



Implementation of the MIAP Common Data Definitions

Phase 2

Paul Spencer

Document Information

Title	Implementation of the MIAP Common Data Definitions (Phase 2)
Version	1.0
Status	Release
Creator(s)	Paul Spencer
Contact	paul.spencer@boynings.co.uk
Subject	Education and Skills
Description	A description of the MIAP Common Data Definitions
Publisher	MIAP Programme
Contributor(s)	See list in document
Date(s)	Created: 2006-05-03
Format	Available as Microsoft Word and Adobe PDF
Identifier	d:\my documents\projects\129-hesa\101-miapcdd-2\reports\finalreport-phase2-v1-0.doc
Language	English
Project	Managing Information Across Partners
Client	HESA
Client Reference	129
Client Contact	Andy Youell (Andy.Youell@hesa.ac.uk)

Change History

0.1	First draft
0.2	Renamed from "tranche 2" to "phase 2" Various minor amendments following the project board of 26 April 2006
1.0	Amendments following final workshop: Brief introduction to Schematron added Section 8.6 on "imprecise dates" removed

CONTENTS

1	Introduction.....	6
2	Background to MIAP and the CDD.....	7
2.1	Background to the Common Data Definitions	7
2.2	Structure the Common Data Definitions.....	7
2.3	Adoption of the Common Data Definitions.....	7
3	Technical Strategy	9
3.1	Use of XML.....	9
3.2	Use of Unicode.....	9
3.3	Use of UTF-8.....	10
4	Character Sets and Languages.....	11
4.1	Implications of the Character Set Support	11
4.2	Roman-Based Character Issues	11
4.2.1	Displaying Data.....	12
4.2.2	Copying or Moving Data.....	12
4.2.3	Comparing Data	13
5	Full Technical Implementation - Background	14
5.1	The Common Data Definitions	14
5.1.1	Introduction	14
5.1.2	Formatting the CDD like the GDSC.....	14
6	Full Technical Implementation - XML Policy.....	15
6.1	Adherence to e-GIF Standards	15
6.1.1	Policy	15
6.1.2	Rationale.....	15
6.2	Use of Elements and Data Types.....	15
6.2.1	Policy	15
6.2.2	Rationale.....	15
6.3	Representation of "no data"	15
6.3.1	Policy	15
6.3.2	Rationale.....	16
6.4	Representing the Reason for Missing Data.....	16
6.4.1	Policy	16
6.4.2	Rationale.....	16
6.5	Representing Metadata.....	16
6.5.1	Policy	16
6.5.2	Rationale.....	16

6.6	Attributes	17
6.6.1	Policy	17
6.6.2	Rationale	17
6.7	Ids in schemas.....	17
6.7.1	Policy	17
6.7.2	Rationale	17
6.8	Elements with Multiple Patterns.....	18
6.8.1	Policy	18
6.8.2	Rationale	18
6.8.3	Example	18
6.9	Use of <code>xml:lang</code>	19
6.9.1	Policy	19
6.9.2	Rationale	19
6.9.3	Policy	19
6.9.4	Rationale	19
6.10	Use of Namespaces.....	19
6.10.1	Policy	19
6.10.2	Rationale	20
6.10.3	Policy	20
6.10.4	Rationale	20
7	Full Technical Implementation - UK Addresses	21
7.1	Alternative Formats.....	21
7.1.1	BS7666	21
7.1.2	UK GovTalk™ UKPostalAddressStructure	24
7.1.3	PAF® Address Format.....	25
7.1.4	UKRLP Proposed Address Format.....	25
7.1.5	UK GovTalk™ InternationalAddressStructure.....	26
7.2	Recommendation.....	26
8	Full Technical Implementation - XML Schemas	28
8.1	Changes to Element Names.....	30
8.2	Definitions Relating to Addresses.....	30
8.3	Definitions Relating to Telephone Numbers	30
8.4	PersonCountryOfDomicile and PersonNationality.....	30
8.5	"Data Items Which Cannot Currently Be Implemented"	30
8.6	Use of <code>xml:lang</code>	31
9	Contributors.....	32
10	References.....	33
	Appendix A. Character Sets and Languages - Background	35
	A.1 Legal Aspects	35

A.2	Other Public Sector Bodies.....	36
A.3	MIAP Participants.....	37
A.4	Required Unicode Characters for UK Languages.....	37
A.5	Options for Language and Character Set Support.....	38
A.6	Conclusions and Implications.....	39

1 Introduction

This report considers the implementation of the MIAP Common Data Definitions.

The report describes the background to the decisions made regarding the implementation. The implementation itself is defined by a set of XML schemas and associated rules that are documented in the CDD. The report is intended for developers of systems that will interface with MIAP participants, both to aid discussion during agreement of the schemas and to provide a record of the reasons for the decisions.

The results were achieved through a combination of interviews, email feedback and workshops. For tranche 1, structured interviews were carried out with many stakeholders and those elsewhere in Government who had experience with the issues involved (see section 9). Drafts of the Common Data Definitions, the schemas and documents were developed as a result of these. The schemas and documents were circulated to the stakeholders for review, comments incorporated and a workshop held to finalise the tranche 1 work. The schemas were quality assured externally to ensure adherence to best practice and e-GIF guidelines. During this process, the project board contributed through feedback on the documents and presentations. The work for tranche 2 was conducted in a similar way, and the results of the work on the two tranches combined into a single set of documents and schemas, referred to here as phase 2.

This report has the following sections:

Section 2 *Background to MIAP and the CDD* provides background information on the CDD and its governance.

Section 3 *Technical Strategy* describes the work undertaken to support the decisions to use XML and Unicode as the basis of the implementation and consideration of the use of UTF-8 as an encoding format for Unicode characters.

Section 4 *Character Sets and Languages* considers the legal aspects, common public sector practice and requirements for support of different languages and character sets.

Section 5 *Full Technical Implementation - Background* provides some background to the implementation. In particular, it discusses the additional work undertaken to represent the CDD in XML in the format used for the Government Data Standards Catalogue (GDSC) [17].

Section 6 *Full Technical Implementation - XML Policy* describes the policies used in addition to e-GIF policies when implementing the CDD as XML schema documents.

Section 7 *Full Technical Implementation - UK Addresses* describes additional work undertaken to consider the implications of different address formats that can be used in the CDD.

Section 8 *Full Technical Implementation - XML Schemas* provides additional information about the schemas themselves.

Migration issues are considered as they occur throughout the text.

2 Background to MIAP and the CDD

The MIAP Group was initially set up with a focus on Post-16 Learning and Skills. However, it quickly became apparent that there were important links with schools and higher education. On this basis MIAP's primary focus was shifted to include Post-16 and higher education, with close links to the DfES 14-19 Strategy and also the programme of work to develop a "New Relationship With Schools". About 40 organisations are members of the broadly-based MIAP Stakeholder Group. A small subset of these, including HESA, the Learning and Skills Council (LSC), and the Qualifications and Curriculum Authority (QCA), form a core group working closely with DfES to achieve delivery of the programme. There is strong participation from the relevant bodies in Wales and Northern Ireland, and increasingly close joint working with Scotland, and the vision for the programme is UK-wide.

The MIAP programme includes a number of specific strands. In addition to Common Data Definitions, the MIAP programme includes work on the UK Register of Learning Providers and the development of a data sharing framework. Future plans for MIAP cover the development of a Unique Learner Number and a Learning Data Interface, with close links to the QCA's proposed Framework for Achievement.

2.1 Background to the Common Data Definitions

The aim of the Common Data Definitions (CDD) strand is to develop a range of standard data definitions that can be used across a variety of data processes within the MIAP arena. The standardisation of key data definitions will facilitate the sharing of data and the construction of a single Learning Data Interface.

2.2 Structure the Common Data Definitions

The common data definitions have been created as a set of syntax-independent definitions and an implementation using XML Schema [1][2].

The syntax-independent definitions have been marked up using XML in the same way as the Government Data Standards Catalogue [17]. This format has been extended to include additional metadata required by the CDD. One benefit of using XML for this purpose is that the CDD can be rendered into different formats. In particular, it exists as an interactive HTML application and in a PDF format that can be easily printed.

The schema implementation has been developed in line with the e-GIF [3] and related Government Guidelines [18]. It comprises a set of common definitions, a set of schema documents implementing the terms in the CDD and some related documents for standard definitions (such as those within addresses) and code lists.

2.3 Adoption of the Common Data Definitions

The CDD should be used wherever data definitions are required within the scope of MIAP. Where possible, the schemas should also be used. If it is not possible to use either the definitions or the schemas, the MIAP governance authority should be notified so that the reasons can be examined and changes made to the CDD if

necessary. The CDD should also be considered as a source of information when any standards are being developed for learner and children's services.

3 Technical Strategy

3.1 Use of XML

When carrying out interviews, all interviewees were asked about their attitude to the use of XML for the CDD and whether they expected any migration problems. There were three types of response:

1. We are already using XML in other areas and would welcome it
2. We are not yet using XML, but have looked at the impact and are happy with the move
3. We are not yet using XML, have not looked at the impact, but are expecting this to happen and will accept it

Some further exploration was carried out with those in the last category. Although not exhaustive, it was apparent in every case they the organizations are using XML-compatible databases, and so the migration should not be hard.

The MIAP CDD will therefore be implemented as a set of XML schemas to the W3C XML Schema Recommendation [1][2].

3.2 Use of Unicode

ISO/IEC 10646-1 [1] defines a multi-octet character set (the Universal Character Set or UCS) which supports the character requirements of most of the world's written languages. The standard supports two encodings. UCS-2 is a two byte per character encoding that supports the first 64K characters. UCS-4 is a four byte per character encoding supporting the full range of characters.

Unicode [5] supports the same set of characters as ISO/IEC 10646-1.

Characters in XML are defined in terms of ISO/IEC 10646-1 and Unicode. The definition of a character in the XML recommendation [6] is

```
Char ::= #x9 | #xA | #xD | [#x20-#xD7FF] | [#xE000-#xFFFF] | [#x10000-#x10FFFF] /* any Unicode character, excluding the surrogate blocks, FFFE, and FFFF. */
```

When appropriate, interviewees were asked about their attitudes to Unicode. Awareness at this level was a lot less apparent than for XML. However, all those consulted are using databases that support Unicode, and the adoption of Unicode should be virtually transparent to them.

Unicode has gone through various versions. The first edition of XML 1.0 specified the use of Unicode version 2.0. The third edition specifies version 3.2 of Unicode. Unicode is currently at version 4.1.0. However, each new version only adds characters, rather than changing existing definitions. Versions are therefore backwards compatible, and XML is pretty lax about specifying the version, with different editions of XML 1.0 specifying the same subset of different Unicode versions.

Since all definitions in the CDD specify character sets that are a subset of Unicode common to all versions, there is no need to specify the version of Unicode to be used.

3.3 Use of UTF-8

Various encodings of UCS-2 and UCS-4 are available to reduce the number of bytes of data required to represent a given character string. The most appropriate to use depends on the range and distribution of characters to be used.

The XML recommendation specifies that "All XML processors *MUST* accept the UTF-8 and UTF-16 encodings of Unicode 3.1". An XML processor can detect which encoding is being used, and no other encodings are mandatory. For compatibility with all XML processors, it is therefore important that only these two encodings are used.

UTF-8 [7] encodes all ASCII characters in a single byte with the top bit clear, using the same code points as ASCII. Other characters are encoded as multi-byte sequences where each byte in the sequence has the top bit set.

UTF-16 [8] encodes all characters in the range U+0000 to U+FFFF as two byte sequences. Characters with codes from U+10000 to U+10FFFF are encoded as a four-byte sequence. Reserved codes in the Unicode code set are used to indicate when a four-byte sequence is being used.

When only the ASCII character set is being used, UTF-8 therefore uses only half as many bytes to encode a given character string as UTF-16. Where the character string uses mainly characters from the ASCII set, UTF-8 will still be far more efficient than UTF-16 in terms of file or message sizes.

Since XML processors must support both UTF-8 and UTF-16 encodings and be able to detect which is in use, there is no need to specify the encoding to use. This really only becomes an issue if people try to use, say, a legacy text editor on an XML document that contains only characters from the ASCII code set. However, for the data that will be transferred under the MIAP programme, UTF-8 will always be more efficient, so this is strongly recommended.

4 Character Sets and Languages

As part of the Tranche 1 implementation, a study was carried out to decide on appropriate character set and language support. Details of this study are included in Appendix A.

The conclusions of this study were:

- The general policy is to support all Latin-based characters for names, addresses and general text fields, but not non-Latin characters.
- All Unicode code charts for Latin characters are supported. These are Basic Latin (excluding the C0 control characters), Latin-1 (excluding the C1 control characters), Latin Extended A, Latin Extended B and Latin Extended Additional. This set corresponds to Unicode code points U+0020 to U+007F and U+00A0 to U+024F.
- Schemas are built in such a way that an individual project can further restrict the set if required.

The character set chosen will support Welsh and Gaelic languages as well as all European and most other languages using a Latin-based character set.

4.1 Implications of the Character Set Support

Like all interoperability projects, the MIAP project involves moving data from one system to another. These systems may be incompatible in several ways, one of which is in the character sets supported. Tranche 1 of the MIAP CDD defined the characters that would be allowed when transferring names and addresses. Tranche 2 defines character sets for other data. This section considers the interoperability issues of the use of these character sets.

Some systems support characters outside the "standard" English alphabet (e.g. the German ß) and characters with accents and other diacritics, and some do not. Some may not even support lower case characters, although none are currently known to fall into this category.

Even where systems support the same character sets, text may not be stored in the same way. For example, someone called Peter Müller may fill in a paper form from which his name is entered into a system as Peter Müller. Meanwhile, he fills in an online form, but does not want to bother with accented characters, so enters his name as Peter Mueller. For this reason, MIAP uses codes where possible (in this case, the Unique Learner Number might be applicable).

Unfortunately, XML cannot solve the problem of incompatible data, but this section is intended to raise and help resolve the issues.

In preparing this section, I have contacted both the Government Schema Group and Interoperability Working Group.

4.2 Roman-Based Character Issues

Data transferred as part of the MIAP programme has three possible uses:

1. displaying data from one or more systems (for example, as part of the learners' portal);
2. copying or moving data from one system to another; and
3. comparing data from one system to data on another.

Each of these is considered in turn.

4.2.1 Displaying Data

Displaying data should not normally cause any problems. It would generally be a mistake to carry out any conversion on data to be displayed, as many conversions (for example, converting a name from upper case to mixed case) cannot be automated reliably.

When displaying data from multiple sources, a suitable key should be used to identify the records to retrieve from each source. The use of plain text should be avoided. If this is not possible, see the section on comparing data.

4.2.2 Copying or Moving Data

Very few conversions of character sets can be automated reliably. The definition of the conversion of a specific character may depend on the language in which it is being used. For this reason, names, addresses and other text fields can contain an optional `xml:lang` attribute to indicate the language in use.

There are three main ways to treat data that is received with characters outside the normal range of 52 alphabetic, 10 numeric and common punctuation characters:

1. You can import the data exactly as it is received. This works well if you know that you support the full character set used, or if your system will degrade gracefully, for example, by replacing an unknown character with a null, and you are happy with any side-effects.
2. If you know the correct conversion for all characters and words that might be in received data, you could automate this conversion.
3. You can check each incoming record against the set of character codes that you support. You can then "quarantine" records that contain character codes outside this set and review these records manually.

If your system only supports upper case characters and you receive mixed case, the conversion is obvious and simple. The other way round causes problems. Although many names will convert correctly by converting all but the initial letter to lower case, this is not always true. In particular, names starting with "MC" or "MAC" need special treatment, as will those containing apostrophes. Converting "MCDONALD" to "McDonald" can be automated easily enough, but "MACDONALD" could convert to either "Macdonald" or "MacDonald".

4.2.3 Comparing Data

Where possible, comparisons should be made using codes rather than textual data. Where a textual comparison is unavoidable, it is best to translate both strings to upper case, and then do a fuzzy comparison using the greatest possible amount of data (e.g. both name and address, rather than just name). If a coded piece of data can be associated with the record, that is even better. For example, comparing a name and date of birth provides good reliability. Algorithms and code for fuzzy searching are freely available on the Internet.

All textual searching is very prone to errors, so coded identifiers should be used whenever possible

5.1 The Common Data Definitions

5.1.1 Introduction

Oakleigh Consulting defined the CDD in the following documents:

1. Common Data Definitions for the MIAP Learning Interface (Part 2) - Final Report and Data Definitions [1]
2. Common Data Definitions About Providers for the MIAP Learning Interface [15]
3. MIAP Programme Common Data Definitions for the MIAP Learning Interface - Tranche 2 – Learner Participation and Achievement Data Final Report [16]

Note that the XML version of the definitions (see 5.1.2) is definitive and the only version being maintained. This can be viewed as either HTML or Adobe PDF.

5.1.2 Formatting the CDD like the GDSC

Boynings Consulting was asked to represent the Common Data Definitions in XML in the same way as the Government Data Standards Catalogue [17]. This format is based on version 2 of the e-Government Metadata Standard with extensions for additional metadata elements. The CDD requires further additional metadata, and so these elements were added in a similar way to the previous additions. The stylesheets were then altered to render these additional items.

As a result, the definitive set of data definitions is now held in XML and can be rendered as HTML or PDF.

While creating this, some additional fields were added to the Oakleigh Consulting Definitions. These provide implementation notes and, for HTML only, links to the schema definitions.

A specific requirement was to indicate the syntax and semantics of the definitions. Oakleigh provided these in the fields "Business Format" for syntax, and "Values" and "Validation" for semantics. These sections have been expanded where it was felt necessary.

For the HTML display, the e-GU stylesheets have been used and extended to display the additional information. For the PDF display, the stylesheets have been re-written to match the Oakleigh format. This was done in part to reduce the size of the document.

6 Full Technical Implementation - XML Policy

The key words "must", "must not", "required", "shall", "shall not", "should", "should not", "recommended", "may", and "optional" in this section are to be interpreted as described in RFC 2119 [8].

This policy supplements the e-Government Schema Guidelines for XML [18].

6.1 Adherence to e-GIF Standards

6.1.1 Policy

Where possible, e-GIF standards must be used. Where this is not possible, a rationale for the variation must be provided. In particular, variations from the Government Data Standards Catalogue must be justified.

6.1.2 Rationale

The e-GIF is mandatory for use across the UK public sector, and so is applicable to this project. Failure to comply with the e-GIF can affect the result of Gateway Reviews and affect project funding.

6.2 Use of Elements and Data Types

6.2.1 Policy

Data types should be defined for groups of elements that use similar type definitions (such as all date elements). Element definitions should only use these types (i.e. they will not use the XML Schema built-in types).

6.2.2 Rationale

There are several elements that use similar types, so it is worth defining types for these. Because of the need to represent nulls and reasons for nulls, XSD data types will generally not be suitable. To maintain flexibility, they will not be used at all in element definitions.

An alternative approach for a set of common definitions is to use a separate data type for each element (the MOD works like this). However, this results in repetition and so makes maintenance harder.

6.3 Representation of "no data"

6.3.1 Policy

Where there is not a specific code for "no data" (such as code "0" for Person Gender), an empty string should be used. This must be accompanied by a reason code.

6.3.2 Rationale

There are two ways to do this. One is to use an impossible value (such as a date of 9999-99-99 or 1999-01-01), the other way is to use a specific text string. The impossible value approach has been used for many years as being "database friendly". However, it is very much implementation specific (for example, some products might reject the first date shown above as not being a real date, while the second will be relevant in some contexts). For that reason, the value "null" is used to indicate lack of data. A data stream can be pre-processed (for example, using XSLT) to provide other values based on the data type.

6.4 Representing the Reason for Missing Data

6.4.1 Policy

Anywhere that a "null" value is used, it must be accompanied by a reason code:

- 1: not provided (reason not specified)
- 2: not sought (no request has been made for the information)
- 3: refused (information requested but not provided)
- 8: see supplementary field
- 9: not applicable

If the code is "8" additional information must be provided.

This code must be encoded in XML schema as optional attributes (see 6.5), with a business rule to indicate when it must be included.

6.4.2 Rationale

Free text is not database friendly and is hard to translate into a code, whereas a code can easily be translated to text when required.

6.5 Representing Metadata

6.5.1 Policy

Additional information (such as the reason for lack of data) shall be represented using attributes in XML Schema.

6.5.2 Rationale

Such data can be represented as elements or attributes. Elements are the more flexible approach, but result in needing another level of hierarchy. Attributes meet the requirements for this data and their use in this context matches the implementation of other e-GIF systems such as those in the HMRC, where currency codes (such as GBP) are implemented as an attribute of elements containing currency values.

The element approach would result in

```
<PersonBirthDate>
  <Value>1955-06-08</Value>
</PersonBirthDate>
```

or

```
<PersonBirthDate>
  <Value>null</Value>
  <Reason>1</Reason>
</PersonBirthDate>
```

While the attribute approach results in:

```
<PersonBirthDate>1955-06-08</PersonBirthDate>
```

or

```
<PersonBirthDate Reason="1">null</PersonBirthDate>
```

The second approach is much less verbose (which can be important with large data sets) and easier to read.

6.6 Attributes

6.6.1 Policy

Attributes shall not be defined globally, but defined locally to the element where they are used, using a globally defined simple type.

6.6.2 Rationale

UK GovTalk discourages the use of globally defined attributes as they lead to namespace prefix problems in instance documents (XML documents based on the schema).

6.7 Ids in schemas

6.7.1 Policy

The `id` attribute shall be used to show the CDD identifier. Where an element is defined globally, the `id` shall be used in the element definition. Where the identifier applies to an attribute, the `id` may be used in the definition of the simple data type used by the attribute definition.

6.7.2 Rationale

It may be useful to have the CDD identifier shown. These are the logical places to put it since we do not often use global attributes.

For example:

```
<xs:element
  name="PersonTitle"
  type="PersonTitleType"
  id="MIAPCDD-000-001"/>
```

6.8 Elements with Multiple Patterns

6.8.1 Policy

Where an element may be populated with data whose patterns vary according to the issuing authority, each pattern should be defined as a data type and the use of `xsi:type` specified on the element. A default pattern should also be provided. A Schematron [20] pattern should be provided that can be used to enforce this.

6.8.2 Rationale

Some elements contain data that could have been issued by one of several authorities. For example, the Qualification Accreditation Number (002 380) has a pattern defined by the Qualification and Curriculum Authority. However, in future, other authorities might define other patterns for a QAN. In the schema, each of these patterns is defined by a data type. An instance document can then specify which is in use using the `xsi:type` attribute. This allows validation of the pattern to take place.

XML Schema does not allow a schema to mandate the use of `xsi:type`. Schematron is another ISO standard schema language for XML that is complementary to XML Schema and allows such rules to be enforced. By embedding the Schematron rule in an `xs:annotation`, applications that wish to implement Schematron can apply the rule directly, while others might either ignore it or hard-code the rule into the application.

6.8.3 Example

simple type definitions defining patterns

```
<xs:simpleType name="GroupAwardNumberType"
  id="MIAPCDD-002-260">
  <xs:restriction base="xs:token">
    <xs:pattern value="[A-Z0-9]{4} [0-9]{2}"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="QualificationAccreditationNumberType"
  id="MIAPCDD-002-380">
  <xs:restriction base="xs:token">
    <xs:pattern value="[GQ][0-9]{7}"/>
    <xs:pattern value="[0-9]{7}[0-9X]"/>
  </xs:restriction>
</xs:simpleType>
```

element definition using `xsi:type` and including Schematron rule

```
<xs:element name="QualificationIdentifier"
  id="MIAPCDD-002-130">
  <xs:annotation>
    <xs:documentation>This element MUST include an xsi:type
attribute. This can currently have a value of either
QualificationAccreditationNumberType or GroupAwardNumberType.
There follows a Schematron rule that can be used to enforce the
presence of the attribute.</xs:documentation>
  <xs:appinfo>
```

```

    <sch:pattern name="Test for existence of correct xsi:type">
      <sch:rule context="miap:QualificationAccreditationNumber">
        <sch:assert
test="@xsi:type='QualificationAccreditationNumberType' or
@xsi:type='GroupAwardNumberType'">The
QualificationAccreditationNumber element must have an xsi:type
attribute with a value of QualificationAccreditationNumberType
or GroupAwardNumberType</sch:assert>
          </sch:rule>
        </sch:pattern>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>

```

usage in instance document

```

<QualificationIdentifier
xsi:type="QualificationAccreditationNumberType">1234567X</Qualif
icationIdentifier>

```

6.9 Use of `xml:lang`

6.9.1 Policy

Plain text fields that could contain data in different languages should allow use of the `xml:lang` attribute.

6.9.2 Rationale

As discussed in section 4, knowledge of the language of a text string can help in transliteration. This attribute can also be used to provide multiple versions of the same data in different languages. For example, a course may have a single identifier, but its title might be provided in both English and Welsh.

6.9.3 Policy

Where the `xml:lang` attribute is allowed, users of the schemas should consider whether multiple versions of the associated element should be allowed.

6.9.4 Rationale

This covers cases such as that above, where a course has a single identifier, but its title might be provided in both English and Welsh.

6.10 Use of Namespaces

6.10.1 Policy

Where existing definitions are used from external schemas, these should normally be imported into the MIAP CDD schemas and retain their original target namespaces.

6.10.2 Rationale

This makes the origin of the definition clear.

6.10.3 Policy

Where existing definitions have to be modified, they shall be moved into the MIAP CDD namespace. Where there are several definitions being used from a single source (such as the e-GIF Address and Personal Details schema) and several have to be modified, all related definitions shall be moved into the CDD MIAP namespace.

6.10.4 Rationale

This applies mostly to the UK GovTalk™ definitions, where character set restrictions make many of them unsuitable. Since the definitions are being altered, it would cause confusion and break the namespace governance to keep them in the UK GovTalk™ namespace. It is pointless and confusing to keep the few remaining definitions within the GovTalk namespace.

7 Full Technical Implementation - UK Addresses

Although not originally asked to look specifically at address formats, we agreed that my experience with other projects could provide useful input to the work of Oakleigh Consulting. This was therefore included in the consultation.

There has been a certain amount of discussion regarding the address formats to use in the CDD. This arises in most interoperability projects because of the different levels of granularity in which addresses are held in different systems. For this reason, this is not just a data format problem - if data is held in incompatible formats, no amount of XML will make them compatible. It is therefore important to recognise that some people may need to modify systems to meet the format on which we eventually settle and there may be a migration to the standards selected.

There are two basic ways in which addresses can be held and transmitted - formatted (e.g. BS7666 [20] and Postcode Address Format (PAF[®]) [22]) and unformatted (or more correctly, semi-formatted as at least the postcode is usually held in a field of its own). The UK GovTalk[™] `UKPostalAddressStructure` is an example of an unformatted address structure.

In most cases, learner addresses are held for statistical purposes and institution addresses for other purposes. Most statistics are based around the postcode, making this an essential field.

Note that the ODPM is currently running a project to create a National Spatial Address Infrastructure (NSAI) in three phases due to end in late 2007. Unfortunately, this project is currently delayed. Meanwhile, the National Land and Property Gazetteer (NLPG) is moving forwards and should have a complete database of BS7666 addresses with maintenance in place by October 2006. Commercial organisations are developing tools (such as mapping between NLPG and PAF[®]) to help use the database. Meanwhile, BS7666 will be updated during 2006.

7.1 Alternative Formats

I will describe five formats here - BS7666, The UK GovTalk[™] `UKPostalAddressStructure` and `InternationalAddressStructure`, the PAF[®] and a format proposed by Neil Catton that has resulted from work on the UKRLP project. The diagrams were generated by XML Spy 2006.

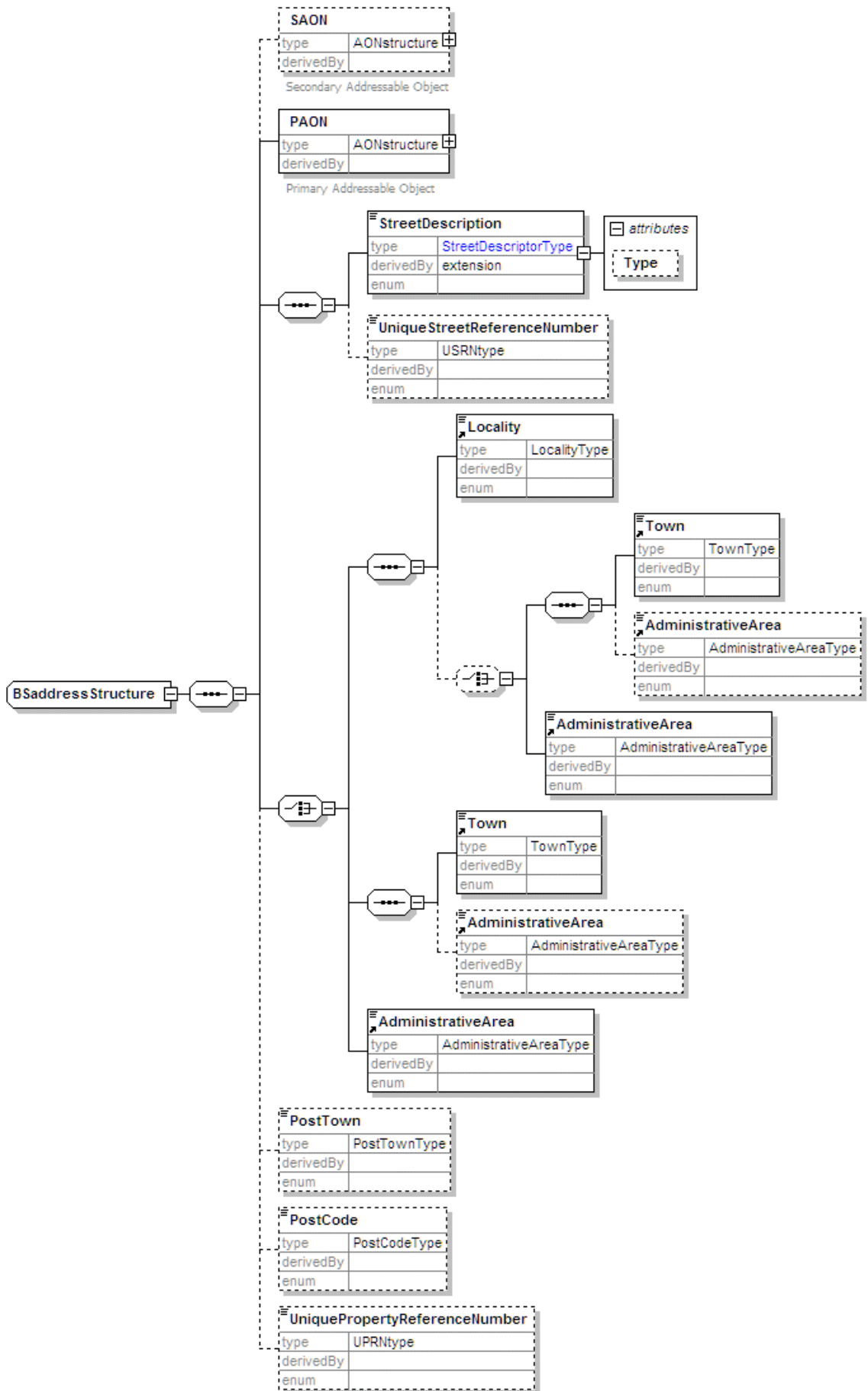
7.1.1 BS7666

The purpose of a BS7666 address is to identify a unit of land or property. The BS7666 address of a property with a postal address can generally be translated into a postal address fairly readily. The 2006 version of BS7666 will improve on the support for postal addresses.

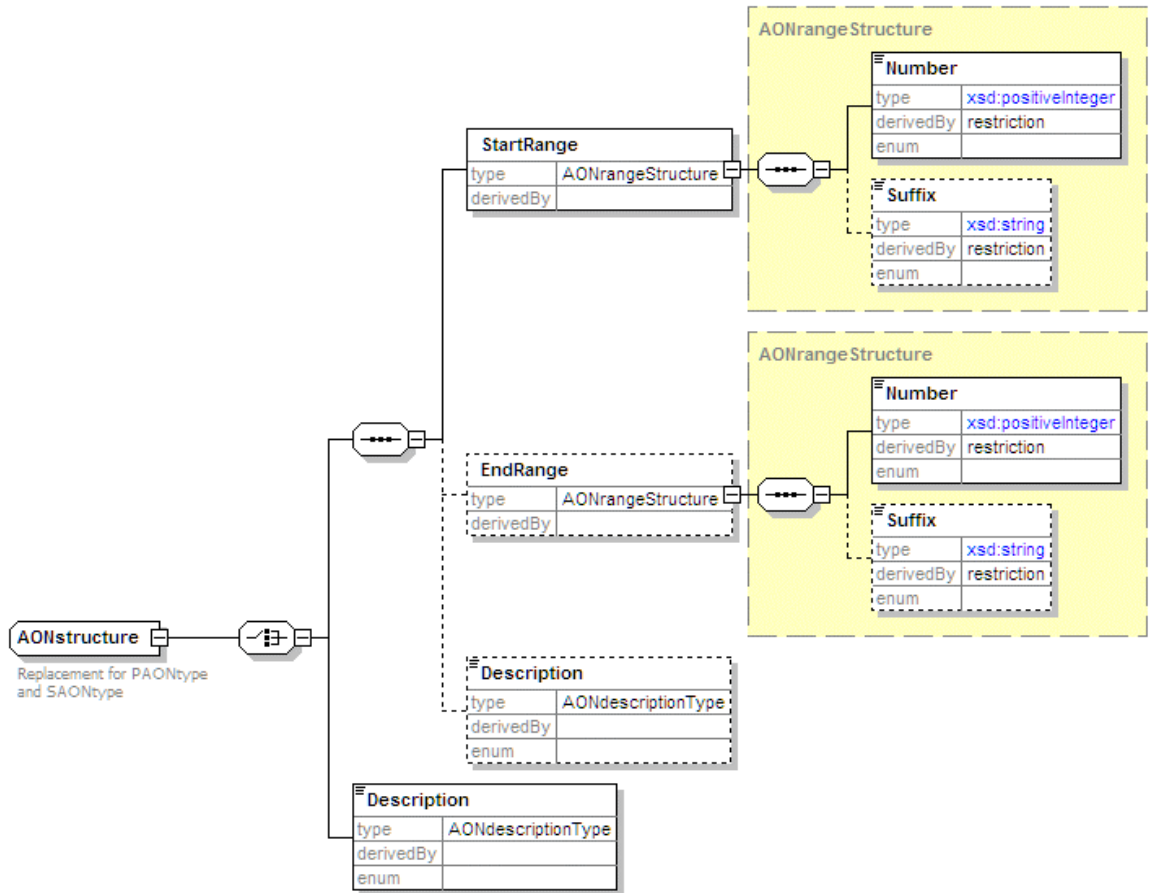
UK national and local government is moving to BS7666 as the main addressing standard, although this has some opposition, particularly from those who only require a postal address. One current handicap with using this format is that there is no way for most people to convert from a Unique Property Reference Number (UPRN) to a full BS7666 address in the way

that they can use a PAF[®] file to convert a house number and postcode into a full postal address. The NSAI may address this.

UK GovTalk[™] has developed an XML schema to describe a BS7666 address. This comprises definitions of the various parts and a sample of the way they can be put together to include the data most people require. This is:



The structure of the SAON and PAON is:

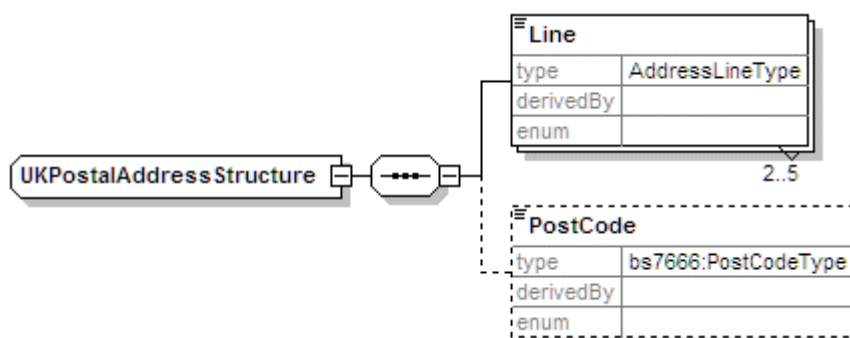


In favour of the BS7666 format is that a National Land and Property Gazetteer is being built, and that it is the preferred addressing standard for the UK public sector. Against it is that not everyone holds address data at the level of granularity required to create a BS7666 address. This applies particularly to a street address such as "1 High Street", which needs to be separated into a primary addressable object ("1") and a street description ("High Street"). It is not possible to create an automated means of doing this that will be 100% accurate.

CBDS uses a BS7666 formatted address format, although it uses postal addresses within this format.

7.1.2 UK GovTalk™ UKPostalAddressStructure

This is the second format used in the public sector, and is recommended for use when the BS7666 format is not suitable. The format is:



This is a simple format that is easy to produce from any system holding address information. However, it is not possible to parse into a more structured format. It is therefore unsuitable if this is a requirement.

7.1.3 PAF[®] Address Format

The PAF[®] address [22] is a structured address format for postal addresses. It is not a preferred address structure in UK GovTalk[™]. The PAF[®] itself is a complete and maintained database of postal addresses, and there are many services available that add value to it.

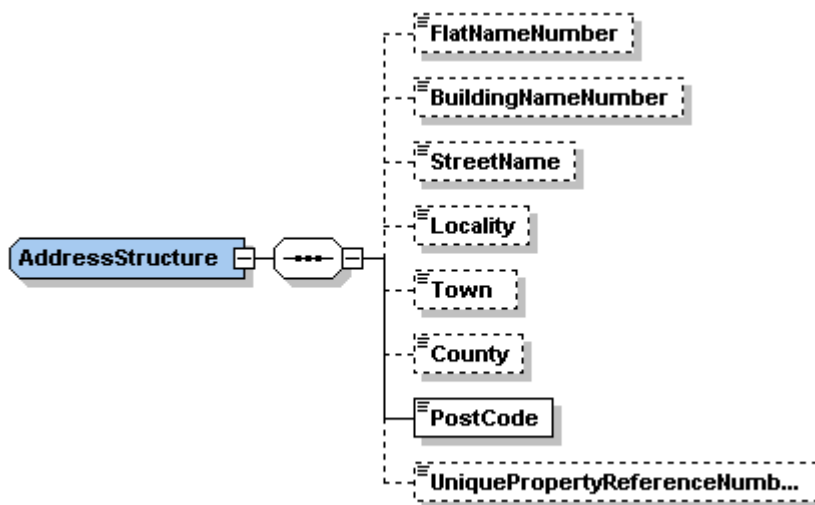
Basically, this is a structured address format relating to mail delivery. It specifies:

- Premises Elements
 - Sub Building Name
 - Building Name
 - Building Number
- Thoroughfare Elements
 - Dependent Thoroughfare Name
 - Dependent Thoroughfare Descriptor
 - Thoroughfare Name
 - Thoroughfare Descriptor
- Locality Elements
 - Double Dependent Locality
 - Dependent Locality
 - Post Town
 - County
 - Postcode

7.1.4 UKRLP Proposed Address Format

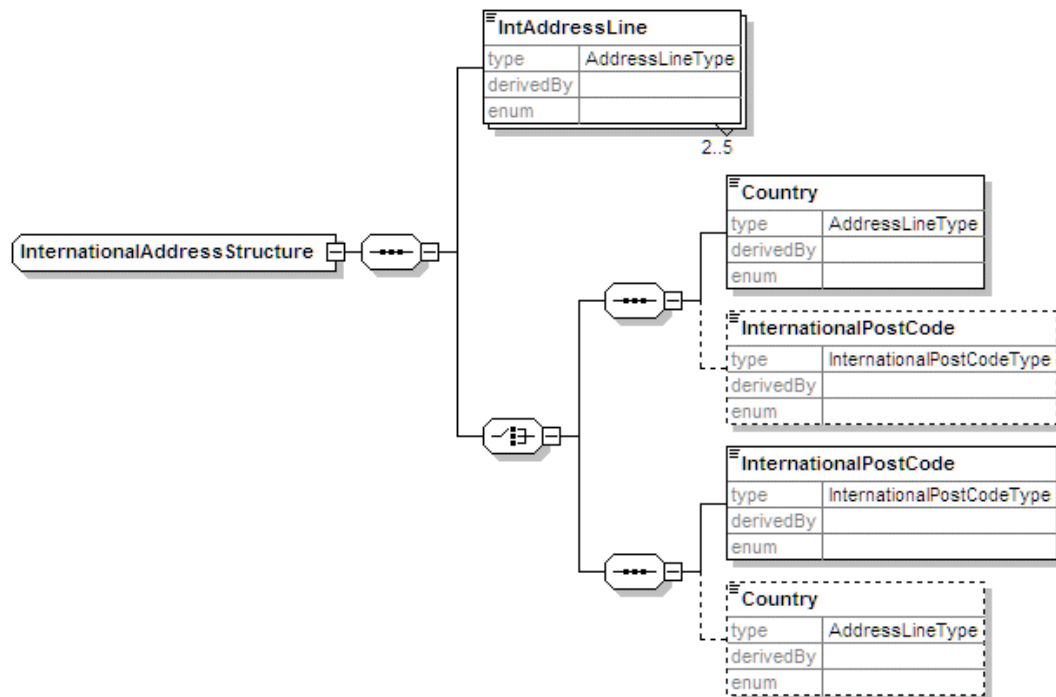
Currently, UKRLP is using the BS7666 address format, but without true BS7666 data in the same way that CBDS is.

However, UKRLP has found that an unstructured address is also not suitable as it cannot be reliably validated. The project has therefore proposed an alternative structured format:



7.1.5 UK GovTalk™ InternationalAddressStructure

This is the e-GIF format for non-UK addresses. In other applications, it is sometimes used for all addresses since it can represent UK addresses as well. However, this is not recommended for MIAP since its use will be rare and it does not provide any validation of the post code.



7.2 Recommendation

The solution adopted is to support three address formats - BS7666, unstructured and international. For the first two, any verification of the address can also be carried in the XML data. This verification can be any of:

- "Unverified"

- "NLPG" if the address has been verified against the National Land and Property Gazetteer
- "NSAI" if the address has been verified against the National Spatial Address Infrastructure
- "LLPG" if the address has been verified against a Local Land and Property Gazetteer
- "CBDS-formatted version of AddressPoint" if the address has been verified against the CBDS-formatted version of AddressPoint
- "PAF" if the address has been verified against a Postcode Address File

The first, and recommended for use whenever possible, is the BS7666 format. If the verification is any of "NLPG", "LLPG", or "NSAI", this must be a real BS7666 address. Otherwise, the address is a postal address formatted into the BS7666 format. If using this format with postal addresses, property names and numbers should ideally be separated as for BS7666 and within the PAF. However, if this is not possible, the following examples should provide guidance on the best formatting:

Address	Preferred	Alternative
1 High Street	PAON/StartRange: 1 StreetDescription: High Street	StreetDescription: 1 High Street
Rose Cottage 1 High Street	PAON/StartRange: 1 PAON/Description: Rose Cottage StreetDescription: High Street	PAON/Description: Rose Cottage StreetDescription: 1 High Street
1 Railway Cottages High Street	SAON/StartRange: 1 PAON/Description Railway Cottages	PAON/Description: 1 Railway Cottages Street Description: High Street
Rose Cottage Somevillage	PAON/Description: Rose Cottage	PAON/Description: Rose Cottage

It is important to note that the alternative formats above are only to be used for postal addresses, and then only if the preferred format cannot be created automatically from the data.

The unstructured address format in e-GIF should be used for BFPO addresses. In this case, the Unit Name should be in an address line and the BFPO number in the postcode field. This avoids the problem of using the postcode NW7 1PX that is associated with all BFPO addresses for statistical analysis for which it is unsuitable. This format can also be used when only a postal address is required. However, because this is difficult to validate against the PAF, it should be a last resort.

The international address format has a number of address lines followed by at least one of postcode and country and should be used only for overseas addresses.

8 Full Technical Implementation - XML Schemas

The CDD is implemented as a set of XML schemas. This section provides some explanatory notes relating to the schemas.

All schemas have been tested with XML Spy 2006 SP1, MSXML4 SP2, Xerces-J 2.6.2 and XSV 2.10-1 and been subject to external QA for which a report is available.

The schema files fall into two groups, plus two additional files.

The first group contains five files of element definitions. These relate to:

- Awarding Organisation
- Participation / Achievement
- Person / Learner
- Provider
- Qualification

The CDD implementation makes use of externally-provided schemas where possible. Where these do not exist, but the schemas developed for the CDD are of more general use, these are also treated as external. These schema documents are in a separate folder from the element definitions and comprise:

- the UK GovTalk™ BS7666 schema modified for use with the CDD
- the UK GovTalk™ Citizen Identification Types schema
- the UK GovTalk™ Common Simple Types schema
- the UK GovTalk™ Organisation Identifier Types schema
- the UK GovTalk™ Person Descriptive Types schema
- the UK GovTalk™ Language Identification Code schema
- the CBDS Ethnicity Types schema
- a Country Classification Code schema developed for the MIAP programme using the ONS classifications developed for the 2011 census
- the W3C schema for XML

The additional files comprise a "CDD Core" containing data types used within the CDD and a file "CDD.xsd" just comprising `xs:include` statements.

Each CDD schema file has the same target namespace. For this reason, any schema document importing schema definitions from more than one file should create a simple schema document containing only `xs:include` statements and import this instead. "CDD.xsd" is an example of such a document. See the e-Government Schema Guidelines for XML [18] for more details.

The definitions in the CDD Core are:

Simple Data Types	Complex Data Types
AddressLineType AddressSetType AddressVerificationType CharityRegistrationNumberType CompaniesHouseReferenceNumberType DateType EmailAddressType GenericIdentifierType GenderType GroupAwardNumberType GradeCodeType GradeSequenceType GradeType InternationalPostCodeType LADcodeType MarkType NameLineType NameSetType NINOType OrganisationNameSetType OrganisationNameType PersonAddressQualifierType PersonBirthDateVerificationType PersonDeathDateVerificationType PersonTitleType QualificationAccreditationNumberType QualificationLevelCodeType ReasonCodeType ResultOfApplicationType ScottishCandidateNumberType TelephoneNumberType TitleSetType TitleType TownType UKPRNtype UniqueLearnerNumberType UniqueTaxReferenceType WebURLtype YesNoType YesNoUnknownType	AddressStructure DateStructure InternationalAddressStructure NameStructure OrganisationNameStructure TitleStructure UKpostalAddressStructure

8.1 Changes to Element Names

The names of the elements representing the data definitions are created by removing spaces from the names of the data definitions themselves. For this reason, minor changes to capitalisation and the names themselves have been made to ensure that element names meet the requirements of the e-Government Schema Guidelines for XML.

8.2 Definitions Relating to Addresses

Because of the redefinition of address formats, the definitions for "...AddressLine" and "...Postcode" have been removed from the CDD and the implementation.

8.3 Definitions Relating to Telephone Numbers

Note that Telephone Number must be all numeric. Some databases may have telephone numbers with extensions containing a text string such as "ext". The extension must be removed before transferring the data. In many cases, this will be an opportunity to update data by replacing a switchboard number and extension with a direct dial number.

8.4 PersonCountryOfDomicile and PersonNationality

These two elements currently allow the full list of codes used by the Office of National Statistics (ONS) and compatible with ISO 3166. Three formats are supported - alpha-3 (e.g. GBR), alpha-2 (e.g. GB) and numeric-3 (e.g. 826). The alpha-2 code is preferred, but there is a 1:1 correspondence, so transliteration is simple. Restrictions on the code set for use for each of `PersonNationality` and `PersonCountryOfDomicile` are likely to be introduced at a later stage in line with work being undertaken by the ONS. As an example, it is likely that XF (England), XH (Scotland) and XI (Wales) will be retained for use as a country of domicile, but not allowed for nationality, which would be GB in each of these cases.

8.5 "Data Items Which Cannot Currently Be Implemented"

The Oakleigh tranche 2 report [15] identifies a list of data items that cannot be implemented. Rather than leave these out of the implementation, they have been implemented in the following generic ways:

CDD ID	Name	Implementation
002 130	Qualification Identifier	Uses <code>xsi:type</code> to make either a <code>GroupAwardNumberType</code> or a <code>QualificationAccreditationNumberType</code>
002 280	Awarding Organisation Identifier	<code>GenericIdentifierType (xs:token)</code>
002 520	Unit Identifier	<code>GenericIdentifierType (xs:token)</code>
002 560	Awarding Organisation Classification Group	<code>GenericIdentifierType (xs:token)</code>

	Identifier	
002 590	Awarding Organisation Classification Identifier	GenericIdentifierType (xs:token)
003 230	Grade Achieved	GradeType (xs:token)
003 240	Grade Code	GradeCodeType (xs:token)

8.6 Use of `xml:lang`

This attribute, which is built into the XML language, can be useful in identifying the language of a text string. For example, a course title might be presented in both English and Welsh. It has therefore been allowed on the generic `TitleStructure`, which is used for all Title definitions, `NameStructure` which is used for people's names, `OrganisationNameStructure`, which is used for both the legal and trading names of organisations and the `BSaddressStructure`.

9 Contributors

The following contributed to this report, either through formal interviews or contributing useful replies to specific questions.

Douglas Ansdell	Scottish Executive
Peter Ashton	Learning and Skills Council
Susanne Aspinall	Learning and Skills Council
Helen Baird	Oakleigh Consulting
Henry Bloomfield	Home Office
Peter F Brown	European Parliament
Jason Bryant	Learning and Skills Council
Neil Catton	Hotcourses
Ruksana Chowdhury	Ofsted
Mike Coulson	MIAP ULN project (LSC)
Mark Cummings	DfES
Natasha Daly	TDA
Alex Daybank	Information Commissioner's Office
Arturo Dell	Camden Council
John Duffy	Scottish Higher Education Funding Council
Hannah Falvey	Higher Education Funding Council for Wales
Eugene Fernandez	Ofsted
Allan Findlay	Student Loans Company
Nigel Gibson	DfES
Andy Greener	HMRC
Anna Harvey	Local e-Government Standards Body
Liz Heal	Higher Education Funding Council for Wales
Andy Hesketh	DfES
Stephen Howarth	Adult Learning Inspectorate
Martin Hughes	Scottish Qualifications Authority
Judy Jerome	Cabinet Office e-Government Unit
Terry Keane	DfES
Sarah King	Grampian Valuation Joint Board
Mark Laybourne	CEO, xPress Software
Sue Matthews	UCAS
Seumas McClenahan	Qualifications and Curriculum Authority
Vic McFettridge	DWP
Elizabeth McLaren	Office for National Statistics
David Morgan	UCAS
Ainga Pillai	Camden Council
Richard Puttock	Higher Education Funding Council for England
Keith Roberts	Oakleigh Consulting
Dominic Rouse	HESA
Robin Sibson	Higher Education Statistics Agency
Amritpal Singh	Home Office
Stuart Smith	UCAS
Vid Vartak	Hotcourses
Ann Wass	DfES
Lowri Williams	Welsh Language Board
Lohan Wolf	Qualifications and Curriculum Authority
Paul Woolman	NHSis
Andy Youell	Higher Education Statistics Agency

10 References

1. XML Schema Part 1: Structures Second Edition *World Wide Web Consortium (W3C)* 28 October 2004 <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>
2. XML Schema Part 2: Datatypes Second Edition *World Wide Web Consortium (W3C)* 28 October 2004 <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>
3. e-Government Interoperability Framework Version 6.1 *Cabinet Office e-Government Unit* 18 March 2005 [http://www.govtalk.gov.uk/documents/eGIFv6_1\(1\).pdf](http://www.govtalk.gov.uk/documents/eGIFv6_1(1).pdf)
4. ISO/IEC 10646:2003 Information technology -- Universal Multiple-Octet Coded Character Set (UCS) *International Organization for Standardization (ISO)* 2003
5. The Unicode Standard 4.0 *The Unicode Consortium* 2004
6. Extensible Markup Language (XML) 1.0 (Third Edition) *World Wide Web Consortium (W3C)* 4 February 2004 <http://www.w3.org/TR/2004/REC-xml-20040204>
7. RFC 2279 - UTF-8, a transformation format of ISO 10646 *The Internet Society* 1998 <http://www.faqs.org/rfcs/rfc2279.html>
8. RFC 2781 - UTF-16, an encoding of ISO 10646 *The Internet Society* 2000 <http://www.faqs.org/rfcs/rfc2781.html>
9. Guidance on Meeting UK Government Commitments in Respect of Irish And Ulster Scots (Version 2) *Northern Ireland Department of Culture Arts and Leisure* August 2005 http://www.dcalni.gov.uk/ContMan/includes/upload/file.asp?ContentID=1100&file=c_23
10. DCAL Language Policy for Irish And Ulster-Scots *Northern Ireland Department of Culture Arts and Leisure* August 2005 <http://www.dcalni.gov.uk/foi/document.asp?doc=676>
11. Gaelic Orthographic Conventions 2005 *Ùghdarras Theiteanas Na H-Alba* August 2005
12. Bilingual Software Guidelines and Draft Standards *Welsh Language Board* December 2004
13. Welsh Language Scheme *Department for Education and Skills* http://www.dfes.gov.uk/cymraeg/welshlang_e_report.shtml
14. Common Data Definitions for the MIAP Learning Interface (Part 2) - Final Report and Data Definitions *Oakleigh Consulting Ltd* May 2005
15. Common Data Definitions About Providers for the MIAP Learning Interface *Oakleigh Consulting Ltd* 31 May 2005
16. MIAP Programme Common Data Definitions for the MIAP Learning Interface - Tranche 2 – Learner Participation and Achievement Data Final Report *Oakleigh Consulting Ltd* January 2006

17. Government Data Standards Catalogue *Cabinet Office e-Government Unit*
<http://www.govtalk.gov.uk/gdsc/html/>
18. e-Government Schema Guidelines for XML Version 3.1 *Office of the e-Envoy*
January 2004 [http://www.govtalk.gov.uk/documents/schema-guidelines-3_1\(1\).doc](http://www.govtalk.gov.uk/documents/schema-guidelines-3_1(1).doc)
19. Key words for use in RFCs to Indicate Requirement Levels *IETF* March 1997
<http://www.ietf.org/rfc/rfc2119.txt>
20. ISO/IEC 19757 Document Schema Definition Languages (DSDL): Part 3: Rule-based validation — Schematron *ISO/IEC* <http://www.schematron.com/iso/dsdl-3-fdis.pdf>
21. BS 7666 Spatial Datasets for Geographical Referencing *British Standards Institution* 2002
22. Why What and How - The Guide to Using the PAF[®] File *Royal Mail* December 2003 ftp://ftp.royalmail.com/Downloads/public/ctf/rm/PAF_Digest_Dec_03.pdf

Appendix A. Character Sets and Languages - Background

Unicode supports virtually every character in every written language. It is therefore important to consider limiting the range of characters available. Too strict a limit could mean that some required characters (either now or in the future) are forbidden, while too lax a limit means that people may not be able to read text (for example, names using Chinese characters) and increases the probability of errors not being detected.

Most MIAP CDD data is coded in some way, so character set issues do not arise. In tranche 1 of the CDD, the exceptions relate mainly to the names and addresses of individuals and institutions. Tranche 2 will add other free text definitions, such as qualification names, where the required language support and character set will have to be considered.

Character sets are considered here from three aspects: the legal implications, common practice in the public sector and the requirements and capabilities of the MIAP participants.

A.1 Legal Aspects

The following laws and charter can be considered to apply to this project:

- The Welsh Language Act 1993
- The Gaelic Language (Scotland) Act 2005
- The Data Protection Act 1998
- The European Charter for Regional or Minority Languages

In many cases, there are guidelines that interpret these:

- Guidance on Meeting UK Government Commitments in Respect of Irish And Ulster Scots (Version 2) [9]
- DCAL Language Policy for Irish And Ulster-Scots [10]
- Gaelic Orthographic Conventions 2005 [11]
- Bilingual Software Guidelines and Draft Standards [12]
- DfES Welsh Language Scheme [13]

The focus of the various laws and guidelines on minority languages is in making information available in the minority language. There appears to be little in terms of supporting data such as names and addresses in these languages. The Welsh Language Act and Gaelic Language (Scotland) Act refer to producing language schemes if asked. The DfES has produced a Welsh Language Scheme. This does not refer to character set support.

Under the fourth principle of the Data Protection Act, individuals can insist that data held about them is correct. Initial advice from the DfES Data Protection Officer indicated that this might mean supporting accented characters. However, advice from

the Information Commissioner is that "the Commissioner does not feel it is necessary for organisations to support the use of accented characters for the electronic recording of personal data such as names" because "it can often be difficult for certain computer systems and software to easily make use of these accented symbols and we feel it would be unreasonable to require data controllers to make provision for this". Given that the reason for not requiring the support of accented characters is the difficulty of implementation, and that this is becoming simpler all the time as more systems provide Unicode support, it is likely that this advice will change over time.

A.2 Other Public Sector Bodies

Information for this section was gathered from:

- HM Revenue and Customs
- Department of Work and Pensions
- Home Office (for the Immigration & Nationality Directorate, Passport Agency and UK Identity Card Programme)
- Cabinet Office e-Government Unit
- Department for Constitutional Affairs (and software vendors) for electoral registers
- Office for National Statistics
- Grampian Valuation Joint Board
- Camden Council
- Gaelic Unit, Scottish Executive
- Welsh Language Board
- Northern Ireland Office Department for Culture and Leisure
- Local e-Government Standards Body
- NHSiS

In general, character set support seems to be "below the radar". The e-GIF XML schemas for items such as names and addresses use a very restricted character set (a subset of ASCII) that is compatible with most public sector systems. However, several new systems developments (such as the Coordinated Online Record of Electors) have found this to be inadequate. The usual solution is to remove all character set restrictions. When I raised this issue at a Government Schema Group meeting, there was a consensus that the restriction should be removed completely, leaving it up to individual projects to apply their own restrictions if necessary.

Departments such as HMRC and DWP using only ASCII for personal data such as names and addresses are beginning to find this to be a problem, with people asking for the accented characters in their names to be used.

Although many areas of Government make information available in languages that do not use a Latin character set, I have found only one (HMRC for Customs data, which supports Cyrillic characters) that supports non-Latin characters in databases. The Immigration & Nationality Directorate, for example, translates names into Latin script. ISO 9:1995 specifies one method of rendering Cyrillic characters in a Latin alphabet. This translation requires accented characters and is reversible if the target language is known. ISO 843:1997 does the same for Greek, and there are others for other non-Latin languages.

There is currently a project in progress to allow online registration of births. This will allow accented characters.

Passports and the National Identity Card programme use the recommendations of ICAO 9303. This specifies the character set defined by ISO 1073-3:1976 for those parts of the document that are machine-readable and does not restrict the set otherwise.

Note that some addresses use non-ASCII characters. For example the address of the Association of London Government is 59½ Southwark Street.

A.3 MIAP Participants

Typically, MIAP participants specify either ASCII or ISO 8859-1 or do not specify what character sets should be used in interactions. In the last case, the applications may only support ASCII characters. Although problems might be expected with this approach, they have not been found in practice. The conclusion is that either wider character sets are not being sent, or, if they are, incorrect characters are being stored and not being noticed.

Several participants felt that accented characters should be supported in the future, but that non-Latin characters were not required. Note that at least one Higher Education institution (Sabhal mòr Ostaig) has an accented character in its name.

One institution has had long discussions with a MIAP participant that cannot deal with accented characters, as a result of which the institution refused to transfer data.

A.4 Required Unicode Characters for UK Languages

Unicode is split into several code charts¹ (see <http://www.unicode.org/charts/>). Those for Latin-based characters are:

Code Chart	Character Range	Notes
Basic Latin	U+0000 - U+007F	Basic ASCII. Includes C0 control characters U+0000 - U+001F
Latin-1	U+0080 - U+00FF	Includes characters required for Welsh and Gaelic languages. Includes C1 control characters U+0080 - U+009F
Latin Extended A	U+0100 - U+017F	Required for Irish Gaelic dotted consonants.
Latin Extended B	U+0180 - U+024F	
Latin Extended Additional	U+1E00 - U+1EFF	Required for Irish Gaelic dotted consonants.

¹ Users of more recent versions of Windows can view Unicode characters using the Character Map utility to view the Lucida Sans Unicode typeface.

The Welsh language requires the following codes:

Circumflex		Acute		Grave		Diaeresis	
Â	194	Á	193	À	192	Ä	196
Ê	202	É	201	È	200	Ë	203
Î	206	Í	205	Ì	204	Ï	207
Ô	212	Ó	211	Ò	210	Ö	214
Û	219	Ú	218	Ù	217	Ü	220
Ŵ	372	Ŷ	7810	Ẁ	7808	Ẃ	7812
ÿ	374	Ý	221	Ỳ	7922	ÿ	376
â	226	á	225	à	224	ä	228
ê	234	é	233	è	232	ë	235
î	238	í	237	ì	236	ï	239
ô	244	ó	243	ò	242	ö	246
û	251	ú	250	ù	249	ü	252
ŵ	373	ŷ	7811	ẁ	7809	ẃ	7813
ÿ	375	ý	253	ỳ	7923	ÿ	255

(Note that the codes shown above are decimal, not hex.
Some characters may not print on some printers.)

Scottish Gaelic requires only the grave accent, which can occur on any vowel. The required character set is therefore a subset of that required for Welsh.

To cover all Gaelic variations (including Irish and Manx) a wider character set is required. I do not have an "official" list of required characters, but according to <http://www.smo.uhi.ac.uk/~oduibhin/mearchlar/fonts.htm>, this includes all the vowels (upper and lower case) with acute and grave accents and, for Manx, both upper and lower "c" with a cedilla. Certain consonants (b, c, d, f, g, m, p, s, t) can also have a dot over them. However, a digraph representation, using the consonant followed by an "h" (bh etc.) is also permitted. Unicode supports all but the dotted consonants in the Basic Latin and Latin-1 supplement code charts. The dotted consonants require the Latin Extended-A (c-dot, g-dot) and Latin Extended Additional (remaining dotted consonants) code charts.

A.5 Options for Language and Character Set Support

There are five main options:

- support only characters in the ASCII character set
- support the characters required for UK languages
- support other Latin-based characters
- support characters for other European languages, such as Greek and Cyrillic characters
- support all characters

Any solution will require work on some systems to ensure compatibility.

One might question whether all characters in all languages need to be supported to meet expectations for personal data. However, people having non Latin-based

characters in their names and addresses usually use Latin equivalents when dealing with the public sector. If they do not, the ISO translations are available. Additionally, supporting languages that do not read from left to right and top to bottom would cause further difficulties in implementation. Neither the MIAP participants nor the wider public sector shows much desire to support non Latin-based characters, either European or others such as those used in the Asian or Arabic countries.

If MIAP supported only those characters in the ASCII character set, those participants currently supporting accented characters would need some form of down-translation. Welsh and Gaelic characters could not be supported as not all of these can be translated to non-accented characters.

MIAP could comply with the Welsh Language Act and the Gaelic Language Act (Scotland) by adding a relatively few additional characters. However, this is likely to be overly restrictive, and is no easier in terms of migration for those currently supporting ASCII only. In particular, it does not provide support for the names of people or institutions with other Latin-based European characters in their names.

The preferred option is therefore to support all Latin-based characters, using ISO 9 and ISO 843 to translate from Cyrillic and Greek characters if required in the future (there is no known current requirement).

A.6 Conclusions and Implications

Based on expectations and increasing public sector practice, the first conclusion is that the general policy should be to support all Latin-based characters for names, addresses and general text fields, but not non-Latin characters.

Beyond this, there is a decision to be made in terms of how far to limit the supported character set. If any accented Latin characters are to be supported, there is unlikely to be any technical impact in supporting them all. However, as mentioned previously, the more characters that are supported, the higher the probability of errors going undetected. Against this, past experience has shown that, whenever a character set is restricted, there will be a requirement at some future time for characters that have been omitted.

On this basis, my recommendation is to provide support for all Unicode code charts for Latin characters, but to build the schemas in such a way that an individual project can further restrict the set if required. This provides support for all Latin-based languages and other languages through translations.

The charts to be supported are: Basic Latin (excluding the C0 control characters), Latin-1 (excluding the C1 control characters), Latin Extended A, Latin Extended B and Latin Extended Additional. This set corresponds to Unicode code points U+0020 to U+007F and U+00A0 to U+024F.

For some people, this raises a migration issue. Currently, some systems used by MIAP participants store non-ASCII characters and some do not. As long as this is the case, there will be a problem transferring data between them. Downgrading to the lowest common denominator (ASCII) is not an option for reasons described above. The onus is therefore on those not currently supporting a wider character set.

The best solution is clearly to upgrade to full Unicode support, and for many, that is not a problem since their databases have the capability. However, this could have knock-on effects that would need to be investigated by each organisation.

Anyone without Unicode support could front-end their systems to translate non-ASCII characters to ASCII in a reversible way. For example, they could use something akin to the Microsoft Word typing convention, representing, for example, "é" as "\e". The backslash character itself would then need to be escaped as "\\". Alternatively, XML character references could be used. These represent characters using their Unicode code point, so "é" would be represented as "é" (as a decimal code) or "é" (as a hex code). Again, the ampersand character would need to be escaped. This solution gives full Unicode support from systems without native support, but involves programming work and additional processing.

The final option is to translate non-ASCII characters to ASCII near equivalents. For example, "é" would be translated to "e". This is the least desirable option and should not be used when the stored data might later be transferred to another MIAP partner.

Before any system changes are made, it is important to review the capabilities of the existing systems, since many will support the recommended character codes without significant change.