Welcome to the XCL project

Project Summary

Today file format definitions are formulated in natural languages. A programmer who wants to decode, encode or render the information has to read through the specification document before translating it into a programming language. Usually the authors of a format definition provide at least a library for the developed format. While this is a well proven process the translation from one format into another format is often an error-prone undertaking. For content holders format conversion is a basic instrument to assure long term access to their digital resources. However today there is no standardised automatic procedure for the evaluation of format conversions available. The **XCL** technology is a highly abstracted and formalised proposal to solve this problem. The idea is to develop a formal **eXtensible Characterisation Extraction Language (XCEL)** that can be used by file format designers or content holders to describe the structure of binary files. With such a description a machine will be able to extract properties from the binary object and to translate it into a generic comparable representation in an **eXtensible Characterisation Definition Language (XCDL)**

XCDL summary

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.

XCEL summary

Together with the Extensible Characterisation Definition Language (XCDL) the Extensible Characterisation Extraction Language (XCEL) builds the Extensible Characterisation Language (XCL). The objective of the XCEL is to describe structures and meanings of digital objects in a machinereadable way. The underlying technology of the XCEL is XML and XML Schema. The idea is to provide an extensible set of XML Schemas defining an XML dialect which enables a file format expert to transform any specified format into a machinereadable XCEL description. In the future it should also be possible to describe composite objects with the XCEL, i.e. objects that include more than one file (e.g. OOXML format). As an XML format the XCEL describes structures in a treelike form.

Welcome to the XCL project

Project Summary

Today file format definitions are formulated in natural languages. A programmer who wants to decode, encode or render the information has to read through the specification document before translating it into a programming language. Usually the authors of a format definition provide at least a library for the developed format. While this is a well proven process the translation from one format into another format is often an error-prone undertaking. For content holders format conversion is a basic instrument to assure long term access to their digital resources. However today there is no standardised automatic procedure for the evaluation of format conversions available. The **XCL** technology is a highly abstracted and formalised proposal to solve this problem. The idea is to develop a formal **eXtensible Characterisation Extraction Language (XCEL)** that can be used by file format designers or content holders to describe the structure of binary files. With such a description a machine will be able to extract properties from the binary object and to translate it into a generic comparable representation in an **eXtensible Characterisation Definition Language (XCDL)**

XCDL summary

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.

XCEL summary

Together with the Extensible Characterisation Definition Language (XCDL) the Extensible Characterisation Extraction Language (XCEL) builds the Extensible Characterisation Language (XCL). The objective of the XCEL is to describe structures and meanings of digital objects in a machinereadable way. The underlying technology of the XCEL is XML and XML Schema. The idea is to provide an extensible set of XML Schemas defining an XML dialect which enables a file format expert to transform any specified format into a machinereadable XCEL description. In the future it should also be possible to describe composite objects with the XCEL, i.e. objects that include more than one file (e.g. OOXML format). As an XML format the XCEL describes structures in a treelike form.

Welcome to the XCL project

Project Summary

Today file format definitions are formulated in natural languages. A programmer who wants to decode, encode or render the information has to read through the specification document before translating it into a programming language. Usually the authors of a format definition provide at least a library for the developed format. While this is a well proven process the translation from one format into another format is often an error-prone undertaking. For content holders format conversion is a basic instrument to assure long term access to their digital resources. However today there is no standardised automatic procedure for the evaluation of format conversions available. The **XCL** technology is a highly abstracted and formalised proposal to solve this problem. The idea is to develop a formal **eXtensible Characterisation Extraction Language (XCEL)** that can be used by file format designers or content holders to describe the structure of binary files. With such a description a machine will be able to extract properties from the binary object and to translate it into a generic comparable representation in an **eXtensible Characterisation Definition Language (XCDL)**

XCDL summary

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.

XCEL summary

Together with the Extensible Characterisation Definition Language (XCDL) the Extensible Characterisation Extraction Language (XCEL) builds the Extensible Characterisation Language (XCL). The objective of the XCEL is to describe structures and meanings of digital objects in a machinereadable way. The underlying technology of the XCEL is XML and XML Schema. The idea is to provide an extensible set of XML Schemas defining an XML dialect which enables a file format expert to transform any specified format into a machinereadable XCEL description. In the future it should also be possible to describe composite objects with the XCEL, i.e. objects that include more than one file (e.g. OOXML format). As an XML format the XCEL describes structures in a treelike form.

Welcome to the XCL project

Project Summary

Today file format definitions are formulated in natural languages. A programmer who wants to decode, encode or render the information has to read through the specification document before translating it into a programming language. Usually the authors of a format definition provide at least a library for the developed format. While this is a well proven process the translation from one format into another format is often an error-prone undertaking. For content holders format conversion is a basic instrument to assure long term access to their digital resources. However today there is no standardised automatic procedure for the evaluation of format conversions available. The **XCL** technology is a highly abstracted and formalised proposal to solve this problem. The idea is to develop a formal **eXtensible Characterisation Extraction Language (XCEL)** that can be used by file format designers or content holders to describe the structure of binary files. With such a description a machine will be able to extract properties from the binary object and to translate it into a generic comparable representation in an **eXtensible Characterisation Definition Language (XCDL)**

XCDL summary

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.

XCEL summary

Together with the Extensible Characterisation Definition Language (XCDL) the Extensible Characterisation Extraction Language (XCEL) builds the Extensible Characterisation Language (XCL). The objective of the XCEL is to describe structures and meanings of digital objects in a machinereadable way. The underlying technology of the XCEL is XML and XML Schema. The idea is to provide an extensible set of XML Schemas defining an XML dialect which enables a file format expert to transform any specified format into a machinereadable XCEL description. In the future it should also be possible to describe composite objects with the XCEL, i.e. objects that include more than one file (e.g. OOXML format). As an XML format the XCEL describes structures in a treelike form.

Long-time preservation

A solution by XCL

by Volker Heydegger, Susanne Kurz, Jan Schnasse, Manfred Thaller

In the last decades a radical change away from analogue up to digital creation, exchanging and discovering of information-resources at very different fields like science, culture, management, commercial and so on is observable. This movement with upward trend provides many new possibilities but it creates in different ways a lot of new problems. One problem is long-term preservation of digital produced resources as well as retrospectively digitized material because this material is in different ways inherently ephemeral.

To provide the digital heritage also in the future it is inevitable to consider two very different areas: the physical and the logical sound condition of data. So on one hand the intactness of each data storage medium must be ensured and on the other hand the guarantee of the correct interpretation of data has to be offered. (Digital data at first are not more than sequences of 0 and 1.) At present the hardware problem is solved by regular interchange and complex backup strategies but obsolescence of software systems that solely ensure the original interpretation of data is an unsolved problem yet.

As a basic principle there are two ways of problem solving: emulation and migration. While people attending the emulation strategy aim at doing a reproduction of obsolete software systems (including operating system, possibly computer architecture and the respective application) inside of actual operating systems, migration means converting at regular intervals and timely (means at a point of time the technological obsolescence is not yet progressed and correct interpretation of data is still possible) obsolete file formats into stable and current formats. One of the following problems by using this strategy is to ensure the all information of the old format is contained in the new one. Automatic detection of loss of information caused by converting file formats isn’t a trivial task especially if a huge number of heterogeneous data is concerned (in such a way at archives and libraries).

Many different kinds of data can be found within those data collections: text and image files (with the different between raster and vector images), audio and video files, virtual reality and 3d graphic files, presentations, spreadsheets, databases and so on. Inside every category many different file formats exist:

- Text formats: pdf, rtf, plain text, MS Word, OpenOffice, WordPerfect, HTML...
- Raster graphics: tiff, png, gif, jpg, psd, ...
- Vektor graphics: svg, eps, cdr, dxf, swf, ...
- Audio formats: aiff, wave, midi, mpeg, wma, ram, snd, ogg, au, ...
- Video formats: avi, mpeg, Quicktime, mov, ogg, wmv, rv, ...
- 3D- graphics /VR: X3D, VRML, U3D, ...

- Presentations: MS PowerPoint, OpenOffice, Keynote, ...
- Spreadsheets: MS Excel, OpenOffice, ...
- Databases: SQL DDL, MS Access, Filemaker, ...

Besides the diversity of formats an additional format intrinsic problem arises. In many cases the file format specification is not a stable and durable one; it underlies development and integration of new functionality so that several versions of a file format specification enhance the diversity in addition. Particularly archives and libraries collecting digital data for a long time are concerned of this.

Another type of complexity is based on the extents of coverage of text information. On principle every text document can be divided into a layout and a content component. Some experts argue that archiving and preserving only content as the essential part of text documents is effectual enough. Apart from the matter of fact that layout can be the a research topic particularly with regard to historical documents, such an approach is leading to problems. For instance: archiving only the content of a table avoids any capturing of comprehension if the order of columns and rows is not longer present. In many cases text formatting, e.g. alternative, bold or italic fonts or indentations, listings or paragraphs, leads to a correct understanding of content. These examples show that a general decision of preserving content is not equivalent with archiving solely content without layout. Even though layout is not always relevant it must be taken into consideration.[1]

The migration of obsolete file formats into current ones is a comparative well-supported procedure of software vendors and with the utmost probability it must be rerun by the concerned institutions. Especially migration of a huge number of heterogeneous data needs elaborate care and control of loss of information caused by the act of migration itself. But at this point there is a lack of automated solutions to identify problems and to separate them from unproblematic cases.

The Cologne method of resolving this set of problems is based on the development of a XML[2]formal language describing structures and content of files. XML is long time field-tested and provides the necessary independence of applications, operating systems and hardware. XML data are human and machine readable, comparatively simple for understanding and presentable without converting through each text editor at arbitrary operation system. Very important is the fact that XML is a general rule type for the development of special languages for special problems. These languages can be processed by every XML compatible application. XML based languages and mark-up schema avail these properties for effective storage, illustration and editing by standard tools. A further benefit of XML is the possibility of document enhancement without encounter the previous content. This is a very important characteristic, specifically in the area of file description, because the number of file formats is more than unclear by now.

The XML based Extensible Characterisation Language (XCL), composed of XCEL[3] and XCDL[4], backs on the idea to capture the data stream of a file and to extract the content as it is constituted by its file format specification, which is therefore transformed into an XCEL document: It is assumed that all data can be consistently described by the abstract language XCEL. On the one hand the XCEL document shapes the structure of a file format and on the other hand it completes the elements of the structure with content. By using a special software (the so-called extractor, see

figure below) a file can be transformed into a general descriptive form. The XCDL provides the formal description for this transformation.

Applying the XCL on digital data originating from different file formats enables to compare content in an automated way, based on a unique representation as XCDL.

Every XCEL document covers only one file format specification, so for every file format specification an equivalent XCEL document is necessary. Such a transformation of semantic description to a structured, machine-readable XML Code is only feasible if the file format specification is completely documented.

At the first stage of the Cologne project a concentration on the potential of an automated and high-performance comparison to verify the loss of information by format converting of a great many of data files is made. This is particularly a problem for one major preservation strategies: file conversion.

Terms and Definitions: Glossar

Item

> An Item is an entity that groups other entities in a structural, logical or semantical way. An Item has always children; in this sense it is non-terminal. The starting Item of any XCEL Instance is called Root Item.

Symbol

> A Symbol is a terminal entity that describes the position, length, encoding and semantics of a byte sequence.

Property

> A Property is a special form of a Symbol in the way, that its value is determined by the file format specification.

Processing

> A Processing encapsulates runtime dependencies that affect the XCEL tree structure or the byte stream depending on the occurrence of single values.

XCEL Tree

> An XCEL Tree is made up of Items, Symbols, Properties and Processings. The root of an XCEL Tree must be an Item. Leaves of an XCEL Tree can either be Symbols, Properties or Processings.

XCEL Element

XCEL Element (short: 'Element') is the shared term for Items, Symbols, Properties and Processings.

XCEL Processor

A tool that is able to process an XCEL instance and any corresponding binary file. The XCEL Processor developed by Planets is called 'Extractor'.

XCEL Schemas

Entirety of the single XML schemas which express parts of the XCEL.

Digital object

A digital object is an abstract term for a beast that lives on a computer and you are thinking of as an entity. The way how a digital object is represented on a digital system is not defined by the digital object as a digital object. e.g. a digital object can be represented as a record, a file, two files, three files etc.

Fileformat

A Fileformat 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.

File

A file or more exact a regular file is the information handled by a special part of the Operating System called Filesystem. A file typically consists of two logicaly distinct blocks of data. The first datablock is handled by the Filesystem and typically is used to store administrative attributes like acces permissions etc. The second block is a sequence of bytes that can be described by a certain Fileformat. To describe the content of these second block is what the XCEL is made for.

XCEL Instance

An XCEL Instance is an XML file that is valid against the XCEL Schemas. Actually it describes a particular file format, e.g. a 'PNG XCEL instance' is an XCEL-conform representation of the PNG file format.

Content model

In this document the term content model is used for the different ways the content of an item can be structured. This structure is specified in the item's 'order' attribute. Valid values for it are 'all', 'sequence' and 'choice', each constituting a different content model. These three content models are described later in the text.