but derived from the source objects format 
specification  extended= no derivation from the source object or the source 
objects specification; 'added'= at a later date inserted
                  </xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:token">
                  <xs:enumeration value="raw"/>
                  <xs:enumeration value="implicit"/>
                  <xs:enumeration value="extended"/>
                  <xs:enumeration value="added"/>
            </xs:restriction>
      </xs:simpleType>
      <!--  ............simple type: catType.............:  -->
      <xs:simpleType name="catType">
            <xs:annotation>
                  <xs:documentation> the properties category:
'descr'=descriptive property, i.e. occurence of object describing property;
'hist'= history property, i.e. property that may appear in a different
shape in the source object which may be resolved in the xcdl description
(e.g., compressed data); 'cont'= content property, i.e. relating directly
to a byte sequence; 'extern'= property that refers to external item, i.e.
not related to objects data, e.g.,  software and hardware used to create
the object.
                  </xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:token">
                  <xs:enumeration value="descr"/>
                  <xs:enumeration value="hist"/>
                  <xs:enumeration value="cont"/>
                  <xs:enumeration value="extern"/>
            </xs:restriction>
      </xs:simpleType>
      <!-- ************************  property name section (child of:
'property')  *************************************  -->
      <xs:element name="name">
            <xs:annotation>
                  <xs:documentation> Function: Wraps a unique property
name, defined by a xcdl names library. Namings for properties which refer
to identical issues may differ depending on the format. The different term
may be added using the 'alias' attribute.
                  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:simpleContent>
                        <xs:extension base="xcdl:nameType">
                              <xs:attribute name="alias" type="xs:string"
use="optional"/>
                        </xs:extension>
                  </xs:simpleContent>
            </xs:complexType>
      </xs:element>
      <!-- ***********************  property value set section  (child of:
'property') ******************************** -->
      <xs:element name="valueSet">
            <xs:annotation>
                  <xs:documentation> Function: Wrapper element for the
properties raw and labelled values. Every value set shall have an
identification number.
                  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="rawVal" minOccurs="0"/>
                        <xs:element ref="labVal" minOccurs="0"/>
                        <xs:element ref="dataRef" minOccurs="0"/>
                        <xs:element ref="propRel" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:element ref="objRel" minOccurs="0"
maxOccurs="unbounded"/>
                  </xs:sequence>
                  <xs:attribute name="id" type="xcdl:idType01Type"
use="required"/>
            </xs:complexType>
      </xs:element>
      <!-- ************************* raw value section (child of:
'valueSet')  ************************************** -->
      <xs:element name="rawVal">
            <xs:annotation>
                  <xs:documentation>Function: Wraps the distinct raw value,
as extracted from the source object ;by default bytes shall be encoded in
UTF-16 for non-binary data, in hex numbers for binary data.
                  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:simpleContent>
                        <xs:extension base="xcdl:unionType01Type"/>
                  </xs:simpleContent>
            </xs:complexType>
      </xs:element>
      <!--  ************************  labelled value section (child of:
'valueSet') ******************************** -->
      <xs:element name="labVal">
            <xs:annotation>
                  <xs:documentation> Function: Wrapping element for
labelled value. A labelled value shall be expressed by its distinct value
and its type.
                  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:sequence>
                        <xs:element name="val" maxOccurs="unbounded">
                              <xs:annotation>
                                    <xs:documentation> The distinct labelled
value. This can either be a UTF-16 encoded string, an integer or decimal
number or a fixed label defined in simple type 'xcdlFixedLabls'. For an
accurate representation some properties values, especially those expressed
in integers, may require additional measurement information.
                                    </xs:documentation>
                              </xs:annotation>
                              <xs:complexType>
                                    <xs:simpleContent>
                                          <xs:extension
base="xcdl:unionType10Type">
                                                <xs:attribute name="unit"
type="xcdl:measureType" use="optional"/>
                                                <xs:attribute name="base"
type="xcdl:measureType" use="optional"/>
                                          </xs:extension>
                                    </xs:simpleContent>
                              </xs:complexType>
                        </xs:element>
                        <xs:element name="type" maxOccurs="unbounded">
                              <xs:annotation>
                                    <xs:documentation> Function: Wraps type
of the labelled value. Some properties distinct values may be expressed as
a sequence of values which are repeatable groups.E.g.: a palette with
entries for colours mixed from red, green and blue channels may be x times
```

```
repeatable depending on the bit depth. In this case the 'group' attribute
is set on value '3', telling the reading tool that a meaningful unit
consists of triplets.
                                    </xs:documentation>
                              </xs:annotation>
                              <xs:complexType>
                                    <xs:simpleContent>
                                          <xs:extension
base="xcdl:labValType">
                                                <xs:attribute name="group"
type="xs:unsignedInt" use="optional"/>
                                          </xs:extension>
                                    </xs:simpleContent>
                              </xs:complexType>
                        </xs:element>
                  </xs:sequence>
            </xs:complexType>
      </xs:element>
      <!-- ***************** data reference section (child of: 'valueSet')
*************************  -->
      <xs:element name="dataRef">
            <xs:annotation>
                  <xs:documentation>Function: Reference to data. This can
either be the source data (element 'data') or normalized data (element
'normData'). Attribute 'ind': Indicator for the type of reference.
                  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="ref" minOccurs="0"
maxOccurs="unbounded"/>
                  </xs:sequence>
                  <xs:attribute name="ind" type="dataRefType"
use="required"/>
            </xs:complexType>
      </xs:element>
      <!-- ........simple type: 'dataRefType': ....... -->
      <xs:simpleType name="dataRefType">
            <xs:annotation>
                  <xs:documentation>Type of data reference: none= no
reference to data; sourceAll= reference to total source data;
sourceSpecific= reference to specific part of source data; normAll=
reference to all normalized data; normSpecific= reference to specific
normalized data. If value is 'specific', 'ref' element shall be included.
                  </xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:token">
                  <xs:enumeration value="none"/>
                  <xs:enumeration value="sourceAll"/>
                  <xs:enumeration value="sourceSpecific"/>
                  <xs:enumeration value="normAll"/>
                  <xs:enumeration value="normSpecific"/>
            </xs:restriction>
      </xs:simpleType>
      <!-- ......... element 'ref':  ............. -->
      <xs:element name="ref">
            <xs:annotation>
                  <xs:documentation> Function: distinct data coordinates
for references. Startposition and Endposition of the bytes within the
referenced data shall be declared using the attributes. Also required is an
identification number for each 'ref' element.
                  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:attribute name="id" type="xcdl:idType01Type"
use="required"/>
                  <xs:attribute name="start" type="xs:unsignedLong"
use="required"/>
                  <xs:attribute name="end" type="xs:unsignedLong"
use="required"/>
            </xs:complexType>
      </xs:element>
      <!-- *************************  property relations section (child of:
'valueSet') ************************* -->
      <xs:element name="propRel">
            <xs:annotation>
                  <xs:documentation>Function: fixing relations among
properties.
                  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:sequence>
                        <xs:element name="rel" maxOccurs="unbounded">
                              <xs:annotation>
                                    <xs:documentation> Function: Relation
parameters: ID of the related property and ID of the related value set
within it. An ID can optionally assigned to a relation.</xs:documentation>
                              </xs:annotation>
                              <xs:complexType>
                                    <xs:attribute name="propId"
type="xcdl:idType01Type" use="required"/>
                                    <xs:attribute name="valSetId"
type="xcdl:idType01Type" use="required"/>
                              </xs:complexType>
                        </xs:element>
                  </xs:sequence>
                  <xs:attribute name="id" type="xcdl:idType01Type"
use="optional"/>
            </xs:complexType>
      </xs:element>
      <!-- **********************  object relations section (child of:
'valueSet')  *************************  -->
      <xs:element name="objRel">
            <xs:annotation>
                  <xs:documentation> Function: fixing relations among
properties and objects, normally in case of embedded objects.
                  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
                  <xs:sequence>
                        <xs:element name="rel" maxOccurs="unbounded">
                              <xs:annotation>
                                    <xs:documentation> Function: Relation
parameters: ID of the related xcdl description and ID of the related object
within it. An ID can optionally assigned to a relation.</xs:documentation>
                              </xs:annotation>
                              <xs:complexType>
                                    <xs:attribute name="xcdlId"
type="xcdl:idType02Type" use="required"/>
                                    <xs:attribute name="objId"
type="xcdl:idType01Type" use="required"/>
                              </xs:complexType>
                        </xs:element>
                  </xs:sequence>
                  <xs:attribute name="id" type="xcdl:idType01Type"
use="optional"/>
            </xs:complexType>
      </xs:element>
</xs:schema>
```

3.3. XCDLBasicTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- extensible characterisation description language: basic types -->
<!-- edited by Volker Heydegger (PLANETS, PC2, University of Cologne, HKI)
-->
<!-- version 1.0, October 31, 2006 -->
<!-- change history: no changes -->
      <xs:schema
      xmlns="http://www.planets-project.eu/xcl/schemata/xcdl"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:nm="http://www.planets-project.eu/xcl/schemata/xclnames"
      xmlns:xcdl="http://www.planets-project.eu/xcl/schemata/xcdl"
      targetNamespace="http://www.planets-project.eu/xcl/schemata/xcdl"
      elementFormDefault="qualified"
      version="1.0"
      xml:lang="en">

      <xs:import schemaLocation="nameLibs/XCLNamesLib.xsd"
namespace="http://www.planets-project.eu/xcl/schemata/xclnames"/>


      <!-- ****************************  union types
******************************************-->
      <xs:simpleType name="unionType01Type">
            <xs:union memberTypes="xs:string xs:hexBinary"/>
      </xs:simpleType>
      <xs:simpleType name="unionType02Type">
            <xs:union memberTypes="xs:string xs:integer"/>
      </xs:simpleType>
      <xs:simpleType name="unionType10Type">
            <xs:union memberTypes="xs:string xs:decimal
xcdl:fixLabelsType"/>
      </xs:simpleType>
<!-- ***********************  identification number types
********************************* -->
      <xs:simpleType name="idType01Type">
            <xs:restriction base="xs:unsignedInt"/>
      </xs:simpleType>
      <xs:simpleType name="idType02Type">
            <xs:restriction base="xcdl:unionType02Type"/>
      </xs:simpleType>
      <!-- *************************  information types
*************************************** -->
      <xs:simpleType name="informType">
            <xs:annotation>
                  <xs:documentation> type of information represented by
data.
                  </xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:string">
                  <xs:enumeration value="text"/>
                  <xs:enumeration value="image"/>
                  <xs:enumeration value="audio"/>
                  <xs:enumeration value="object"/>
                  <xs:enumeration value="other"/>
            </xs:restriction>
      </xs:simpleType>
      <!-- ********************** format identifier type
***************************************  -->
      <xs:simpleType name="formatIdentifierType">
            <xs:restriction base="xs:string"/>
      </xs:simpleType>
      <!-- ***********************  measure types
********************************************-->
      <xs:simpleType name="measureType">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="bit"/>
                  <xs:enumeration value="twip"/>
                  <xs:enumeration value="pixel"/>
                  <xs:enumeration value="inch"/>
                  <xs:enumeration value="meter"/>
                  <xs:enumeration value="palette"/>
            </xs:restriction>
      </xs:simpleType>
      <!-- *********************** simple type for xcl defined namings for
properties: *********************************************** -->
      <xs:simpleType name="nameType">
      <xs:annotation>
                  <xs:documentation> union of xcl defined namings for xcdl
properties</xs:documentation>
            </xs:annotation>
            <xs:union memberTypes="nm:xclImageBasicNames
nm:xclImageExtendedNames nm:xclAudioBasicNames nm:xclAudioExtendedNames
            nm:xclMultimediaBasicNames nm:xclMultimediaExtendedNames
nm:xclOtherBasicNames nm:xclOtherExtendedNames"/>
      </xs:simpleType>
      <!-- ***********************simple type for xcl defined data types
for xcdl labelled value types:
*********************************************** -->
      <xs:simpleType name="labValType">
      <xs:annotation>
                  <xs:documentation>derived from xcl defined data
types</xs:documentation>
            </xs:annotation>
            <xs:union memberTypes="nm:xclDataTypes"/>
      </xs:simpleType>
      <!-- ***********************simple type for xcl defined namings for
xcdl properties' labelled values:
*********************************************** -->
      <xs:simpleType name="fixLabelsType">
      <xs:annotation>
                  <xs:documentation> union of xcl defined namings for xcdl
fixed labellings</xs:documentation>
            </xs:annotation>
            <xs:union memberTypes="nm:xclImageValueLabels
nm:xclAudioValueLabels nm:xclMultimediaValueLabels
nm:xclOtherValueLabels"/>
      </xs:simpleType>

</xs:schema>
```

4. Shared Schemata: Name Libraries

4.1 Initial definitions

Within the entire architecture of the extensible characterisation language, i.e. for both languages XCDL and XCEL, there are some common namings and usages for properties, symbols and property values' data types. This shared vocabulary is spread over a couple of separate XML schemata, which are used in common.
At the moment, there are four shared schemata outlined in table NL-1.

File	Includes	Namespace
XCLNamesLib	XCLBasicNamesLib.xsd XCLImageNamesLib.xsd	xmlns:nm= “http://www.planets-project.eu/xcl/schemata/xclnames“
XCLDataTypesLib.xsd		xmlns:nm= “http://www.planets-project.eu/xcl/schemata/xclnames“
XCLBasicNamesLib.xsd	XCLDataTypesLib.xsd	xmlns:nm= “http://www.planets-project.eu/xcl/schemata/xclnames“
XCLImageNamesLib.xsd	XCLBasicNamesLib.xsd	xmlns:nm= “http://www.planets-project.eu/xcl/schemata/xclnames“

Table NL-1: Name libraries currently integrated into the entire architecture.

All libraries are in the same namespace. Schemata which include or import the XCLBasicNamesLib.xsd' have access to elements defined in the 'XCLDataTypesLib.xsd' as well. Schemata which include or import the 'XCLImageNamesLib.xsd' have access to elements on the 'XCLDataTypesLib.xsd' as well as to the 'XCLBasicNamesLib.xsd'.
The 'XCLNamesLib.xsd' includes all specific name libraries. The collection will be successively completed in later versions of the XCL. Currently 'XCLImageNamesLib.xsd' is included besides the basic names library.
For an overview see figure NL-2.

A short description of the single libraries' features is listed here:

4.1.1 XCLDataTypesLib.xsd

Contains data type definitions as for example:

1. int

2. int8
3. int16
4. ...
5. uint8
6. uint16
7. ...
8. string
9. fixLabel

...

The list of data types can be extended, if other data types are required. The data types are defined as enumerations and can be referred to as nm:dataTypeDefinition, where „nm“ indicates the namespace „xclnames“. This ensures consistent name assignment in XCEL and XCDL.

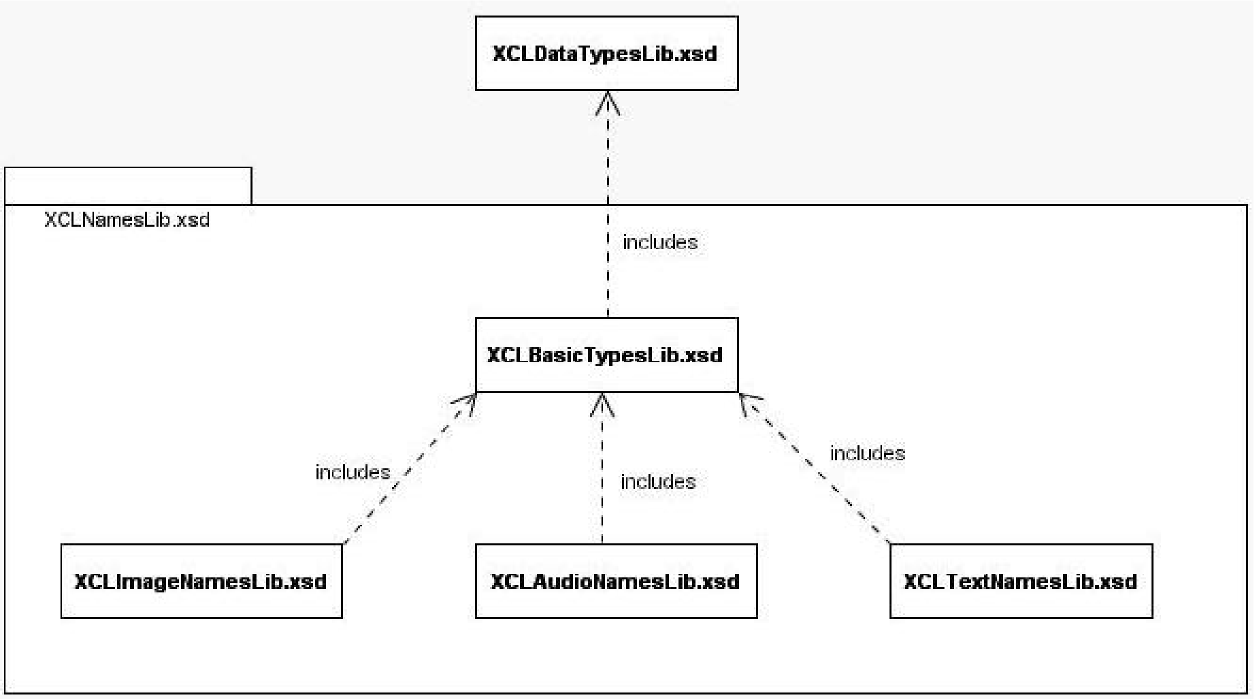


Fig. NL-2: Structure and dependencies of the Name Libraries bundle

XCLBasicNamesLib.xsd

- Includes ‘XCLDataTypesLib.xsd’

- Contains basic name definitions, i.e. names that might occur in every file regardless of the specific file format

- Enumerations can be extended easily.

- For shared use of XCEL and XCDL

- Structured into:
    - Global Element Names: names that occur once per file (i.e. „signature“, „file length“...)
    - Basic Element Names: „byteOrder“, „checksum“....

(included as ‘nm:xclBasicNamesDefinitions’)

XCLImageNamesLib.xsd

- Includes ‘XCLBasicNamesLib.xsd’
- Contains name definitions that are specific to image files
- Enumerations can be extended easily
- For shared use of XCEL and XCDL
- Structured into:

XCLImageBasicNames:

- width
- height
- colorDepth
- compression
- ...

XCLImageExtendedNames:

artist

copyright

cellHeight

cellWidth

...

XCLImageValueLabels:

compression methods: zlibDeflateInflate, CCITT_group3,...

image (colour) types: greyscale, truecolour, palette, ...

(included as nm:xclImageNameDefinitions)

4.2 XCLNamesLib.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- extensible characterisation language: basic names library for XCDL and
XCEL documents -->
<!-- created by PLANETS, PC2, University of Cologne, HKI -->
<!-- version 1.0, October 31, 2006 -->
<!-- change history: no changes -->
      <xs:schema
            xmlns="http://www.planets-project.eu/xcl/schemas/xclnames"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:nm="http://www.planets-project.eu/xcl/schemas/xclnames"
            targetNamespace="http://www.planets-
project.eu/xcl/schemas/xclnames"
            elementFormDefault="qualified"
            version="1.0"
            xml:lang="en">

            <xs:include schemaLocation="XCLBasicNamesLib.xsd"/>
            <xs:include schemaLocation="XCLImageNamesLib.xsd"/>
            <!--
*******************************************************************
xclNamesDefinitions type
********************************************************************* -->
            <xs:simpleType name="xclNamesDefinitions">
                  <xs:annotation>
                        <xs:documentation>high-level simple type for xcdl
and xcel defined namings of symbols, properties and properties' labelled
values</xs:documentation>
                  </xs:annotation>
                  <xs:union memberTypes="nm:xclImageNamesDefinitions
nm:xclAudioNamesDefinitions nm:xclMultimediaNamesDefinitions
nm:xclTextNamesDefinitions nm:xclOtherNamesDefinitions"></xs:union>
            </xs:simpleType>
      </xs:schema>
```

4.3 XCLDataTypesLib.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- extensible characterisation language: data types library for XCDL and
XCEL documents -->
<!-- created by PLANETS, PC2, University of Cologne, HKI -->
<!-- version 1.0, October 31, 2006 -->
<!-- change history: no changes -->
<xs:schema xmlns="http://www.planets-project.eu/xcl/schemas/xclnames"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:nm="http://www.planets-
project.eu/xcl/schemas/xclnames" targetNamespace="http://www.planets-
project.eu/xcl/schemas/xclnames" elementFormDefault="qualified"
version="1.0" xml:lang="en">
      <!-- ****************** data type definitions
************************ -->
      <xs:simpleType name="xclDataTypeDefinition">
            <xs:union memberTypes="nm:xclDataTypes"/>
      </xs:simpleType>
      <!-- *******************  data types
******************************** -->
      <xs:simpleType name="xclDataTypes">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="decimal">
                        <xs:annotation>
                              <xs:documentation>standard mathematical
concept of decimal numbers, negativ or positive, no limit on places and
decimal places</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="int">
                        <xs:annotation>
                              <xs:documentation>standard mathematical
concept of numbers, negative or positive, no limit on
places</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="int8">
                        <xs:annotation>
                              <xs:documentation>signed 8 bits
integer</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="int16">
                        <xs:annotation>
                              <xs:documentation>signed 16 bits
integer</xs:documentation>
```

```
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="int32">
                        <xs:annotation>
                              <xs:documentation>signed 32 bits
integer</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="int64">
                        <xs:annotation>
                              <xs:documentation>signed 64 bits
integer</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="uint8">
                        <xs:annotation>
                              <xs:documentation>unsigned 8 bits
integer</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="uint16">
                        <xs:annotation>
                              <xs:documentation>unsigned 16 bits
integer</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="uint32">
                        <xs:annotation>
                              <xs:documentation>unsigned 32 bits
integer</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="uint64">
                        <xs:annotation>
                              <xs:documentation>unsigned 64 bits
integer</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="string">
                        <xs:annotation>
                              <xs:documentation>Any encoded character
string.</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="fixLabel">
                        <xs:annotation>
                              <xs:documentation>A unique label that conforms
to xcdl predefined fixed labellings</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="freeLabel">
                        <xs:annotation>
                              <xs:documentation>A non xcdl predefined
labelling (should only be used if a fixed label is not
available)</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="dateTime">
                        <xs:annotation>
                              <xs:documentation>type for full date and time
according to ISO 8601</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
```

```
            </xs:restriction>
      </xs:simpleType>
</xs:schema>
```

4.4 XCLBasicNamesLib.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- extensible characterisation language: basic names library for XCDL and
XCEL documents -->
<!-- created by PLANETS, PC2, University of Cologne, HKI -->
<!-- version 1.0, October 31, 2006 -->
<!-- change history: no changes -->
      <xs:schema
      xmlns="http://www.planets-project.eu/xcl/schemas/xclnames"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:nm="http://www.planets-project.eu/xcl/schemas/xclnames"
      targetNamespace="http://www.planets-project.eu/xcl/schemas/xclnames"
      elementFormDefault="qualified"
      version="1.0"
      xml:lang="en">

      <xs:include schemaLocation="XCLDataTypesLib.xsd"/>
      <xs:include schemaLocation="XCLImageNamesLib.xsd"/>

      <!--
******************************************************************
xclNamesDefinitions type
******************************************************************** -->
      <xs:simpleType name="xclNamesDefinitions">
      <xs:annotation>
                  <xs:documentation>high-level simple type for xcdl and
xcel defined namings of symbols, properties and properties' labelled
values</xs:documentation>
            </xs:annotation>
            <xs:union memberTypes="nm:xclImageNamesDefinitions
nm:xclAudioNamesDefinitions nm:xclMultimediaNamesDefinitions
nm:xclTextNamesDefinitions nm:xclOtherNamesDefinitions"></xs:union>
      </xs:simpleType>

      <!--   will be sifted out to files "AudioNamesLib.xsd",
"MultimediaNamesLib.xsd", ... (according to XCLImageNamesLib.xsd) :  -->
      <!-- ************* audio types section  ******************** -->
      <xs:simpleType name="xclAudioNamesDefinitions">
            <xs:union memberTypes="nm:xclAudioBasicNames
nm:xclAudioExtendedNames nm:xclAudioValueLabels"></xs:union>
      </xs:simpleType>
      <xs:simpleType name="xclAudioBasicNames">
            <xs:annotation>
                  <xs:documentation>simple type for xcdl and xcel namings
for audio based objects, including the names for the most common
properties/ symbols</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:string"/>
      </xs:simpleType>
      <xs:simpleType name="xclAudioExtendedNames">
            <xs:annotation>
                  <xs:documentation>simple type for xcdl and xcel namings
for audio based objects, including the names for specific properties/
symbols</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:string"/>
      </xs:simpleType>
      <xs:simpleType name="xclAudioValueLabels">
            <xs:annotation>
                  <xs:documentation>simple type for xcdl and xcel
labellings for property/ symbol values of audio based
objects</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:string"/>
      </xs:simpleType>
      <!-- ***********  multimedia types section ***************  -->
      <xs:simpleType name="xclMultimediaNamesDefinitions">
            <xs:union memberTypes="nm:xclMultimediaBasicNames
nm:xclMultimediaExtendedNames nm:xclMultimediaValueLabels"/>
      </xs:simpleType>
      <xs:simpleType name="xclMultimediaBasicNames">
            <xs:annotation>
                  <xs:documentation>simple type for xcdl and xcel namings
for multimedia based objects, including the names for the most common
properties/ symbols</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:string"/>
      </xs:simpleType>
      <xs:simpleType name="xclMultimediaExtendedNames">
            <xs:annotation>
                  <xs:documentation>simple type for xcdl and xcel namings
for multimedia based objects, including the names for specific properties/
symbols</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:string"/>
      </xs:simpleType>
      <xs:simpleType name="xclMultimediaValueLabels">
            <xs:annotation>
                  <xs:documentation>simple type for xcdl and xcel
labellings for property/ symbol values of multimedia based
objects</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:string"/>
      </xs:simpleType>
      <!-- ***********  text types section ***************  -->
      <xs:simpleType name="xclTextNamesDefinitions">
            <xs:union memberTypes="nm:xclTextBasicNames
nm:xclTextExtendedNames nm:xclTextValueLabels"/>
      </xs:simpleType>
      <xs:simpleType name="xclTextBasicNames">
            <xs:annotation>
                  <xs:documentation>simple type for xcdl and xcel namings
for text based objects, including the names for the most common properties/
symbols</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:string"/>
      </xs:simpleType>
      <xs:simpleType name="xclTextExtendedNames">
            <xs:annotation>
                  <xs:documentation>simple type for xcdl and xcel namings
for text based objects, including the names for specific properties/
symbols</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:string"/>
      </xs:simpleType>
      <xs:simpleType name="xclTextValueLabels">
            <xs:annotation>
                  <xs:documentation>simple type for xcdl and xcel
labellings for property/ symbol values of text based
objects</xs:documentation>
```

```
            </xs:annotation>
            <xs:restriction base="xs:string"/>
      </xs:simpleType>
<!-- ***********  other types section ***************  -->
      <xs:simpleType name="xclOtherNamesDefinitions">
            <xs:union memberTypes="nm:xclOtherBasicNames
nm:xclOtherExtendedNames nm:xclOtherValueLabels"/>
      </xs:simpleType>
      <xs:simpleType name="xclOtherBasicNames">
            <xs:annotation>
                  <xs:documentation>simple type for xcdl and xcel namings
for non-categorized objects, including the names for the most common
properties/ symbols</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:string"/>
      </xs:simpleType>
      <xs:simpleType name="xclOtherExtendedNames">
            <xs:annotation>
                  <xs:documentation>simple type for xcdl and xcel namings
for non-categorized objects, including the names for specific properties/
symbols</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:string"/>
      </xs:simpleType>
      <xs:simpleType name="xclOtherValueLabels">
            <xs:annotation>
                  <xs:documentation>simple type for xcdl and xcel
labellings for property/ symbol values of non-categorized
objects</xs:documentation>
            </xs:annotation>
            <xs:restriction base="xs:string"/>
      </xs:simpleType>

      <!-- ****************************** basic name definitions - names of
properties/ symbols common to all types of objects
**************************************** -->
      <xs:simpleType name="xclBasicNameDefinitions">
            <xs:union memberTypes=" nm:xclGlobalElementNames
nm:xclBasicElementNames nm:xclTimeTypes nm:byteOrder"/>
      </xs:simpleType>
      <xs:simpleType name="xclGlobalElementNames">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="signature">
                        <xs:annotation>
                              <xs:documentation>A sequence of bytes that
uniquely identifies a certain fileformat</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="filelength">
                        <xs:annotation>
                              <xs:documentation>The length in bytes of the
complete file</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
            </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="xclBasicElementNames">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="byteOrder">
                        <xs:annotation>
                              <xs:documentation>Ordering of bytes for multi-
byte data values within a file</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="checksum">
                        <xs:annotation>
                              <xs:documentation>A form of redundancy check,
as a measure for protecting the integrity of data</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="delimiter">
                        <xs:annotation>
                              <xs:documentation>Character or sequence of
characters, without any other semantic meaning, but to separate two
meaningful units</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
            </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="byteOrder">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="bigEndian">
                        <xs:annotation>
                              <xs:documentation>the high-order byte of the
number is stored in memory at the lowest address, and the low-order byte at
the highest address
                              </xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="littleEndian">
                        <xs:annotation>
                              <xs:documentation>the low-order byte of the
number is stored in memory at the lowest address, and the high-order byte
at the highest address
                              </xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
            </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="xclTimeTypes">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="yyear">
                        <xs:annotation>
                              <xs:documentation>a 2 byte date-time
specification indicating a year, as in 2006</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="year">
                        <xs:annotation>
                              <xs:documentation>a 1 byte date-time
specification indicating a year, as in 06</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="month">
                        <xs:annotation>
                              <xs:documentation>a 1 byte date-time
specification indicating a month</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="day">
                        <xs:annotation>
                              <xs:documentation>a 1 byte date-time
specification indicating a day</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="hour">
```

```
                        <xs:annotation>
                              <xs:documentation>a 1 byte date-time
specification indicating an hour</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="minute">
                        <xs:annotation>
                              <xs:documentation>a 1 byte date-time
specification indicating a minute</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="second">
                        <xs:annotation>
                              <xs:documentation>a 1 byte date-time
specification indicating a second</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
            </xs:restriction>
      </xs:simpleType>
</xs:schema>
```

4.5 XCLImageNamesLib.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- extensible characterisation language: image properties' name
definitions library for XCDL and XCEL documents -->
<!-- created by PLANETS, PC2, University of Cologne, HKI -->
<!-- version 1.0, October 31, 2006 -->
<!-- change history: no changes -->
<xs:schema xmlns="http://www.planets-project.eu/xcl/schemas/xclnames"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:nm="http://www.planets-
project.eu/xcl/schemas/xclnames" targetNamespace="http://www.planets-
project.eu/xcl/schemas/xclnames" elementFormDefault="qualified"
version="1.0" xml:lang="en">
      <!-- **************************  union type for all image specific
xcl namings *********************** -->
      <xs:simpleType name="xclImageNamesDefinitions">
            <xs:union memberTypes="nm:xclImageBasicNames
nm:xclImageExtendedNames nm:xclImageValueLabels"/>
      </xs:simpleType>
      <!-- ************************* xcl basic (common) names for images
************************ -->
      <xs:simpleType name="xclImageBasicNames">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="width">
                        <xs:annotation>
                              <xs:documentation>Width of an image.
Corresponds to the horizontal dimension (x-axis)</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="height">
                        <xs:annotation>
                              <xs:documentation>Height of an image.
Corresponds to the vertical dimension (y-axis)</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="colourDepth">
                        <xs:annotation>
                              <xs:documentation>Number of information units
per measure unit to express a certain colour value; usually bits per sample
or per palette index.
```

```
                              </xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="samplesPerPixel">
                        <xs:annotation>
                              <xs:documentation>number of intersections
(components) of a pixel in an image</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="imageType">
                        <xs:annotation>
                              <xs:documentation>Photometric interpretation
of image pixels. E.g., in type 'greyscale' each pixel conforms to one
sample.</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="fillOrder">
                        <xs:annotation>
                              <xs:documentation>The logical order of bits
within a byte.
                              </xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="compression">
                        <xs:annotation>
                              <xs:documentation>algorithm applied to data
for the purpose of minimizing storage size</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="gamma">
                        <xs:annotation>
                              <xs:documentation>numerical parameter used to
describe approximations to certain non-linear transfer functions
encountered in image capture and reproduction  </xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="rgbPalette">
                        <xs:annotation>
                              <xs:documentation>palette for index-coloured
image type. Each pixel is a composition of red, green and blue colour
values which shall appear in exactly this order</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="transparency">
                        <xs:annotation>
                              <xs:documentation>Alpha information that
allows the reference image to be reconstructed</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="1931IEC_ChromaticityRedX">
                        <xs:annotation>
                              <xs:documentation> value x of pair xy
specifying red colour according to 1031 IEC</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="1931IEC_ChromaticityRedY">
                        <xs:annotation>
                              <xs:documentation>value y of pair xy
specifying red colour according to 1031 IEC</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="1931IEC_ChromaticityGreenX">
                        <xs:annotation>
```

```
                              <xs:documentation>value x of pair xy
specifying green colour according to 1031 IEC</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="1931IEC_ChromaticityGreenY">
                        <xs:annotation>
                              <xs:documentation>value y of pair xy
specifying green colour according to 1031 IEC</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="1931IEC_ChromaticityBlueX">
                        <xs:annotation>
                              <xs:documentation>value x of pair xy
specifying blue colour according to 1031 IEC</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="1931IEC_ChromaticityBlueY">
                        <xs:annotation>
                              <xs:documentation>value y of pair xy
specifying green colour according to 1031 IEC</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="whitePointX">
                        <xs:annotation>
                              <xs:documentation>chromaticity of a computer
display's nominal white x value</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="whitePointY">
                        <xs:annotation>
                              <xs:documentation>chromaticity of a computer
display's nominal white y value</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="backgroundColour">
                        <xs:annotation>
                              <xs:documentation>solid colour for the
background of an image to be used when presenting the
image</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="resolutionX">
                        <xs:annotation>
                              <xs:documentation>number of pixels per
resolution unit in horizontal direction</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="resolutionY">
                        <xs:annotation>
                              <xs:documentation>number of pixels per
resolution unit in vertical direction</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="resolutionUnit">
                        <xs:annotation>
                              <xs:documentation>measure unit for measuring
resolution</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="blackWhiteInterpretation">
                        <xs:annotation>
                              <xs:documentation>photometric interpretation
in black and white based images</xs:documentation>
```

```
                        </xs:annotation>
                  </xs:enumeration>
            </xs:restriction>
      </xs:simpleType>
      <!--********************************* names for image properties,
extended group:  ********************** -->
      <xs:simpleType name="xclImageExtendedNames">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="timeLastMod">
                        <xs:annotation>
                              <xs:documentation>last modification date of
source object [Compatibility: PNG 1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="significantBits">
                        <xs:annotation>
                              <xs:documentation>number of bits that are
significant in the samples [Compatibility: PNG 1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="compressionFlag">
                        <xs:annotation>
                              <xs:documentation>[Compatibility: PNG
1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="filter">
                        <xs:annotation>
                              <xs:documentation>transformation applied to an
array of scanlines with the aim of improving their compressibility
[Compatibility: PNG 1.1]
                              </xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="interlace">
                        <xs:annotation>
                              <xs:documentation>algorithm for encoding
raster image data in multiple layers with the purpose of improved
transmission and image built up [Compatibility: PNG 1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="ICCProfileName">
                        <xs:annotation>
                              <xs:documentation>Name of the description of
the colour space (in the form of an International Color Consortium (ICC)
profile) to which the samples in the image conform [Compatibility: PNG
1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="suggestedPaletteName">
                        <xs:annotation>
                              <xs:documentation>name of a reduced palette
that may be used when the display device is not capable of displaying the
full range of colours in the image [Compatibility: PNG
1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="suggestedPaletteSampleDepth">
                        <xs:annotation>
                              <xs:documentation>sample depth of a suggested
palette [Compatibility: PNG 1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="suggestedPaletteRGB">
                        <xs:annotation>
                              <xs:documentation>palette with red, green and 
blue values; shall appear in rgb order [Compatibility: PNG 
1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="suggestedPaletteAlpha">
                        <xs:annotation>
                              <xs:documentation>alpha value for suggested 
palette [Compatibility: PNG 1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="suggestedPaletteFrequency">
                        <xs:annotation>
                              <xs:documentation>frequency of suggested 
palette  [Compatibility: PNG 1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="ICC-1/ICC-1A_RenderingIntent">
                        <xs:annotation>
                              <xs:documentation>suggestion for rendering an 
image according to ICC-1/ICC-1A [Compatibility: PNG 1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="histogram">
                        <xs:annotation>
                              <xs:documentation>approximate usage frequency 
of each colour in the palette index (rgbPalette) [Compatibility: PNG 
1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="textualDataKeyword">
                        <xs:annotation>
                              <xs:documentation>keyword for textual 
information associated with the image [Compatibility: PNG 
1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="textualDataString">
                        <xs:annotation>
                              <xs:documentation>Textual information 
associated with the image [Compatibility: PNG 1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="textualDataLanguage">
                        <xs:annotation>
                              <xs:documentation>language of textual 
informationassociated with the image [Compatibility: PNG 
1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="rowsPerStrip">
                        <xs:annotation>
                              <xs:documentation>number of rows in each strip 
except possibly the last strip  [Compatibility: TIFF 
6.0]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="artist">
                        <xs:annotation>
                              <xs:documentation>Person who created the image 
[Compatibility: TIFF 6.0]</xs:documentation>
```

```
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="cellHeight">
                        <xs:annotation>
                              <xs:documentation>The height of the dithering
or halftoning matrix used to create a dithered or
                                    halftoned bilevel file (y-axis)
[Compatibility: TIFF 6.0]     </xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="cellWidth">
                        <xs:annotation>
                              <xs:documentation>The width of the dithering
or halftoning matrix used to create a dithered or
                                    halftoned bilevel file (x-axis)
[Compatibility: TIFF 6.0]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="copyright">
                        <xs:annotation>
                              <xs:documentation>Copyright notice
[Compatibility: TIFF 6.0]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="grayResponseCurve">
                        <xs:annotation>
                              <xs:documentation>For greyscale data, the
optical density of each possible pixel value [Compatibility: TIFF
6.0]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
            </xs:restriction>
      </xs:simpleType>
      <!-- ***************************************************************
fixed labels for images
**************************************************************** -->
      <xs:simpleType name="xclImageValueLabels">
            <xs:union memberTypes="nm:imageTypeType nm:fillOrderType
nm:blackWhiteInterpretationType nm:compressionMethodType
nm:compressionFlagType nm:filterMethodType nm:interlaceMethodType
nm:truecolourChannelType nm:ICC-1_ICC-1A_RenderingIntentType
nm:unspecifiedType"/>
      </xs:simpleType>
      <xs:simpleType name="imageTypeType">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="grayscale">
                        <xs:annotation>
                              <xs:documentation>Image type for which each
pixel is a grayscale sample.</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="truecolour">
                        <xs:annotation>
                              <xs:documentation>Image type for which each
pixel is a red/green/blue triplet.</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="palette">
                        <xs:annotation>
                              <xs:documentation>Indexed-colour image type.
Each pixel is a palette index with a series of red/green/blue
triplets.</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="grayscaleAlpha">
                        <xs:annotation>
                              <xs:documentation>Each pixel is a grayscale
sample followed by an alpha sample [Compatibility: PNG
1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="truecolourAlpha">
                        <xs:annotation>
                              <xs:documentation>Each pixel is a truecolour
sample followed by an alpha sample [Compatibility: PNG
1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="bilevel">
                        <xs:annotation>
                              <xs:documentation>Image type for images that
can only store black and white image data[Compatibility: TIFF
6.0]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
            </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="fillOrderType">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="high-order">
                        <xs:annotation>
                              <xs:documentation>Pixels are arranged within a
byte such that pixels with lower column values are stored in the higher-
order bits of the byte.
                              </xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="low-order">
                        <xs:annotation>
                              <xs:documentation>Pixels are arranged within a
byte such that pixels with lower column values are stored in the lower-
order bits of the byte.
                              </xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
            </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="blackWhiteInterpretationType">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="whiteIsZero">
                        <xs:annotation>
                              <xs:documentation>photometric interpretation
for images based on black/white. 0 is interpreted as white
colour.</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="blackIsZero">
                        <xs:annotation>
                              <xs:documentation>photometric interpretation
for images based on black/white. 0 is interpreted as black
colour.</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
            </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="compressionMethodType">
```

```
            <xs:restriction base="xs:string">
                  <xs:enumeration value="zlibDeflateInflate">
                        <xs:annotation>
                              <xs:documentation>compression method using
zlib [Compatibility: PNG 1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="CCITT_Group3">
                        <xs:annotation>
                              <xs:documentation>1-dimensional modified
Huffman RLE encoding [Compatibility: TIFF 6.0]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="packBits">
                        <xs:annotation>
                              <xs:documentation>pack bits run length
compression [Compatibility: TIFF 6.0]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="uncompressed"/>
            </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="compressionFlagType">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="compressed">
                        <xs:annotation>
                              <xs:documentation>[Compatibility: PNG
1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="uncompressed">
                        <xs:annotation>
                              <xs:documentation>[Compatibility: PNG
1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
            </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="filterMethodType">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="adaptive">
                        <xs:annotation>
                              <xs:documentation>[Compatibility: PNG
1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
            </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="interlaceMethodType">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="adam7">
                        <xs:annotation>
                              <xs:documentation>[Compatibility: PNG
1.1]</xs:documentation>
                        </xs:annotation>
                  </xs:enumeration>
                  <xs:enumeration value="noInterlace"/>
            </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="truecolourChannelType">
            <xs:restriction base="xs:string">
                  <xs:enumeration value="truecolourRed">
                        <xs:annotation>
```

```
                    <xs:documentation>Red channel of a truecolour
image [Compatibility: PNG 1.1]</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="truecolourGreen">
                <xs:annotation>
                    <xs:documentation>Green channel of a
truecolour image [Compatibility: PNG 1.1]</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="truecolourBlue">
                <xs:annotation>
                    <xs:documentation>Blue channel of a truecolour
image [Compatibility: PNG 1.1]</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="ICC-1_ICC-1A_RenderingIntentType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="perceptual">
                <xs:annotation>
                    <xs:documentation>Rendering intent for images
preferring good adaptation to the output device gamut at the expense of
colorimetric accuracy, such as photographs [Compatibility: PNG
1.1]</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="relativeColorimetric">
                <xs:annotation>
                    <xs:documentation>Rendering intent for images
requiring colour appearance matching (relative to the output device white
point), such as logos [Compatibility: PNG 1.1]</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="saturation">
                <xs:annotation>
                    <xs:documentation>Rendering intent for images
preferring preservation of saturation at the expense of hue and lightness,
such as charts and graphs [Compatibility: PNG 1.1]
                    </xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="absoluteColorimetric">
                <xs:annotation>
                    <xs:documentation>Rendering intent for images
requiring preservation of absolute colorimetry, such as previews of images
destined for a different output device (proofs) [Compatibility: PNG
1.1]</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="unspecifiedType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="inch">
                <xs:annotation>
                    <xs:documentation>Measure of
length.</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="centimeter">
                <xs:annotation>
```

```
                    <xs:documentation>Measure of
length.</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="metre">
                <xs:annotation>
                    <xs:documentation>Measure of
length.</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="unknown">
                <xs:annotation>
                    <xs:documentation>label for any value that is
unknown</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

Appendix

Example XCEL: The PNG Specification

A png transferred to a XCEL description in general conforms to the following structure:

There is an item for every Chunk in the png file.

Each of these items contains

a symbol containing the length of the data field

a property, which indicates the chunk type, i.e. the name of the chunk described by the item, as given in the file format specification

an item describing the actual data

the checksum encoded as a 4 byte symbol

ordering items:

There are items without semantic meaning, used only for structuring purposes.

Containers, which indicate whether the content of an item is to be processed as ordered or unordered elements are identified by a capital „C“, i.e IDC01_I01. IDI03_C01

ordered elements have to appear in the file in the exact same order as listed in the xcel.

unordered elements can have any startposition in the file within the range given by the container item.

implicit items:

File properties which are only given in the file sepcification but not in the bitstream, are declared in a header outside the parsing area.

For the png this is the default physical dimension, which can be overriden by the pHYS chunk and the byte order, which is big Endian for the whole file.

```
<?xml version="1.0" encoding="UTF-8"?>
<XCELDocument xmlns="http://www.planets-
project.eu/xcl/schemas/xcelstructure"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      xsi:schemaLocation="http://www.planets-
project.eu/xcl/schemas/xcelstructure
      ItemLibs/XCELImageItems.xsd">

      <!-- this is an instance of the XCEL draft version 0  for a PNG
object (not complete)
      values indicating a length or position are interpreted as Bytes, if
not indicated otherwise-->

      <!--<select name="PNG"></select>-->

      <!--  XCEL header for PNGs -->
      <!--default for physical dimension of pixels, will be overidden if
file contains a pHYS chunk -->
      <!--  <item xsi:type="basicItem" identifier="IDH">
                  <item xsi:type="implicitImageItem" identifier="IDH_I01">
```

```
                        <item xsi:type="implicitItem"
identifier="IDH_I01_S01">
                              <name xsi:type="images">
                                    <value>x</value>
                                    <label>resolutionX</label>
                              </name>
                        </item>
                        <item xsi:type="implicitItem"
identifier="IDH_I01_S02">
                              <name xsi:type="images">
                                    <value>x</value>
                                    <label>resolutionY</label>
                              </name>
                        </item>
                        <item xsi:type="implicitItem"
identifier="IDH_I01_S03">
                              <name xsi:type="images">
                                    <value>unknown</value>
                                    <label>resolutionUnit</label>
                              </name>
                        </item>
                  </item>
            </item>-->
      <!-- end of header -->

      <item xsi:type="basicItem" identifier="IDC01" byteOrder="bigEndian">
            <name>PNG</name>
            <property xsi:type="basicProperty" identifier="ID01_S01"
                  originalName="signature">
                  <range>
                        <startposition
xsi:type="fixedPosition"></startposition>
                        <length xsi:type="fixedLength">8</length>
                  </range>
                  <value>137 80 78 71 13 10 26 10</value>
                  <name>signature</name>
            </property>
            <!-- Image header chunk: -->
            <item xsi:type="basicItem" identifier="ID01_I01"
originalName="IHDR">
                  <symbol xsi:type="basicSymbol" identifier="ID01_I01_S01"
                        originalName="chunkDataLength">
                        <range>
                              <startposition
xsi:type="fixedPosition"></startposition>
                              <length xsi:type="fixedLength">4</length>
                        </range>
                        <name>dataBlockLength</name>
                  </symbol>
                  <property xsi:type="basicProperty"
identifier="ID01_I01_S02"
                        originalName="chunkType">
                        <range>
                              <startposition
xsi:type="fixedPosition"></startposition>
                              <length xsi:type="fixedLength">4</length>
                        </range>
                        <value>73 72 68 82</value>
                        <name>dataBlockType</name>
                  </property>
                  <item xsi:type="basicItem" identifier="ID01_I01_I01"
                        originalName="chunkData">
                        <symbol xsi:type="basicImageSymbol"
                              identifier="ID01_I01_I01_S01"
originalName="width">
                              <range>
                                    <startposition xsi:type="fixedPosition">
                                          </startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>width</name>
                        </symbol>
                        <symbol xsi:type="basicImageSymbol"
                              identifier="ID01_I01_I01_S02"
originalName="height">
                              <range>
                                    <startposition xsi:type="fixedPosition">
                                          </startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>height</name>
                        </symbol>
                        <symbol xsi:type="selectiveImageSymbol"
                              identifier="ID01_I01_I01_S03"
originalName="bitDepth">
                              <range>
                                    <startposition xsi:type="fixedPosition">
                                          </startposition>
                                    <length
xsi:type="fixedLength">1</length>
                              </range>
                              <validValues>
                                    <keyValue>1</keyValue>
                                    <keyValue>2</keyValue>
                                    <keyValue>4</keyValue>
                                    <keyValue>6</keyValue>
                                    <keyValue>8</keyValue>
                                    <keyValue>16</keyValue>
                              </validValues>
                              <name>colourDepth</name>
                        </symbol>
                        <symbol xsi:type="selectiveImageSymbol"
                              identifier="ID01_I01_I01_S04"
originalName="colourType">
                              <range>
                                    <startposition xsi:type="fixedPosition">
                                          </startposition>
                                    <length
xsi:type="fixedLength">1</length>
                              </range>
                              <valueInterpretation>
                                    <valueLabel>greyscale</valueLabel>
                                    <keyValue>0</keyValue>
                                    <valueLabel>truecolour</valueLabel>
                                    <keyValue>2</keyValue>
                                    <valueLabel>indexedColour</valueLabel>
                                    <keyValue>3</keyValue>
                                    <valueLabel>greyscaleAlpha</valueLabel>
                                    <keyValue>4</keyValue>
                                    <valueLabel>truecolourAlpha</valueLabel>
                                    <keyValue>6</keyValue>
                              </valueInterpretation>
                              <name>imageType</name>
                        </symbol>
                        <symbol xsi:type="selectiveImageSymbol"
                              identifier="ID01_I01_I01_S05"
                              originalName="compressionMethod">
                              <range>
                                    <startposition xsi:type="fixedPosition">
                                          </startposition>
                                    <length
xsi:type="fixedLength">1</length>
                              </range>
                              <valueInterpretation>

      <valueLabel>zlibDeflateInflate</valueLabel>
                                    <keyValue>0</keyValue>
                              </valueInterpretation>
                              <name>compression</name>
                        </symbol>
                        <symbol xsi:type="selectiveImageSymbol"
                              identifier="ID01_I01_I01_S06"
originalName="filterMethod">
                              <range>
                                    <startposition xsi:type="fixedPosition">
                                          </startposition>
                                    <length
xsi:type="fixedLength">1</length>
                              </range>
                              <valueInterpretation>
                                    <valueLabel>adaptive</valueLabel>
                                    <keyValue>0</keyValue>
                              </valueInterpretation>
                              <name>filter</name>
                        </symbol>
                        <symbol xsi:type="selectiveImageSymbol"
                              identifier="ID01_I01_I01_S07"
                              originalName="interlaceMethod">
                              <range>
                                    <startposition xsi:type="fixedPosition">
                                          </startposition>
                                    <length
xsi:type="fixedLength">1</length>
                              </range>
                              <valueInterpretation>
                                    <valueLabel>adam7</valueLabel>
                                    <keyValue>1</keyValue>
                                    <valueLabel>nointerlace</valueLabel>
                                    <keyValue>0</keyValue>
                              </valueInterpretation>
                              <name>interlace</name>
                        </symbol>
                  </item>
                  <!-- end of ChunkData -->
                  <symbol xsi:type="basicSymbol" identifier="ID01_I01_S03"
                        originalName="crc">
                        <range>
                              <startposition
xsi:type="fixedPosition"></startposition>
                              <length xsi:type="fixedLength">4</length>
                        </range>
                        <name>checksum</name>
                  </symbol>
                  <name>header</name>
```

```
            </item>
            <!-- end of IDHR-Chunk -->
            <item xsi:type="basicItem" identifier="IDC02" ordered="false"
                  optional="true">
                  <!--IDAT start-->
                  <item xsi:type="basicItem" identifier="IDI02"
originalName="IDAT"
                        multiple="true" optional="true">

                        <symbol xsi:type="fileStructureSymbol"
identifier="IDI02_S01"
                              originalName="chunkDataLength">
                              <range>
                                    <startposition
xsi:type="expected">33</startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>dataBlockLength</name>
                        </symbol>

                        <property xsi:type="fileStructureProperty"
identifier="IDI02_S02"
                              originalName="ChunkType">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>73 68 65 84</value>
                              <name>dataBlockType</name>
                        </property>

                        <symbol xsi:type="fileStructureSymbol"
identifier="IDI02_S03">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length xsi:type="referencedLength"
ref="IDI02_S01">
                                    </length>
                              </range>
                              <name>dataBlock</name>
                        </symbol>

                        <!-- end of Chunk Data -->
                        <symbol xsi:type="basicSymbol"
identifier="IDI02_S04"
                              originalName="crc">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>checksum</name>
                        </symbol>
                  </item>
                  <!-- end of IDAT-Chunk -->

                  <!--gAMMA start-->
                  <item xsi:type="basicItem" optional="true"
identifier="IDI03"
                        originalName="gAMA">
                        <symbol xsi:type="fileStructureProperty"
identifier="IDI03_S01"
                              originalName="chunkDataLength">
                              <range>
                                    <startposition
xsi:type="expected">33</startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>0 0 0 4</value>
                              <name>dataBlockLength</name>
                        </symbol>
                        <property xsi:type="fileStructureProperty"
identifier="IDI03_S02"
                              originalName="ChunkType">
                              <range>
                                    <startposition xsi:type="sequential">
                                    </startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>103 65 77 65</value>
                              <name>dataBlockType</name>
                        </property>
                        <item xsi:type="fileStructureItem"
identifier="IDI03_I01"
                              originalName="chunkData">
                              <symbol xsi:type="basicImageSymbol"
identifier="IDI03_I01_S01"
                                    originalName="imageGamma">
                                    <range>
                                          <startposition
xsi:type="sequential">
                                          </startposition>
                                          <length
xsi:type="fixedLength">4</length>
                                    </range>
                                    <name>gamma</name>
                              </symbol>
                              <name>dataBlock</name>
                        </item>
                        <!-- end of Chunk Data -->
                        <symbol xsi:type="basicSymbol"
identifier="IDI03_S03"
                              originalName="crc">
                              <range>
                                    <startposition xsi:type="sequential">
                                    </startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>checksum</name>
                        </symbol>
                  </item>
                  <!-- end of gAMA-Chunk -->
                  <!--PLTE start-->
                  <item xsi:type="basicItem" identifier="IDI04"
optional="true"
                        originalName="PLTE">
                        <symbol xsi:type="fileStructureSymbol"
identifier="IDI04_S01"
                              originalName="chunkDataLength">
                              <range>
                                    <startposition
xsi:type="expected">33</startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>dataBlockLength</name>
                        </symbol>
                        <property xsi:type="fileStructureProperty"
identifier="IDI04_S02"
                              originalName="ChunkType">
                              <range>
                                    <startposition xsi:type="sequential">
                                    </startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>80 76 84 69</value>
                              <name>dataBlockType</name>
                        </property>
                        <item xsi:type="imageItem" identifier="IDI04_I01"
                              originalName="chunkData">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length xsi:type="referencedLength"
ref="IDI04_S01">
                                          </length>
                              </range>
                              <symbol xsi:type="basicImageSymbol"
identifier="IDI04_I01_S01"
                                    originalName="rgb" multiple="true">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">3</length>
                                    </range>
                                    <name>rgb</name>
                              </symbol>
                              <name>rgbPalette</name>
                        </item>
                        <!-- end of Chunk Data -->
                        <symbol xsi:type="basicSymbol"
identifier="IDI04_S03"
                              originalName="crc">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>checksum</name>
                        </symbol>

                  </item>
                  <!-- end of PLTE-Chunk -->
                  <!--sBIT Start-->
```

```
            <item xsi:type="basicItem" optional="true"
identifier="IDI05"
                originalName="sBIT">
                <symbol xsi:type="fileStructureSymbol"
identifier="IDI05_S01"
                    originalName="chunkDataLength">
                    <range>
                        <startposition
xsi:type="expected">33</startposition>
                        <length
xsi:type="fixedLength">4</length>
                    </range>
                    <name>dataBlockLength</name>
                </symbol>
                <property xsi:type="fileStructureProperty"
identifier="IDI05_S02"
                    originalName="ChunkType">
                    <range>
                        <startposition
xsi:type="sequential"></startposition>
                        <length
xsi:type="fixedLength">4</length>
                    </range>
                    <value>115 66 73 84</value>
                    <name>dataBlockType</name>
                </property>
                <item xsi:type="fileStructureItem"
identifier="IDI05_I01"
                    originalName="chunkData">
                    <symbol xsi:type="basicImageSymbol"
                        identifier="IDI01_I05_I01_S01"
                        originalName="significant Bits">
                        <range>
                            <startposition
xsi:type="sequential"></startposition>
                            <length
xsi:type="referencedLength" ref="IDI05_S01">
                            </length>
                        </range>
                        <name>significantBits</name>
                    </symbol>
                    <name>dataBlock</name>
                </item>
                <!-- end of Chunk Data -->
                <symbol xsi:type="basicSymbol"
identifier="IDI05_S03"
                    originalName="crc">
                    <range>
                        <startposition xsi:type="sequential">
                        </startposition>
                        <length
xsi:type="fixedLength">4</length>
                    </range>
                    <name>checksum</name>
                </symbol>
            </item>
            <!-- end of sBIT-Chunk -->
            <!--tIME start-->
            <item xsi:type="basicItem" identifier="IDI06"
optional="true"
                originalName="tIME">
```

```
                        <property xsi:type="fileStructureProperty"
identifier="IDI06_S01"
                              originalName="chunkDataLength">
                              <range>
                                    <startposition
xsi:type="expected">33</startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>0 0 0 7</value>
                              <name>dataBlockLength</name>
                        </property>
                        <property xsi:type="fileStructureProperty"
identifier="IDI06_S02"
                              originalName="ChunkType">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>116 73 77 69</value>
                              <name>dataBlockType</name>
                        </property>
                        <item xsi:type="fileStructureItem"
identifier="IDI06_I01"
                              originalName="chunkData">
                              <item xsi:type="imageItem"
identifier="IDI06_I01_I01">
                                    <symbol identifier="IDI06_I01_I01_S01"
originalName="year"
                                          xsi:type="basicSymbol">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">2</length>
                                          </range>
                                          <name>yyear</name>
                                    </symbol>
                                    <symbol identifier="IDI06_I01_I01_S02"
                                          originalName="month"
xsi:type="basicSymbol">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">1</length>
                                          </range>
                                          <name>month</name>
                                    </symbol>
                                    <symbol identifier="IDI06_I01_I01_S03"
originalName="day"
                                          xsi:type="basicSymbol">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">1</length>
```

```
                                          </range>
                                          <name>day</name>
                                    </symbol>
                                    <symbol identifier="IDI06_I01_I01_S04"
originalName="hour"
                                          xsi:type="basicSymbol">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">1</length>
                                          </range>
                                          <name>hour</name>
                                    </symbol>
                                    <symbol identifier="IDI06_I01_I01_S05"
                                          originalName="minute"
xsi:type="basicSymbol">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">1</length>
                                          </range>
                                          <name>minute</name>
                                    </symbol>
                                    <symbol identifier="IDI06_I01_I01_S06"
                                          originalName="second"
xsi:type="basicSymbol">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">1</length>
                                          </range>
                                          <name>second</name>
                                    </symbol>
                                    <name>timeLastMod</name>
                              </item>
                              <!--end of imageItem timeLastMod -->
                              <name>dataBlock</name>
                        </item>
                        <!-- end of Chunk Data -->
                        <symbol xsi:type="basicSymbol"
identifier="IDI06_S03"
                              originalName="crc">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>checksum</name>
                        </symbol>
                  </item>
                  <!-- end of tIME-Chunk -->
                  <!--bKGD start-->
                  <item xsi:type="basicItem" identifier="IDI07"
optional="true"
                        originalName="bKGD" ordered="true">
                        <item xsi:type="basicItem" identifier="IDI07_C01"
ordered="false">

                              <item xsi:type="imageItem"
identifier="IDI07_C01_I01"
                                    optional="true">
                                    <property
xsi:type="fileStructureProperty"
                                          identifier="IDI07_C01_I01_S01"
                                          originalName="chunkDataLength">
                                          <range>
                                                <startposition
xsi:type="expected">
                                                      33</startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>
                                          <value>0 0 0 2</value>
                                          <name>dataBlockLength</name>
                                    </property>
                                    <property
xsi:type="fileStructureProperty"
                                          identifier="IDI07_C01_I01_S02"
                                          originalName="ChunkType">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>
                                          <value>98 75 71 68</value>
                                          <name>dataBlockType</name>
                                    </property>
                                    <symbol identifier="IDI07_C01_I01_S03"
                                          xsi:type="basicImageSymbol">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">2</length>
                                          </range>
                                          <name>greyscale</name>
                                    </symbol>
                                    <name>backgroundColour</name>
                              </item>
                              <!-- end of backgroundcolor - greyscale
definition-->
                              <item xsi:type="basicItem"
identifier="IDI07_C01_I02"
                                    optional="true">
                                    <property
xsi:type="fileStructureProperty"
                                          identifier="IDI07_C01_I02_S01"
                                          originalName="chunkDataLength">
                                          <range>
                                                <startposition
xsi:type="expected">
                                                      33</startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>
                                          <value>0 0 0 6</value>
                                          <name>dataBlockLength</name>
                                    </property>
                                    <property
xsi:type="fileStructureProperty"
                                          identifier="IDI07_C01_I02_S02"
                                          originalName="ChunkType">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>
                                          <value>98 75 71 68</value>
                                          <name>dataBlockType</name>
                                    </property>
                                    <item xsi:type="imageItem"
identifier="IDI07_C01_I02_I01">
                                          <symbol
identifier="IDI07_C01_I02_I01_S01"
                                                xsi:type="basicImageSymbol">
                                                <range>
                                                      <startposition
xsi:type="sequential">
                                                            </startposition>
                                                      <length
xsi:type="fixedLength">2</length>
                                                </range>
                                                <name>truecolourRed</name>
                                          </symbol>
                                          <symbol
identifier="IDI07_C01_I02_I01_S02"
                                                xsi:type="basicImageSymbol">
                                                <range>
                                                      <startposition
xsi:type="sequential">
                                                            </startposition>
                                                      <length
xsi:type="fixedLength">2</length>
                                                </range>
                                                <name>truecolourGreen</name>
                                          </symbol>
                                          <symbol
identifier="IDI07_C01_I02_I01_S03"
                                                xsi:type="basicImageSymbol">
                                                <range>
                                                      <startposition
xsi:type="sequential">
                                                            </startposition>
                                                      <length
xsi:type="fixedLength">2</length>
                                                </range>
                                                <name>truecolourBlue</name>
                                          </symbol>
                                          <name>backgroundColour</name>
                                    </item>
                                    <!-- end of chunkData -->
                              </item>
                              <!-- end of background Colour truecolor
definition -->
                              <item xsi:type="imageItem"
identifier="IDI07_C01_I03"
                                    optional="true">
                                    <property
xsi:type="fileStructureProperty"
                                          identifier="IDI07_C01_I03_S01"
                                          originalName="chunkDataLength">
                                          <range>
                                                <startposition
xsi:type="expected">
                                                      33</startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>
                                          <value>0 0 0 1</value>
                                          <name>dataBlockLength</name>
                                    </property>
                                    <property
xsi:type="fileStructureProperty"
                                          identifier="IDI07_C01_I03_S02"
                                          originalName="ChunkType">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>
                                          <value>98 75 71 68</value>
                                          <name>dataBlockType</name>
                                    </property>
                                    <symbol identifier="IDI07_C01_I03_S03"
                                          xsi:type="basicImageSymbol">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">1</length>
                                          </range>
                                          <name>palette</name>
                                    </symbol>
                                    <name>backgroundColour</name>
                              </item>
                              <!-- end of background color index color
definition -->
                        </item>
                        <!--end of unordered Item-->
                        <symbol xsi:type="basicSymbol"
identifier="IDI07_S01"
                              originalName="crc">
                              <range>
                                    <startposition xsi:type="sequential">
                                    </startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>checksum</name>
                        </symbol>
                  </item>
                  <!-- end of bKGD-Chunk -->

                  <!--start of hist-->
                  <item xsi:type="basicItem" identifier="IDI08"
optional="true"
                        originalName="hIST">
                        <symbol xsi:type="fileStructureSymbol"
identifier="IDI08_S01"
                              originalName="chunkDataLength">
                              <range>
                                    <startposition
xsi:type="expected">33</startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>dataBlockLength</name>
                        </symbol>
                        <property xsi:type="fileStructureProperty"
identifier="IDI08_S02"
                              originalName="ChunkType">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>104 73 83 84</value>
                              <name>dataBlockType</name>
                        </property>
                        <item xsi:type="fileStructureItem"
identifier="IDI08_I01"
                              originalName="chunkData">
                              <symbol xsi:type="basicImageSymbol"
identifier="IDI08_I01_S01"
                                    originalName="frequency">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="referencedLength" ref="IDI08_S01">
                                                </length>
                                    </range>
                                    <name>histogram</name>
                              </symbol>
                              <name>dataBlock</name>
                        </item>
                        <!-- end of Chunk Data -->
                        <symbol xsi:type="basicSymbol"
identifier="IDI08_S03"
                              originalName="crc">
                              <range>
                                    <startposition xsi:type="sequential">
                                    </startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>checksum</name>
                        </symbol>
                  </item>
                  <!-- end of hIST-Chunk -->

                  <!--start of tEXt-->
                  <item xsi:type="basicItem" identifier="IDI09"
originalName="tEXt"
                        optional="true">

                        <symbol xsi:type="fileStructureSymbol"
identifier="IDI09_S01"
                              originalName="chunkDataLength">
                              <range>
                                    <startposition
xsi:type="expected"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>dataBlockLength</name>
                        </symbol>
                        <property xsi:type="fileStructureProperty"
identifier="IDI09_S02"
                              originalName="ChunkType">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>116 69 88 116</value>
                              <name>dataBlockType</name>
                        </property>

                        <item xsi:type="basicItem" identifier="IDI09_I01">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length xsi:type="referencedLength"
ref="IDI09_S01">
                                          </length>
                              </range>

                              <symbol xsi:type="selectiveImageSymbol"
multiple="true"
                                    identifier="IDI09_I01_S01"
interpretation="Latin1">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <validValues xsi:type="valueRange">
                                          <startRange>1</startRange>
                                          <endRange>255</endRange>
                                    </validValues>
                                    <name>textualDataKeyword</name>
                              </symbol>

                              <property xsi:type="basicProperty"
identifier="IDI09_I01_S02">
                                    <range>
                                          <startposition
xsi:type="expected"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <value>0</value>
                                    <name>delimiter</name>
                              </property>

                              <symbol xsi:type="selectiveImageSymbol"
multiple="true"
                                    interpretation="Latin1"
identifier="IDI09_I01_S03">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <validValues xsi:type="valueRange">
                                          <startRange>0</startRange>
                                          <endRange>255</endRange>
                                    </validValues>
                                    <name>textualDataString</name>
                              </symbol>
                        </item> <!-- end of Chunk Data -->

                        <symbol xsi:type="basicSymbol"
identifier="IDI09_S03"
                              originalName="crc">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>

                              <name>checksum</name>
                        </symbol>
                  </item> <!-- end of tEXt-Chunk -->

                  <!--start of iTXt-->
                  <item xsi:type="basicItem" identifier="IDI10"
originalName="iTXt"
                        optional="true">

                        <symbol xsi:type="fileStructureSymbol"
identifier="IDI10_S01"
                              originalName="chunkDataLength">
                              <range>
                                    <startposition
xsi:type="expected">33</startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>dataBlockLength</name>
                        </symbol>
                        <property xsi:type="fileStructureProperty"
identifier="IDI10_S02"
                              originalName="ChunkType">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>105 84 88 116</value>
                              <name>dataBlockType</name>
                        </property>

                        <item xsi:type="basicItem" identifier="IDI10_I01">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length xsi:type="referencedLength"
ref="IDI10_S01">
                                          </length>
                              </range>

                              <symbol xsi:type="selectiveImageSymbol"
multiple="true"
                                    identifier="IDI10_I01_S01"
interpretation="utf-8">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <validValues xsi:type="valueRange">
                                          <startRange>1</startRange>
                                          <endRange>255</endRange>
                                    </validValues>
                                    <name>textualDataKeyword</name>
                              </symbol>
                              <property xsi:type="basicProperty"
identifier="IDI10_I01_S02">
                                    <range>
                                          <startposition
xsi:type="expected">33</startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <value>0</value>
                                    <name>delimiter</name>
                              </property>

                              <symbol xsi:type="selectiveImageSymbol"
                                    identifier="IDI10_I01_S03"
                                    originalName="compressionFlag">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <valueInterpretation>

      <valueLabel>compressed</valueLabel>
                                          <keyValue>1</keyValue>

      <valueLabel>uncompressed</valueLabel>
                                          <keyValue>0</keyValue>
                                    </valueInterpretation>
                                    <name>compression</name>
                              </symbol>
                              <symbol xsi:type="selectiveImageSymbol"
                                    identifier="IDI10_I01_S04"
                                    originalName="compressionMethod">
```

```
                                   <range>
                                         <startposition
xsi:type="sequential"></startposition>
                                         <length
xsi:type="fixedLength">1</length>
                                   </range>
                                   <valueInterpretation>

      <valueLabel>zlibDeflateInflate</valueLabel>
                                         <keyValue>0</keyValue>
                                   </valueInterpretation>
                                   <name>compression</name>
                             </symbol>
                             <symbol xsi:type="selectiveImageSymbol"
multiple="true"
                                   identifier="IDI10_I01_S05"
interpretation="iso-646">
                                   <range>
                                         <startposition
xsi:type="sequential"></startposition>
                                         <length
xsi:type="fixedLength">1</length>
                                   </range>
                                   <validValues xsi:type="valueRange">
                                         <startRange>1</startRange>
                                         <endRange>255</endRange>
                                   </validValues>
                                   <name>textualDataLanguage</name>
                             </symbol>
                             <property xsi:type="basicProperty"
identifier="IDI10_I01_S06">
                                   <range>
                                         <startposition
xsi:type="expected">33</startposition>
                                         <length
xsi:type="fixedLength">1</length>
                                   </range>
                                   <value>0</value>
                                   <name>delimiter</name>
                             </property>
                             <symbol xsi:type="basicImageSymbol"
multiple="true"
                                   interpretation="utf-8"
identifier="IDI10_I01_S07">
                                   <range>
                                         <startposition
xsi:type="sequential"></startposition>
                                         <length
xsi:type="fixedLength">1</length>
                                   </range>

                                   <name>textualDataKeyword</name>
                             </symbol>
                             <property xsi:type="basicProperty"
identifier="IDI10_I01_S08">
                                   <range>
                                         <startposition
xsi:type="expected">33</startposition>
                                         <length
xsi:type="fixedLength">1</length>
                                   </range>
                                   <value>0</value>
```

```
                                    <name>delimiter</name>
                              </property>
                              <symbol xsi:type="basicImageSymbol"
multiple="true"
                                    interpretation="utf-8"
identifier="IDI10_I01_S09">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <name>textualDataKeyword</name>
                              </symbol>
                        </item> <!-- end of Chunk Data -->

                        <symbol xsi:type="basicSymbol"
identifier="IDI10_S03"
                              originalName="crc">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>

                              <name>checksum</name>
                        </symbol>
                  </item> <!-- end of iTXt-Chunk -->

                  <!--start of zTXt-->
                  <item xsi:type="basicItem" identifier="IDI11"
originalName="zTXt"
                        optional="true">

                        <symbol xsi:type="fileStructureSymbol"
identifier="IDI11_S01"
                              originalName="chunkDataLength">
                              <range>
                                    <startposition
xsi:type="expected">33</startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>dataBlockLength</name>
                        </symbol>
                        <property xsi:type="fileStructureProperty"
identifier="IDI11_S02"
                              originalName="ChunkType">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>122 84 88 116</value>
                              <name>dataBlockType</name>
                        </property>

                        <item xsi:type="basicItem" identifier="IDI11_I01">
                              <range>
```

```
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length xsi:type="referencedLength"
ref="IDI11_S01">
                                          </length>
                              </range>

                              <symbol xsi:type="selectiveImageSymbol"
multiple="true"
                                    identifier="IDI11_I01_S01"
interpretation="Latin1">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <validValues xsi:type="valueRange">
                                          <startRange>1</startRange>
                                          <endRange>255</endRange>
                                    </validValues>
                                    <name>textualDataKeyword</name>
                              </symbol>
                              <property xsi:type="basicProperty"
identifier="IDI11_I01_S02">
                                    <range>
                                          <startposition
xsi:type="expected">33</startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <value>0</value>
                                    <name>delimiter</name>
                              </property>
                              <symbol xsi:type="selectiveImageSymbol"
                                    identifier="IDI01_I15_I01_S04"
                                    originalName="compressionMethod">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <valueInterpretation>

      <valueLabel>zlibDeflateInflate</valueLabel>
                                          <keyValue>0</keyValue>
                                    </valueInterpretation>
                                    <name>compression</name>
                              </symbol>
                              <symbol xsi:type="selectiveImageSymbol"
multiple="true"
                                    interpretation="Latin1"
identifier="IDI11_I01_S03">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <validValues xsi:type="valueRange">
                                          <startRange>1</startRange>
                                          <endRange>255</endRange>
                                    </validValues>
                                    <name>compressedText</name>
                              </symbol>
                        </item> <!-- end of Chunk Data -->

                        <symbol xsi:type="basicSymbol"
identifier="IDI11_S03"
                              originalName="crc">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>

                              <name>checksum</name>
                        </symbol>
                  </item> <!-- end of zTXt-Chunk -->

                  <!-- start of tRNS -->
                  <item xsi:type="basicItem" identifier="IDI12"
optional="true"
                        originalName="tRNS" ordered="true">

                        <item xsi:type="imageItem" identifier="IDI12_C01"
ordered="false">

                              <item xsi:type="imageItem"
identifier="IDI12_C01_I01"
                                    optional="true">
                                    <property
xsi:type="fileStructureProperty"
                                          identifier="IDI12_C01_I01_S01"
                                          originalName="chunkDataLength">
                                          <range>
                                                <startposition
xsi:type="expected">
                                                      33</startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>
                                          <value>0 0 0 2</value>
                                          <name>dataBlockLength</name>
                                    </property>

                                    <property
xsi:type="fileStructureProperty"
                                          identifier="IDI12_C01_I01_S02"
                                          originalName="ChunkType">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>
                                          <value>116 82 78 83</value>
                                          <name>dataBlockType</name>
                                    </property>
                                    <symbol identifier="IDI12_C01_I01_S03"
                                          xsi:type="basicImageSymbol">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">2</length>
                                          </range>
                                          <name>greyscale</name>
                                    </symbol>
                                    <name>transparency</name>
                              </item>
                              <!-- end of transparency - greyscale
definition-->

                              <item xsi:type="imageItem"
identifier="IDI12_C01_I02"
                                    optional="true">
                                    <property
xsi:type="fileStructureProperty"
                                          identifier="IDI12_C01_I02_S01"
                                          originalName="chunkDataLength">
                                          <range>
                                                <startposition
xsi:type="expected">
                                                      33</startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>
                                          <value>0 0 0 6</value>
                                          <name>dataBlockLength</name>
                                    </property>
                                    <property
xsi:type="fileStructureProperty"
                                          identifier="IDI12_C01_I02_S02"
                                          originalName="ChunkType">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>
                                          <value>116 82 78 83</value>
                                          <name>dataBlockType</name>
                                    </property>
                                    <item xsi:type="imageItem"
identifier="IDI12_C01_I02_I01">
                                          <range>
                                                <startposition
xsi:type="sequential"/>
                                                <length
xsi:type="fixedLength">6</length>
                                          </range>

                                          <symbol
identifier="IDI12_C01_I02_I01_S01"
                                                xsi:type="basicImageSymbol">
                                                <range>
                                                      <startposition
xsi:type="sequential">
                                                            </startposition>
                                                      <length
xsi:type="fixedLength">2</length>
                                                <\/range>
                                                <name>truecolourRed</name>
                                          </symbol>
                                          <symbol
identifier="IDI12_C01_I02_I01_S02"
                                                xsi:type="basicImageSymbol">
                                                <range>
                                                      <startposition
xsi:type="sequential">
                                                            </startposition>
                                                      <length
xsi:type="fixedLength">2</length>
                                                </range>
                                                <name>truecolourGreen</name>
                                          </symbol>
                                          <symbol
identifier="IDI12_C01_I02_I01_S03"
                                                xsi:type="basicImageSymbol">
                                                <range>
                                                      <startposition
xsi:type="sequential">
                                                            </startposition>
                                                      <length
xsi:type="fixedLength">2</length>
                                                </range>
                                                <name>truecolourBlue</name>
                                          </symbol>
                                          <name>truecolour</name>
                                    </item>
                                    <!-- end of chunkData -->
                                    <name>transparency</name>
                              </item>
                              <!-- end of background Colour truecolor
definition -->
                              <item xsi:type="imageItem"
identifier="IDI12_C01_I03"
                                    optional="true">
                                    <symbol xsi:type="fileStructureProperty"
                                          identifier="IDI12_C01_I03_S01"
                                          originalName="chunkDataLength">
                                          <range>
                                                <startposition
xsi:type="expected">
                                                      33</startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>

                                          <name>dataBlockLength</name>
                                    </symbol>
                                    <property
xsi:type="fileStructureProperty"
                                          identifier="IDI12_C01_I03_S02"
                                          originalName="ChunkType">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
```

```
                                          </range>
                                          <value>116 82 78 83</value>
                                          <name>dataBlockType</name>
                                    </property>

                                    <item xsi:type="fileStructureItem"
                                          identifier="IDI12_C01_I03_I01"
optional="true">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="referencedLength"

      ref="IDI12_C01_I03_S01"/>
                                          </range>

                                          <symbol
xsi:type="selectiveImageSymbol"

      identifier="IDI12_C01_I03_I01_S0"
                                                multiple="true">
                                                <range>
                                                      <startposition
xsi:type="sequential"/>

                                                      <length
xsi:type="fixedLength">1</length>
                                                </range>
                                                <validValues
xsi:type="valueRange">

      <startRange>0</startRange>

      <endRange>255</endRange>
                                                </validValues>
                                                <name>palette</name>
                                          </symbol>
                                          <name>dataBlock</name>

                                    </item>
                                    <name>transparency</name>
                              </item>
                              <!-- end of transperency color index color
definition -->
                              <name>transparency</name>
                        </item>
                        <!--end of unordered Item-->
                        <symbol xsi:type="basicSymbol"
identifier="IDI12_S01"
                              originalName="crc">
                              <range>
                                    <startposition xsi:type="sequential">
                                    </startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>checksum</name>
                        </symbol>
                  </item>
                  <!-- end of tRNS-Chunk -->
                  <!--start of cHRM-->
                  <item xsi:type="basicItem" identifier="IDI13"
originalName="cHRM"
                        optional="true">
                        <symbol xsi:type="fileStructureSymbol"
identifier="IDI13_S01"
                              originalName="chunkDataLength">
                              <range>
                                    <startposition
xsi:type="expected">33</startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>dataBlockLength</name>
                        </symbol>
                        <property xsi:type="fileStructureProperty"
identifier="IDI13_S02"
                              originalName="ChunkType">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>99 72 82 77</value>
                              <name>dataBlockType</name>
                        </property>

                        <item xsi:type="basicItem" identifier="IDI13_I01">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length xsi:type="referencedLength"
ref="IDI13_S01">
                                          </length>
                              </range>

                              <item xsi:type="basicItem"
identifier="IDI13_I01_I01">
                                    <symbol xsi:type="basicImageSymbol"
                                          identifier="IDI13_I01_I01_S01">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>
                                          <name>whitePointX</name>
                                    </symbol>
                                    <symbol xsi:type="basicImageSymbol"
                                          identifier="IDI13_I01_I01_S02">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>
                                          <name>whitePointY</name>
                                    </symbol>
                              </item>
                              <item xsi:type="basicItem"
identifier="IDI13_I01_I02">
                                    <symbol xsi:type="basicImageSymbol"
                                          identifier="IDI13_I01_I02_S01">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>

      <name>1931IEC_ChromaticityRedX</name>
                                    </symbol>
                                    <symbol xsi:type="basicImageSymbol"
                                          identifier="IDI13_I01_I02_S02">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>

      <name>1931IEC_ChromaticityRedY</name>
                                    </symbol>
                              </item>
                              <item xsi:type="basicItem"
identifier="IDI13_I01_I03">
                                    <symbol xsi:type="basicImageSymbol"
                                          identifier="IDI13_I01_I03_S01">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>

      <name>1931IEC_ChromaticityGreenX</name>
                                    </symbol>
                                    <symbol xsi:type="basicImageSymbol"
                                          identifier="IDI13_I01_I03_S02">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>

      <name>1931IEC_ChromaticityGreenY</name>
                                    </symbol>
                              </item>
                              <item xsi:type="basicItem"
identifier="IDI13_I01_I04">
                                    <symbol xsi:type="basicImageSymbol"
                                          identifier="IDI13_I01_I04_S01">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>

      <name>1931IEC_ChromaticityBlueX</name>
                                    </symbol>
                                    <symbol xsi:type="basicImageSymbol"
                                          identifier="IDI13_I01_I04_S02">
                                          <range>
                                                <startposition
xsi:type="sequential">
                                                      </startposition>
                                                <length
xsi:type="fixedLength">4</length>
                                          </range>

      <name>1931IEC_ChromaticityBlueY</name>
                                    </symbol>
                              </item>

                        </item> <!-- end of Chunk Data -->

                        <symbol xsi:type="basicSymbol"
identifier="IDI13_S03"
                              originalName="crc">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>

                              <name>checksum</name>
                        </symbol>
                  </item> <!-- end of  cHRM-Chunk -->

                  <!--start of iCCP-->
                  <item xsi:type="basicItem" identifier="IDI14"
originalName="iCCP"
                        optional="true">
                        <override ref="IDI03"></override>
                        <override ref="IDI13"></override>
                        <symbol xsi:type="fileStructureSymbol"
identifier="IDI14_S01"
                              originalName="chunkDataLength">
                              <range>
                                    <startposition
xsi:type="expected">33</startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>dataBlockLength</name>
                        </symbol>
                        <property xsi:type="fileStructureProperty"
identifier="IDI14_S02"
                              originalName="ChunkType">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>105 67 67 80</value>
                              <name>dataBlockType</name>
                        </property>

                        <item xsi:type="basicItem" identifier="IDI14_I01">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length xsi:type="referencedLength"
ref="IDI14_S01">
                                          </length>
                              </range>

                              <symbol xsi:type="selectiveImageSymbol"
multiple="true"
                                    identifier="IDI14_I01_S01"
interpretation="Latin1">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <validValues xsi:type="valueRange">
                                          <startRange>1</startRange>
                                          <endRange>255</endRange>
                                    </validValues>
                                    <name>textualDataKeyword</name>
                              </symbol>
                              <property xsi:type="basicProperty"
identifier="IDI14_I01_S02">
                                    <range>
                                          <startposition
xsi:type="expected">33</startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <value>0</value>
                                    <name>delimiter</name>
                              </property>

                              <symbol xsi:type="selectiveImageSymbol"
                                    identifier="IDI14_I01_S03"
                                    originalName="compressionMethod">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <valueInterpretation>

      <valueLabel>zlibDeflateInflate</valueLabel>
                                          <keyValue>0</keyValue>
                                    </valueInterpretation>
                                    <name>compression</name>
                              </symbol>
                              <symbol xsi:type="basicImageSymbol"
multiple="true"
                                    interpretation="utf-8"
identifier="IDI14_I01_S04">
                                    <range>
```

```
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="undefinedLength"></length>
                                    </range>
                                    <name>compressedText</name>
                              </symbol>
                        </item> <!-- end of Chunk Data -->

                        <symbol xsi:type="basicSymbol"
identifier="IDI14_S03"
                              originalName="crc">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>

                              <name>checksum</name>
                        </symbol>
                  </item> <!-- end of  iCCP-Chunk -->

                  <!--start of sRGB-->
                  <item xsi:type="basicItem" identifier="IDI15"
originalName="sRGB"
                        optional="true">
                        <override ref="IDI03"></override>
                        <override ref="IDI13"/>
                        <symbol xsi:type="fileStructureSymbol"
identifier="IDI15_S01"
                              originalName="chunkDataLength">
                              <range>
                                    <startposition
xsi:type="expected">33</startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>dataBlockLength</name>
                        </symbol>
                        <property xsi:type="fileStructureProperty"
identifier="IDI15_S02"
                              originalName="ChunkType">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>115 82 71 66</value>
                              <name>dataBlockType</name>
                        </property>

                        <item xsi:type="basicItem" identifier="IDI15_I01">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">1</length>
                              </range>

                              <symbol xsi:type="selectiveImageSymbol"
```

```
                                    identifier="IDI15_I01_S01">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <valueInterpretation>

      <valueLabel>perceptual</valueLabel>
                                          <keyValue>0</keyValue>

      <valueLabel>relativeColorimetric</valueLabel>
                                          <keyValue>1</keyValue>

      <valueLabel>saturation</valueLabel>
                                          <keyValue>2</keyValue>

      <valueLabel>absoluteColorimetric</valueLabel>
                                          <keyValue>3</keyValue>
                                    </valueInterpretation>

                                    <name>ICC-1/ICC-
1A_RenderingIntent</name>
                              </symbol>

                        </item><!-- end of Chunk Data -->

                        <symbol xsi:type="basicSymbol"
identifier="IDI15_S03"
                              originalName="crc">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>

                              <name>checksum</name>
                        </symbol>
                  </item> <!-- end of  sRGB-Chunk -->

                  <!--start of pHYS-->
                  <item xsi:type="basicItem" identifier="IDI16"
originalName="pHYS"
                        optional="true">

                        <override ref="IDH_I01"/>

                        <symbol xsi:type="fileStructureSymbol"
identifier="IDI16_S01"
                              originalName="chunkDataLength">
                              <range>
                                    <startposition
xsi:type="expected">33</startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>dataBlockLength</name>
                        </symbol>
                        <property xsi:type="fileStructureProperty"
identifier="IDI16_S02"
                              originalName="ChunkType">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>112 72 89 115</value>
                              <name>dataBlockType</name>
                        </property>

                        <item xsi:type="basicItem" identifier="IDI16_I01">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">9</length>
                              </range>

                              <symbol xsi:type="basicImageSymbol"
                                    identifier="IDI16_I01_S01">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">4</length>
                                    </range>
                                    <name>resolutionX</name>
                              </symbol>

                              <symbol xsi:type="basicImageSymbol"
                                    identifier="IDI16_I01_S02">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">4</length>
                                    </range>
                                    <name>resolutionY</name>
                              </symbol>
                              <symbol xsi:type="selectiveImageSymbol"
                                    identifier="IDI16_I01_S03">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <valueInterpretation>
                                          <valueLabel>unknown</valueLabel>
                                          <keyValue>0</keyValue>
                                          <valueLabel>metre</valueLabel>
                                          <keyValue>1</keyValue>
                                    </valueInterpretation>
                                    <name>resolutionUnit</name>
                              </symbol>

                        </item><!-- end of Chunk Data -->
                        <symbol xsi:type="basicSymbol"
identifier="IDI16_S03"
                              originalName="crc">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>

                              <name>checksum</name>
                        </symbol>
                  </item> <!-- end of  pHYS-Chunk -->

                  <!--start of sPLT-->
                  <item xsi:type="basicItem" identifier="IDI17"
originalName="sPLT"
                        optional="true" multiple="true">
                        <symbol xsi:type="fileStructureSymbol"
identifier="IDI17_S01"
                              originalName="chunkDataLength">
                              <range>
                                    <startposition
xsi:type="expected">33</startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <name>dataBlockLength</name>
                        </symbol>
                        <property xsi:type="fileStructureProperty"
identifier="IDI17_S02"
                              originalName="ChunkType">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>
                              <value>115 80 76 84</value>
                              <name>dataBlockType</name>
                        </property>

                        <item xsi:type="basicItem" identifier="IDI17_I01">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length xsi:type="referencedLength"
ref="IDI17_S01">
                                          </length>
                              </range>

                              <symbol xsi:type="selectiveImageSymbol"
                                    identifier="IDI17_I01_S01"
interpretation="Latin1"
                                    multiple="true">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <validValues xsi:type="valueRange">
```

```
                                          <startRange>1</startRange>
                                          <endRange>255</endRange>
                                    </validValues>
                                    <name>suggestedPaletteName</name>
                              </symbol>

                              <property xsi:type="basicProperty"
identifier="IDI17_I01_S02">
                                    <range>
                                          <startposition
xsi:type="sequential">
                                                33</startposition>
                                          <length
xsi:type="fixedLength">1</length>
                                    </range>
                                    <value>0</value>
                                    <name>delimiter</name>
                              </property>
                              <name>PaletteName</name>


                              <item xsi:type="basicItem" ordered="false"
                                    identifier="IDI17_I01_C01">

                                    <item xsi:type="basicItem"
identifier="jan1"

                                          name="8BitDepth">

                                          <property
xsi:type="selectiveImageSymbol"
                                                identifier="jan3">
                                                <range>
                                                      <startposition
xsi:type="sequential">
                                                            </startposition>
                                                      <length
xsi:type="fixedLength">1</length>
                                                </range>
                                                <value>8</value>

      <name>suggestedPaletteSampleDepth</name>
                                           </property>

                                          <item xsi:type="imageItem"

      identifier="IDI17_I01_C01_I01" optional="true"

      originalName="suggestedPaletteRGB8">

                                                <symbol
xsi:type="basicImageSymbol"

      identifier="IDI17_I01_C01_I01_S01">
                                                      <range>
                                                            <startposition
xsi:type="sequential">

      </startposition>
                                                            <length
xsi:type="fixedLength">1</length>
                                                      </range>
      <name>truecolourRed</name>
                                               </symbol>
                                               <symbol>
xsi:type="basicImageSymbol"

      identifier="IDI17_I01_C01_I01_S02">
                                                     <range>
                                                           <startposition
xsi:type="sequential">

      </startposition>
                                                           <length
xsi:type="fixedLength">1</length>
                                                     </range>

      <name>truecolourGreen</name>
                                               </symbol>
                                               <symbol
xsi:type="basicImageSymbol"

      identifier="IDI17_I01_C01_I01_S03">
                                                     <range>
                                                           <startposition
xsi:type="sequential">

      </startposition>
                                                           <length
xsi:type="fixedLength">1</length>
                                                     </range>

      <name>truecolourBlue</name>
                                               </symbol>
                                               <symbol
xsi:type="basicImageSymbol"
                                                     identifier="jan6">
                                                     <range>
                                                           <startposition
xsi:type="sequential">

      </startposition>
                                                           <length
xsi:type="fixedLength">1</length>
                                                     </range>
                                                     <name>alpha</name>
                                               </symbol>

      <name>suggestedPaletteRGB</name>
                                         </item>
                                         <!-- end of item suggested Palette
RGB for 8 bit Depth -->
                                         <name>sPLTERGB8</name>
                                   </item><!--8bitDepth-->

                                   <item xsi:type="basicItem"
identifier="jan2"
                                         name="16BitDepth">
                                         <property
xsi:type="selectiveImageSymbol"
                                               identifier="jan4">
                                               <range>
```

```
                                                      <startposition
xsi:type="sequential">

                                                            </startposition>
                                                      <length
xsi:type="fixedLength">1</length>
                                                </range>
                                                <value>16</value>

      <name>suggestedPaletteSampleDepth</name>
                                          </property>
                                          <item xsi:type="imageItem"
optional="true"

      identifier="IDI17_I01_C01_I02">

                                                <symbol
xsi:type="basicImageSymbol"

      identifier="IDI17_I01_C01_I02_S01">
                                                      <range>
                                                            <startposition
xsi:type="sequential">

      </startposition>
                                                            <length
xsi:type="fixedLength">2</length>
                                                      </range>

      <name>truecolourRed</name>
                                                </symbol>
                                                <symbol
xsi:type="basicImageSymbol"

      identifier="IDI17_I01_C01_I02_S02">
                                                      <range>
                                                            <startposition
xsi:type="sequential">

      </startposition>
                                                            <length
xsi:type="fixedLength">2</length>
                                                      </range>

      <name>truecolourGreen</name>
                                                </symbol>
                                                <symbol
xsi:type="basicImageSymbol"

      identifier="IDI17_I01_C01_I02_S03">
                                                      <range>
                                                            <startposition
xsi:type="sequential">

      </startposition>
                                                            <length
xsi:type="fixedLength">2</length>
                                                      </range>

      <name>truecolourBlue</name>
                                                </symbol>
                                                <symbol
xsi:type="basicImageSymbol"
```

```
                                                      identifier="jan5">
                                                      <range>
                                                            <startposition
xsi:type="sequential">

      </startposition>
                                                            <length
xsi:type="fixedLength">2</length>
                                                      </range>
                                                      <name>alpha</name>
                                                </symbol>

      <name>suggestedPaletteRGB</name>
                                          </item>
                                          <!-- end of item suggested Palette
RGB for 16 bit Depth -->
                                    </item> <!--16BitDepth-->
                              </item>
                              <!-- end of unordered Container for suggested
Palette RGB -->

                              <symbol xsi:type="basicImageSymbol"
identifier="IDI17_I01_S04"
                                    originalName="frequency"
multiple="true">
                                    <range>
                                          <startposition
xsi:type="sequential"></startposition>
                                          <length
xsi:type="fixedLength">2</length>
                                    </range>
                                    <name>suggestedPaletteFrequency</name>
                              </symbol>
                        <!-- end of Chunk Data -->
                        </item>
                        <symbol xsi:type="basicSymbol"
identifier="IDI17_S03"
                              originalName="crc">
                              <range>
                                    <startposition
xsi:type="sequential"></startposition>
                                    <length
xsi:type="fixedLength">4</length>
                              </range>

                              <name>checksum</name>
                        </symbol>
                  </item> <!-- end of  sPLT-Chunk -->

            </item>
            <!-- end of unordered container -->
            <!--IEND start-->
            <item xsi:type="basicItem" identifier="ID01_I18"
originalName="IEND"
                  multiple="false">
                  <property xsi:type="basicProperty"
identifier="ID01_I18_S01"
                        originalName="chunkDataLength">
                        <range>
                              <startposition
xsi:type="expected"></startposition>
                              <length xsi:type="fixedLength">4</length>
                        </range>
                        <value>0 0 0 0</value>
                        <name>dataBlockLength</name>
                  </property>
                  <property xsi:type="basicProperty"
identifier="ID01_I18_S02"
                        originalName="ChunkType">
                        <range>
                              <startposition
xsi:type="sequential"></startposition>
                              <length xsi:type="fixedLength">4</length>
                        </range>
                        <value>73 69 78 68</value>
                        <name>dataBlockType</name>
                  </property>
                  <symbol xsi:type="basicSymbol" identifier="ID01_I18_S04"
                        originalName="crc">
                        <range>
                              <startposition
xsi:type="sequential"></startposition>
                              <length xsi:type="fixedLength">4</length>
                        </range>
                        <name>checksum</name>
                  </symbol>
            </item>
            <!-- end of IEND-Chunk -->
      </item>
      <!-- end of ordered container -->
</XCELDocument>
```

Example XCDL: Instance of a PNG file

```
<?xml version="1.0" encoding="UTF-16"?>
<!-- this is an instance of the XCDL for a PNG formatted image. Note that this example includes an
exhaustive and redundant use of XCDL elements for demonstration purposes. In practice, according
to the specific usage,  this may not be reasonable. Also note that this example is not including all of
the functionality of the XCDL. This also depends on the intended specific usage and the type of
content of the digital object. -->
<xcdl xmlns="http://www.planets-project.eu/xcl/schemas"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.planets-project.eu/xcl/schemas            XCDLCore.xsd"
        id="A123456">
        <object id="0" natFormat="png11">
                <data
id="0">670000000774494D4507D60A05170000000000000000000774494D4507D60A051702
00000000000000000467414D41000186A031E8965F00000003734249540101017C2E7782000
00001735247420100000000000000206348524D00007A26000080840000FA00000080E80000
75300000EA6000003A9800001770000000000000000C504C544500FF00FF0000FFFF000000F
F653F2BBA00000001624B474401000000000000000868495354000500050005000000000000
0000000974524E5390909090909090909000000000000000097048597300000078000000780
0000000000000002173504C5450414C4554544549444154001003E803E803E8006400C807D0
07D007D0006400C8000000000000002249444154789C63F81FBA8A018419F030F0C9C118ABA
02C7C0CA8D2FFF81883CC3D00BD497F81EA898E1B0000001A69545874695458743100303064
650069545874310048414C4C4F5F4A414E000000000000000049454E44AE426082
        </data>
        <normData id="1"
type="image">789C63F81FBA8A018419F030F0C9C118ABA02C7C0CA8D2FFF81883CC3D00BD
497F81
        </normData>
                <property id="1" source="raw" cat="descr">
                        <name>width</name>
                        <valueSet id="1">
                                <rawVal>00000020</rawVal>
                                <labVal>
                                        <val unit="pixel">32</val>
                                        <type>int</type>
                                </labVal>
                                <dataRef ind="normAll">
                                        <ref id="1" start="0" end="33"/>
                                </dataRef>
                        </valueSet>
                </property>
                <property id="2" source="raw" cat="descr">
                        <name>height</name>
                        <valueSet id="1">
                                <rawVal>00000020</rawVal>
                                <labVal>
                                        <val unit="pixel">32</val>
                                        <type>int</type>
                                </labVal>
                                <dataRef ind="normAll"/>
                        </valueSet>
                </property>
                <property id="3" source="raw" cat="descr">
                        <name alias="bitDepth">colourDepth</name>

                        <valueSet id="1">
                                <rawVal >02</rawVal>
                                <labVal>
                                        <val unit="bit" base="palette">2</val>
```

```
                  <type>int</type>
            </labVal>
            <dataRef ind="normAll"/>
      </valueSet>
</property>
<property id="4" source="raw" cat="descr">
      <name alias="colourType">imageType</name>
      <valueSet id="1">
            <rawVal >03</rawVal>
            <labVal>
                  <val>palette</val>
                  <type>fixLabel</type>
            </labVal>
            <dataRef ind="normAll"/>
      </valueSet>
</property>
<property id="5" source="raw" cat="descr">
      <name>compression</name>
      <valueSet id="1">
            <rawVal >00</rawVal>
            <labVal>
                  <val>zlibDeflateInflate</val>
                  <type>fixLabel</type>
            </labVal>
            <dataRef ind="normAll"/>
      </valueSet>
</property>
<property id="6" source="raw" cat="descr">
      <name>filter</name>
      <valueSet id="1">
            <rawVal >00</rawVal>
            <labVal>
                  <val>adaptive</val>
                  <type>fixLabel</type>
            </labVal>
            <dataRef ind="normAll"/>
      </valueSet>
</property>
<property id="7" source="raw" cat="descr">
      <name>interlace</name>
      <valueSet id="1">
            <rawVal >00</rawVal>
            <labVal>
                  <val>noInterlace</val>
                  <type>fixLabel</type>
            </labVal>
            <dataRef ind="normAll"/>
      </valueSet>
</property>
<property id="8" source="raw" cat="descr">
      <name>gamma</name>
      <valueSet id="1">
            <rawVal>0000B18F</rawVal>
            <labVal>
                  <val>0.45455</val>
                  <type>decimal</type>
            </labVal>
            <dataRef ind="normAll"/>
      </valueSet>
</property>
<property id="9" source="raw" cat="descr">
      <name>significantBits</name>
```

```
                  <valueSet id="1">
                        <rawVal >010101</rawVal>
                        <labVal>
                              <val >1 1 1</val>
                              <type group="3">int</type>
                        </labVal>
                        <dataRef ind="normAll"/>
                  </valueSet>
            </property>
            <property id="10" source="raw" cat="descr">
                  <name>rgbPalette</name>
                  <valueSet id="1">
                        <rawVal>00FF00FF0000FFFF000000FF</rawVal>
                        <labVal>
                              <val>0 255 0 255 0 0 255 255 0 0 0 255</val>
                              <type group="3">int</type>
                        </labVal>
                        <dataRef ind="normAll"/>
                  </valueSet>
            </property>
            <property id="11" source="raw" cat="descr">
                  <name>transparency</name>
                  <valueSet id="1">
                        <rawVal>90 90 90 90 90 90 90 90 90</rawVal>
                        <labVal>
                              <val>144 144 144 144 144 144 144 144 144</val>
                              <type group="3">int</type>
                        </labVal>
                        <dataRef ind="normAll"/>
                  </valueSet>
            </property>
            <property id="12" source="raw" cat="descr">
                  <name>timeLastMod</name>

                  <valueSet id="1">
                        <rawVal>07D60A05170000</rawVal>
                        <labVal>
                              <val>2006-09-05T23:00:00</val>
                              <type>dateTime</type>
                        </labVal>
                        <dataRef ind="normAll"/>
                  </valueSet>
                  <valueSet id="2">
                        <rawVal>07D60A05170200</rawVal>
                        <labVal>
                              <val>2006-09-05T23:02:00</val>
                              <type>dateTime</type>
                        </labVal>
                        <dataRef ind="normAll"/>
                  </valueSet>
            </property>
            <property id="13" source="raw" cat="descr">
                  <name>1931IEC_ChromaticityRedX</name>

                  <valueSet id="1">
                        <rawVal>0000FA00</rawVal>
                        <labVal>
                              <val>0.64</val>
                              <type>decimal</type>
                        </labVal>
                        <dataRef ind="normAll"/>
                  </valueSet>
```

```
        </property>
        <property id="14" source="raw" cat="descr">
            <name>1931IEC_ChromaticityRedY</name>
                <valueSet id="1">
                <rawVal>000080E8</rawVal>
                <labVal>
                    <val>0.33</val>
                    <type>decimal</type>
                </labVal>
                <dataRef ind="normAll"/>
            </valueSet>
        </property>
        <property id="15" source="raw" cat="descr">
            <name>1931IEC_ChromaticityGreenX</name>
            <valueSet id="1">
                <rawVal>00007530</rawVal>
                <labVal>
                    <val>0.3</val>
                    <type>decimal</type>
                </labVal>
                <dataRef ind="normAll"/>
            </valueSet>
        </property>
        <property id="16" source="raw" cat="descr">
            <name>1931IEC_ChromaticityGreenY</name>
            <valueSet id="1">
                <rawVal>0000EA60</rawVal>
                <labVal>
                    <val>0.6</val>
                    <type>decimal</type>
                </labVal>
                <dataRef ind="normAll"/>
            </valueSet>
        </property>
        <property id="17" source="raw" cat="descr">
            <name>1931IEC_ChromaticityBlueX</name>
            <valueSet id="1">
                <rawVal>00003A98</rawVal>
                <labVal>
                    <val>0.15</val>
                    <type>decimal</type>
                </labVal>
                <dataRef ind="normAll"/>
            </valueSet>
        </property>
        <property id="18" source="raw" cat="descr">
            <name>1931IEC_ChromaticityBlueY</name>
            <valueSet id="1">
                <rawVal>00001770</rawVal>
                <labVal>
                    <val>0.06</val>
                    <type>decimal</type>
                </labVal>
                <dataRef ind="normAll"/>
            </valueSet>
        </property>
        <property id="19" source="raw" cat="descr">
            <name>whitePointX</name>
            <valueSet id="1">
                <rawVal>00007A26</rawVal>
                <labVal>
                    <val>0.3127</val>
```

```
                  <type>decimal</type>
            </labVal>
            <dataRef ind="normAll"/>
      </valueSet>
</property>
<property id="20" source="raw" cat="descr">
      <name>whitePointY</name>
      <valueSet id="1">
            <rawVal>00008084</rawVal>
            <labVal>
                  <val>0.329</val>
                  <type>decimal</type>
            </labVal>
            <dataRef ind="normAll"/>
      </valueSet>
</property>
<property id="21" source="raw" cat="descr">
      <name>ICC-1/ICC-1A_RenderingIntent</name>
      <valueSet id="1">
            <rawVal>01</rawVal>
            <labVal>
                  <val>relativeColorimetric</val>
                  <type>fixLabel</type>
            </labVal>
            <dataRef ind="normAll"/>
      </valueSet>
</property>
<property id="22" source="raw" cat="descr">
      <name>backgroundColour</name>
      <valueSet id="1">
            <rawVal>01</rawVal>
            <labVal>
                  <val>0 255 0</val>
                  <type group="3">int</type>
            </labVal>
            <dataRef ind="normAll"/>
      </valueSet>
</property>
<property id="23" source="raw" cat="descr">
      <name>histogram</name>
            <valueSet id="1">
            <rawVal>0005000500050005</rawVal>
            <labVal>
                  <val>5 5 5 5</val>
                  <type>int</type>
            </labVal>
            <dataRef ind="normAll"/>
      </valueSet>
</property>
<property id="24" source="raw" cat="descr">
      <name>resolutionX</name>
      <valueSet id="1">
            <rawVal>00000078</rawVal>
            <labVal>
                  <val unit="pixel">120</val>
                  <type>int</type>
            </labVal>
            <dataRef ind="normAll"/>
      </valueSet>
</property>
<property id="25" source="raw" cat="descr">
      <name>resolutionY</name>
```

```xml
            <valueSet id="1">
                <rawVal>00000078</rawVal>
                <labVal>
                    <val unit="pixel">120</val>
                    <type>int</type>
                </labVal>
                <dataRef ind="normAll"/>
            </valueSet>
        </property>
        <property id="26" source="raw" cat="descr">
            <name>suggestedPaletteName</name>
            <valueSet id="1">
                <rawVal>50414C4554544549444154</rawVal>
                <labVal>
                    <val>PALETTEIDAT</val>
                    <type>string</type>
                </labVal>
                <dataRef ind="normAll"/>
                <propRel id="1">
                    <rel propId="27" valSetId="1"/>
                    <rel propId="28" valSetId="1"/>
                    <rel propId="29" valSetId="1"/>
                    <rel propId="30" valSetId="1"/>
                </propRel>
                <propRel id="2">
                    <rel propId="27" valSetId="1"/>
                    <rel propId="28" valSetId="2"/>
                    <rel propId="29" valSetId="1"/>
                    <rel propId="30" valSetId="1"/>
                </propRel>
            </valueSet>
        </property>
        <property id="27" source="raw" cat="descr">
            <name>suggestedPaletteSampleDepth</name> <!-- =sPLT -->
            <valueSet id="1">
                <rawVal>0010</rawVal>
                <labVal>
                    <val>16</val>
                    <type>int</type>
                </labVal>
                <dataRef ind="normAll"/>
                <propRel id="1">
                    <rel propId="26" valSetId="1"/>
                    <rel propId="28" valSetId="1"/>
                    <rel propId="29" valSetId="1"/>
                    <rel propId="30" valSetId="1"/>
                </propRel>
                <propRel id="2">
                    <rel propId="26" valSetId="1"/>
                    <rel propId="28" valSetId="2"/>
                    <rel propId="29" valSetId="1"/>
                    <rel propId="30" valSetId="1"/>
                </propRel>
            </valueSet>
        </property>
        <property id="28" source="raw" cat="descr">
            <name>suggestedPaletteRGB</name>
            <valueSet id="1">
                <rawVal>03E803E803E8</rawVal>
                <labVal>
                    <val>1000 1000 1000</val>
                    <type group="3">int</type>
                </labVal>
                <dataRef ind="normAll"/>
                <propRel id="1">
                    <rel propId="26" valSetId="1"/>
                    <rel propId="27" valSetId="1"/>
                    <rel propId="29" valSetId="1"/>
                    <rel propId="30" valSetId="1"/>
                </propRel>
            </valueSet>
            <valueSet id="2">
                <rawVal>07D007D007D0</rawVal>
                <labVal>
                    <val>2000 2000 2000</val>
                    <type group="3">int</type>
                </labVal>
                <dataRef ind="normAll"/>
                <propRel id="1">
                    <rel propId="26" valSetId="1"/>
                    <rel propId="27" valSetId="1"/>
                    <rel propId="29" valSetId="2"/>
                    <rel propId="30" valSetId="2"/>
                </propRel>
            </valueSet>
        </property>
        <property id="29" source="raw" cat="descr">
            <name>suggestedPaletteAlpha</name>
            <valueSet id="1">
                <rawVal>0064</rawVal>
                <labVal>
                    <val>100</val>
                    <type>int</type>
                </labVal>
                <dataRef ind="normAll"/>
                <propRel id="1">
                    <rel propId="26" valSetId="1"/>
                    <rel propId="27" valSetId="1"/>
                    <rel propId="28" valSetId="1"/>
                    <rel propId="30" valSetId="1"/>
                </propRel>
                <propRel id="2">
                    <rel propId="26" valSetId="1"/>
                    <rel propId="27" valSetId="1"/>
                    <rel propId="28" valSetId="2"/>
                    <rel propId="30" valSetId="1"/>
                </propRel>
            </valueSet>
        </property>
        <property id="30" source="raw" cat="descr">
            <name>suggestedPaletteFrequency</name>
            <valueSet id="1">
                <rawVal>00C8</rawVal>
                <labVal>
                    <val>200</val>
                    <type group="3">int</type>
                </labVal>
                <dataRef ind="normAll"/>
                <propRel id="1">
                    <rel propId="26" valSetId="1"/>
                    <rel propId="27" valSetId="1"/>
                    <rel propId="28" valSetId="1"/>
                    <rel propId="29" valSetId="1"/>
                </propRel>
```

```
                        <propRel id="2">
                              <rel propId="26" valSetId="1"/>
                              <rel propId="27" valSetId="1"/>
                              <rel propId="28" valSetId="2"/>
                              <rel propId="29" valSetId="1"/>
                        </propRel>
                  </valueSet>
            </property>
            <property id="31" source="raw" cat="descr">
                  <name>textualDataKeyword</name>
                  <valueSet id="1">
                        <rawVal>6954587431</rawVal>
                        <labVal>
                              <val>iTXT1</val>
                              <type>string</type>
                        </labVal>
                        <dataRef ind="normAll"/>
                  </valueSet>
            </property>
            <property id="32" source="raw" cat="descr">
                  <name>textualDataString</name>
                  <valueSet id="1">
                        <rawVal>48414C4C4F5F4A414E</rawVal>
                        <labVal>
                              <val>HALLO_JAN</val>
                              <type>string</type>
                        </labVal>
                        <dataRef ind="normAll"/>
                  </valueSet>
            </property>
            <property id="32" source="raw" cat="descr">
                  <name>textualDataLanguage</name>
                  <valueSet id="1">
                        <rawVal>6465</rawVal>
                        <labVal>
                              <val>de</val>
                              <type>string</type>
                        </labVal>
                        <dataRef ind="normAll"/>
                  </valueSet>
            </property>
            <property id="33" source="implicit" cat="descr">
                  <name>blackWhiteInterpretation</name>
                  <valueSet id="1">
                        <labVal>
                              <val>blackIsZero</val>
                              <type>fixLabel</type>
                        </labVal>
                        <dataRef ind="normAll"/>
                  </valueSet>
            </property>
      </object>
</xcdl>
```

References

[ESRI98]

ESRI shape file technical description (1998).

http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf

[MS05]

Microsoft Office Open XML Formats Overview (2005).

http://www.microsoft.com/office/preview/itpro/fileoverview.mspx

[OAIS02]

Reference Model for an Open Archival Information System (OAIS) (2002).

http://public.ccsds.org/publications/archive/650x0b1.pdf

[PNG03]

PNG specification (second edition) (2003).

http://www.w3.org/TR/PNG/

[PRO06]

The technical registry PRONOM (2006).

http://www.nationalarchives.gov.uk/pronom/

[RTF01]

Rich text format version 1.7 (2001).

http://support.microsoft.com/kb/269575/de?spid=3252&sid=global