Sigma Systems, Inc.

Student Identity Defined:
A Comparison of the Data Elements of Four Higher-Education Standards

Paul Heald
Sigma Systems, Inc.
August 2, 2010

Technical Brief

Electronic Student Data Exchange: Student Identity

Preface

Foreword

Areas of Communality

Identifier

Date of Birth

Race and Ethnicity

Gender

Name

Postal Address

Telephone Number

Email Address

Residency

Other Systems: NIEM

Other Systems: HR-XML

Other Systems: MedBiquitous

Appendices

Appendix A: Larger Breakdown of the Four Educational Standards

Appendix B: Proposed Schema for the Identifier

Appendix C: Proposed Schema for the Date of Birth

Appendix D: Proposed Schema for the Name

Appendix E: Proposed Schema for the Address

Appendix F: Proposed Schema for the Electronic Contact Information

Appendix G: Proposed Overall Identity Schema

Appendix F: Proposed XML Schema

Preface

Last year the U.S. Department of Education Federal Student Aid (FSA) shared with our community their plans to move to a service-oriented architecture. This will be extended to participants in Title IV financial aid to provide fully protected real-time exchange of data. A similar requirement is emerging from the National Center of Education Statistics (NCES) and their $500 million funding of state data systems that must exchange student data with colleges and universities.

From its very beginning, the Kuali Student Systems Project's goal was creating a core student system framework and enabling an industry wide service-oriented architecture; first implemented as a set of interfaces between modules and then Web Services. Our experience at the University of California, Berkeley suggests many universities will adopt Kuali software products and/or its standards. Many other third party software systems will be designed or redesigned to integrate with Kuali software, provide service oriented alternatives or use the Kuali standards in the higher education space.

There is a new requirement emerging from the integration of learning systems and student information systems that utilizes a similar architecture. This fall a similar network will be piloted for the exchange of data, including financial aid data, with foreign universities.

We are following these initiatives to ensure that our own software development is consistent with them. Our software will be easily integrated with emerging higher education software. We are also participating in the harmonization efforts by the standards setting bodies. All of us are encouraging commonality where there is functional overlap. The goal is a single standard data transport network to serve higher education in the U.S.

We believe Paul Heald's work should be shared with the information technology communities as a contribution to harmonization and the design of integration.

Randy Timmons

Sigma Systems Inc.

Denver, Colorado

Foreword

Many different standards exist to track student data through the computer systems that are used to trace student achievement in the institutions of education around the country and the world. Many commercial student management systems (such as BlackBoard, Oracle, Sungard and Datatel) use proprietary structures to record information about students. Open-source offerings such as Kuali publish their standards, but don't necessarily provide easy access to the data[1]. Other published standards, such as the SIFAssociation's(Schools Interoperability Framework Association)Standard specifically attempt to provide a standard for information interchange about students[2].SIFA is especially important because the U.S. Department of Education is providing more than US$ 250 million to develop SIFA compliant-software for state systems.

With more and more schools using information systems to track student data, and with students in most industrialized nations attending at least two educational institutions (and often, far more) over the course of their educational career, the benefits of transferrable student data become obvious. Internal systems could rely on a standard method to communicate about student, allowing the registrar's system to communicate with the library system and the financial aid system using a standard vocabulary. Externally, students transfer frequently between institutions or enroll at more than one institution.[3] This could be as simple as a student transferring from the local elementary school to the local high school, or from a college in the United States to a college in Europe for a semester. Students also transfer credits from other universities, either through previous studies, through the use of consortium agreements, or through the use of online learning systems allowing them to gain educational credits from other institutions worldwide. If student data could be transferred safely and securely from one institution to another, the benefits, in terms of reducing data re-entry and the ensuing corrections is huge, not to mention the greater quantity of data available to the institution, would be huge.

This is the initial report of a series of reports that will look at best practices for many forms of student data. However, this report looks most specifically at the fundamental problem of exchanging student identity. Once student identity can be established, the process of exchanging other information about the student becomes much easier.

---

[1] Here Kuali refers to the various higher education administrative systems under development and supported by the Kual Foundation [http://www.kuali.org/].

[2] The Educause standard (EduPerson) has not been included for reference, as it is mostly used to authenticate a user on a LDAP setup. It is not designed for the transmission of personal information across a network. The sharable content object reference model (SCORM) is a format designed for exchanging course content between learning management systems. Although individual learning management systems may make use of identity information, the SCORM standard only processes two forms of identity information. One is the full name of the user, the other is a user ID, as assigned by the LMS. No other personal information is specified by SCORM. There also exists euroCRIS (European Current Research Information Systems)(2010a, 2010b, 2010c)is a standards body serving researchers in the European Research Area which has developed CERIF (Common European Research Information Format). The specifications include fully qualified precise XML schemas.

[3] Simultaneous or overlapping enrollment in two or more institutions may present U.S. colleges and universities with the need for real-time data to remain compliant with U.S. federal financial aid regulations.

The Standards Compared

This report breaks down many of the data elements defined by four different standards or systems: the Postsecondary Education Standards Council (PESC) the School's Interoperability Framework (SIF), the U.S. Department of Education (ED) and Kuali. The aim is to look for communalities and seek problems that would inhibit data exchange between different educational institutions. These particular standards have been chosen as they represent four of the most current data standards that relate specifically to educational standards. Later in the paper, four standards not directly related to general education (NIEM, HR-XML and MedBiquitous are described)

The Postsecondary Education Standards Council (PESC)

PESC was founded in 1997, and aims to "produce standards to bring about cost-effective connectivity between data systems". The association specifically cites the need for educational data to be transferrable from the moment at which a student enters high school through "the successful completion of the educational experience"[4]. Although PESC has a number of data standards,this analysis specifically concentrates on their XML Transcript specification.

There is an agreement between PESC and Federal Student Aid, an organization within the U.S. Department of Education, to have common data elements. This agreement does not apply to data transport since the Department does not yet have real-time transaction systems[5].

The School's Interoperability Framework (SIF)

The SIF Association concentrates on interoperability between the K-12 schools, and uses many of the Department of Education's definitions[6] for its statistical tracking. SIF standards are used to allow schools within a zone to communicate with one another through a "zone integration server". Often, this means that the standard is used to collate data within a school district. Oklahoma legislates the use of SIF within educations systems in that state.

Because may school districts and their schools have installed SIFA-compliant software, SIFA compliance becomes important to the exchange of student data to support Longitudinal Student Data Systems with states[7].

---

[4] Cited from the PESC website, at www.pesc.org

[5] PESC Postsecondary Electronics Standards Council. (2002:26 February). Agreement on the Use of Standards in Electronic Exchanges Support Federal Financial Aid Between the US Department of Education's Office of Student Financial Assistance and the Postsecondary Electronic Standards Council. Washington DC USA: Postsecondary Electronics Standards Council. www.immagic.com/eLibrary/ARCHIVES/GENERAL/PESC/P020226.pdf

[6] The National Center for Educational Statistics (NCES) has specifications for tracking many statistical aspects educationally-related data. Data elements in these specifications may not be identical to those of Federal Student Aid, an organization within the U.S. Department of Education.

[7] Data Quality Campaign (2006: 7 November). Creating a Longitudinal Data System: Using Data To Improve Student Achievement. Washington DC USA: Data Quality Campaign. Available from www.immagic.com/eLibrary/ARCHIVES/GENERAL/DATAQ_US/D061107C.pdf

The U.S. Department of Education (NCES)

The Department of Education defines a Student "object", which is used as the model by which the federal agency collects data about students. This model is not designed specifically for any form of student transfer, rather as a model to allow statistics to be compiled.

Kuali

The Kuali project is a community of institutions that have come together to produce an open-source administrative suite for higher education. The middleware component, Kuali Rice, includes an identity management system called KIM (Kuali Identity Management). Rice allows different areas of the Kuali system to communicate with each other, sharing information as appropriate for the application. At the present time, Kuali has no feature to export student data outside of Kuali compliant applications. The comparison of student identity was derived from the database model underlying Kuali[8].

Areas of Communality

The following table attempts to compare those parts of the data that are common to three or more of the four standards being contrasted. Nine common sets of information were found between the standards, comprising identifiers, date of birth of the individual, the race or ethnicity of the individual, gender, name, postal address(es) telephone number(s), email address(es) and residency indicators. A more in-depth look at these standards, showing areas of non-concurrence can be found in Appendix A.

[8] The data model can be located at http://rice.kuali.org/documentation/1.0.1.1/db_diagrams/RiceKRIM1.0.pdf

Electronic Student Data Exchange: Student Identity

Type of Data Stored	PESC	SIF	ED	Kuali
Identifier	A range of allowable options, based on numbers issued in Canada and the US, such as the US Social Security Number, etc.	A range of allowable types of ID, specific to the area of implementation.	Enumerated identification numbers as listed in NCES0140	Allows for a range of identity indicators which are locally defined in the Kuali database
Date of Birth	Stored as yyyy-mm-dd Also allows a birthday field (mm-dd)	Stored as yyyy-mm-dd and also includes birthdate verification	Stored as yyyy-mm-dd and also includes birthdate verification	DateTime field
Race/Ethnicity	Not Stored	Enumerated race types per ED Hispanic or Latino Flag Multiple ethnicities with percentage assignments possible.	Enumerated race types Hispanic or Latino Flag Multiple ethnicities with percentage assignments possible. Large quantity of other migrant data	Ethnicity defined in the database. Only a single ethnicity can be stored
Gender	Male, female, unreported	Male, female, unreported	01- Female, 02- Male	Gender defined by ISO5218
Name [in addition to First, Middle, Last] All systems allow alternate names to be stored.[9]	Prefix, Suffix, Composite Name	Preferred Name, Prefix, Type of Name	Alias, Nickname, Generation code, Composite Name, Type, Former legal name, Last name at birth, Tribal or Clan name	Title, Suffix, Type
Postal Address	Allows for Domestic and International addresses Core of Line1..3, City Domestic has state and zip International has province, postcode and country code.	Freeform and atomic address areas conforming to US standard address. Global x, y coordinates.	Freeform and atomic address areas conforming to US standard address with added country code and enumerated address type.	Lines 1..3, City, State, Postal Code, Country Internally Defined Address Type
Telephone Number(s)	Country, Area, Number, Extension	Freeform number, extension, type and listed status	Freeform number, extension, type and listed status	Country, Number, Extension, Type
Email Address	Freeform	Freeform address with enumerated type.	Freeform address with enumerated type.	Freeform address with internally defined type.
Residency Indicator	State, territory or country code, with status of resident, nonresident or unreported.	List of countries of residency based on ISO31661/ISO31663 country name[10]	Residency status with respect to the school district is flagged	In-state residency is flagged

[9] To accommodate complex names—especially Arabic names—some specifications using First, Other, and Last name. The only difference would be the length of the representation. For an excellent discussion of name storage, see *Whose Name Is It: Names, Ownership and Databases.*

[10] ISO 3166-1 refers to the two character country codes available for with both English and French country names. The ISO codes are published by the Maintenance Agency (http://www.iso.org/iso/country_codes/iso_3166_code_lists.htm). "ISO 3166-2 contains a complete breakdown into a relevant level of administrative subdivisions of all countries listed in ISO 3166-1. The code elements used consist of the alpha-2 code element from ISO 3166-1 followed by a separator and a further string of up to three alphanumeric characters." [www.iso.org/iso/country_codes/background_on_iso_3166/iso_3166-2.htm#relationship]

Initial Points of Note

During the breakdown of each of these standards, it became very obvious that each of them has been designed with a limitednumber of purposes in mind. This is less surprising with the Department of Education (ED) and Kuali data sets, as neither of these are specifically designed to allow system interoperability. Both the PESC and SIF standards are designed from the perspective of allowing student records to be transferred between different institutions.

Identifier

In most electronic record systems, each individual can be recognized by some form of unique identification key. In some systems, this identifier is commonly known to the user and is even used by the person. This is the case in many universities, where your student number or student id can be used to access online records, and may be required when submitting papers, exams, or official correspondence. In other systems, more often smaller systems, the unique id for each person in the system is used only as an internal way to tie records together. A third form of identification could be a number produced by an external issuing organization, known to be unique, and used often as a means to identify a person. Examples include the US Social Security number (SSN)[11], the Swedish Personnummer, or the serial number of an identifying document such as a passport or driver's license.

The use of such identifiers is essential for reasons of database integrity and speed, and in a proposed system where records can be moved electronically, there needs to be a way to disambiguate records coming from different institutions and correctly associate them with the correct people within the system.

However, the choice of identifiers is not an easy one. It is common for educational and other institutions in the U.S. to use the Social Security number as the "catch-all" identifier for people. However, this suffers from several problems. The first, and sometimes least thought about problem is that not everyone has a U.S. social security number. If a system is being designed to be able to transfer student records between institutions, it seems short sighted to insist on a standard that only U.S. citizens and residents can use. Secondly, most educational institutions cannot require a person to disclose their social security number (though private institutions can refuse to accept a student who does not disclose this number). Thirdly, the storage of the social security number along with other personally identifying information makes such records a target for identity thieves[12]. As such, the use of the social security number as an identifier has been discouraged by the U.S. Department of Education. There has been a notable trend away from the use of social security numbers in recent years by educational institutions in the U.S. (many opting to issue their own identification numbers to replace them); thoughthis does not necessarily mean that those

[11] As most identitfying documents in the U.S. are issued by the states, the Social Security number has become the de facto identification number used within the United States. Designed as a tax reference number, the number is now issued to most U.S. Citizens when they are babies, and is required in many different situations. Though few institutions are able to require a user to give their number to them, many private insitutions will refuse to give service to users who will not release their SSN.
[12] The use of social security numbers as identifiers in the United States is problematic. Certain states prohibit their use as a primary means of student identification. See Arizona ARS 15-1823, 44-1376.

institutions are not storing social security numbers, or using them to tie together different forms of records.

For all their associated problems, however, identifiers such as the social security number have the utility of generally being known by the user, and are therefore very useful when verifying that records coming from another system do, in fact, belong to the user.

Across all of the different systems, a number of different identifiers are allowed. The PESC standard, for example, defines the use of the SIN (Canadian Social Insurance Number), NSN (Canadian National Student Number), SSN (The U.S. Social Security Number) as well as identifiers assigned by the sending and receiving institution. These latter numbers make sense in a system designed to transfer student information from one system to another, and are the recommended means of identification with the PESC standard. The problem with the use of these numbers is that they require a prior negotiation to identify the student which is defined outside of the PESC standard. Ideally, it should be possible to take student records from different sources and be able to electronically match them together, with little or no user intervention.

Date of Birth

The date-of-birth field is almost entirely standardized across the systems. All of the systems (except Kuali, which does not mandate the storage format other than defining it as a datetime field) adopt yyyy-mm-dd as the format for storing the date of birth. The four number year ensures that the system should be valid for nearly 8,000 years, and storing the different parts of the date individually make the data easier to extract and use. SIF also allows for an optional “birthday” field, which does not include the year. ED and PESC specify a means of storing how the birth date was verified. This code relates to a list of documents deemed as acceptable means of establishing dates of birth as defined by the U.S. department of education.

SIF’s birthday field poses problems as it breaks normal form when the information is stored within a database. Such a field can easily be derived from the full birthdate. The use of NCES standards for birthdate validation makes the system more difficult to use internationally, as other countries are not likely to use the same document enumeration. The lack of any form of digital signatures relating to the verification also poses problems for an institution relying on the verification of another institution. As such, its transmission between institutions seems less than useful.

Race and Ethnicity

The race and ethnicity of the student is useful for statistical report keeping. It is absent from the PESC Standard, which probably reflects the way in which the data is to be used (transmission from one institution to another). The biggest problem with race from this perspective is that the classifications of race and ethnicity are often particular to the recording country, and are thus difficult to encapsulate. For example, Native American races are often of great statistical importance in American countries, but rather less useful in comparisons in a European country. Equally, Basque ethnicity may be an important identity to record within a Spanish institution, but less so in an Australian institution. Moreover, not only are such classifications geographically dependent, some classifications are temporally dependent. In the last century, U.S. authorities

would record“Irish” as an ethnicity, whereas today, the demarking of Irish ethnicity as distinct from other Europeans has given way to the demarking of Hispanic and Latino ethnicities.

Although race is an important part of identity from a social and political perspective, its use as a marker of identity here presents some important problems within an electronic system which can span regional boundaries.

Gender

The seemingly simple question of gender is dealt with in similar ways across the four systems. The only real variation is the means of encoding gender. A difference that does exist is whether or not the system allows the gender to be unreported or not, and under the ISO Standard, as used by Kuali, there exists the possibility of using both unknown and unreported codes, in addition to male and female.

Gender, however, like race and ethnicity can be problematic. As an educational record can last almost from the time of birth to the time of death, it should be acknowledgedthat gender might not remain a singular value during that time. How different countries, and indeed different jurisdictions within the U.S. define gender is also not uniform (some states only recognize the biological gender of the person at birth, other states recognize the gender of the person as that person self identifies (and will change birth records accordingly), yet otherstates will only recognize changes in gender if surgical interventions have been performed etc. There are also differences in the way that states and countries will and won’t alter documentation relating to such changes, making the issue of gender rather more complicated that it might first appear.

As such, the utility and implementation of a gender marker in this document requires more analysis than it might first appear.As with race and ethnicity, it seems logical to move gender outside of the identification of the individual.

Names

A person’s name is their primary means of identification in human discourse, and in order to render a system that both meets the needs of the computer systems that process these documents, as well as the needs of the people that use them, the name of the person as stored in the record needs to be looked at very carefully.

On initial inspection, one would assume that the question of storing names would be an easy one. It might be assumed that as people both have, and know their names, matching them in a database would be a relatively easy task. However, like many of these elements, problems arise when trying to define standards that can be used internationally.

All of the systems more or less store the name in the same format. This, however, leads to a number of problems. All of the systems assume that all names can be stored following the “American” format of First Name, Middle Name, Last Name, with title and generation information. While this works for a great number of names, and many other names can be changed to fit this system, it does not provide a framework to accurately record name information. Notably, Arabic names cannot easily be manipulated to follow this format. Moreover, when many Asian names are

forced to follow this format, name-matching systems that are not aware of the cultural weight given to specific parts of the name fail to be efficient. Studies have shown that poor name information in databases makes retrieval of records particularly difficult[13]. Without a flexible system that can accurately record name information, it is very difficult to match names to people, which compounds the problem of identifiers mentioned in the earlier section. Some systems allow for a non-atomic freeform field in which the name can be stored, however, this compromise violates normal form, and may lead to update problems in any database which relies upon it.

Postal Addresses

All four systems define the address as a number of address lines, with atomic fields for city, state, zip/postal code, and some allow the country code to be added as well. The PESC standard makes an attempt to bring together both US and international addresses, but falls short by only defining a system that will allow Canadian addresses to be entered with any ease. Other systems allow "international" addresses to be entered as freeform text.

In order for these records to be useful and operate for a greater number of people, standards have to be devised that are not specific to a single area or purpose. If this data is to be leveraged, not only as a means of tracing student achievement for the individual, but also as a means to track statistical information, such definitions are unacceptable.

While there is a great deal of communality between different countries' postal addressing systems, there is no single standard. Most post offices will deliver mail addressed incorrectly, especially if the mail is coming from an international destination. However, it is not acceptable to simply make the question of postal addressing a problem for the post office. This specific area requires serious analysis. Moreover, poor storage of the postal information makes searching the data less effective.

Telephone Numbers

Thanks to a fairly standardized telephone number formatting scheme implemented by the ITU[14], the storage of telephone numbers is one of the easier fields in databases. Most of the systems allow for the storage of the telephone number, area code, local number and country code, along with extension numbers if needed. Areas of non standardization include the way that different telephone numbers are enumerated and categorized.

Email Addresses

Due to their already international nature, email addresses are stored in most systems in a very similar way. It's worth noting that the length of the field is important, as some institutions have had problems with truncating email addresses when the storage field was too short. Given that the

[13] See "Whose Name is it? Names, Ownership and Databases" by Kerry Dematteis, Richard Lutz and Heather McCallum-Bayliss.

[14] International Telecommunications Union, www.itu.int

maximum length of email addresses is fixed, this problem should not exist[15]. The way that enumerated email addresses are stored is not standard across the systems.

Residency Indicator

As with gender, race and ethnicity, the use of residency in a student record requires that residency be defined in terms as required by the locality using that information. Doubtless, the need to know if a student is resident in a particular area is useful for administrative purposes in a school environment, for example, deciding on tuition rates for certain colleges. However, there is no clear definition for residency even within the U.S.[16] Moreover, residency may change over time. All of the systems make some attempt to trace the residency of the student. None address the issues posed by the recording of such information.

[15] RF5321 defining the SMTP limits the length of the email address to 254 characters. Although technically an email address can be 320 characters long according to other specifications, it will be rejected by the SMTP.

[16] Each state defines its own definitions of residency, which may be different relating to payment of taxes, vehicle registration, and residency for issues such as 'in-state' tuition payments for student. The federal government defines the rules for residency affecting federal taxation and immigration status. It should be noted that residency status may be different, depending on the matter in question. A foreign worker, for example, may be resident for tax purposes, but non-resident for immigration purposes.

Other Systems

In addition to the four educational systems compared, the analysis includes several other sources of computer-based identity models. Those models are presented here.

NIEM (Version 2.0)

The national information-exchange model is a framework for the exchange of information between federal agencies who participate. The framework originated in the Department of Justice, and the Department of Education does not participate. Many of the data elements recorded in the core of the schema (where the PersonType is defined) are not relevant to education, therefore This analysis is limited o only those elements where there is a crossover with the elements studied in the other systems.

Identifier

NIEM allows for an array of identifiers, notably the PersonLicense (which can be used to store any form of government issued license, which includes other attributes, such as issuing jurisdiction, effective and expiration dates, etc. The system also allows for Passport information to be stored in a similar format. This method of storing several identifiers is similar to some of the educational systems. What is useful to note is the ability to store different identifiers from different agencies while retaining their unique nature.

Date of Birth

In keeping with the judicial roots of NIEM, the date of birth of an individual can be stored in several formats, with an assertion of the accuracy of the date of birth. Many of the NIEM attributes seem excessive for the needsof educational records, given that a person enrolling in an educational institution would rarely have a reason to lie about their date of birth. For the sake of easy comparison, the use of a single method to record date of birth is suggested for educational records.

Race and Ethnicity

PersonEthnicity along with a number of other personal characteristics are defined in the specification. Again, the judicial roots of the system are apparent here, and many of the recorded attributes (such as eye color, length of hair etc.) are unnecessary for the purposes of identity in an educational system.

Gender

A simple PersonSex element is used to define the gender of the person.

Name

Somewhat surprisingly, the complex name element follows only the U.S. name standard (prefix, given, middle, surname, suffix, maiden name and a full name field). The ability to store and match names that follow different formats seems surprising, especially given the law enforcement background of the standard. The addition of a maiden name field seems superfluous as alternate

names can be listed. Whether or not the maiden name field can be used for other previous names is not clear from the standard.

Postal Addresses

Postal addresses, following U.S. conventions can be stored in a structured format, as can any form of address in an unstructured format. It is perhaps less surprising that addresses are stored in U.S. format natively as the roots of the system are to track people within the United States.

Telephone Numbers

NIEM allows for multiple telephone numbers, stored in both North American (NANP [North American Numbering Plan]) and international formats.

Email Addresses

Email addresses are stored as a freeform string, with enumerated types[17].

Residency Indicators

Both a simple flag and a textual description of residency are stored. The system also has an immigration component which give much more specific details relating to residency, but this module is beyond the scope of this comparison.

HR-XML (Version 3.0)

The HR-XML standard is used to exchange information between human resources systems. It should be noted that HR-XML attempts to encapsulate education information from the perspective of a potential employer.

Identifier

The schema provides areas to record any number of enumerated personal identifiers, including validity dates for the type of document. The enumeration is extendible, allowing for other forms of identifiers that are not specific to a country.

Date of Birth

The date of birth is stored as a flexible date/time field, allowing the storage of both the birth date and the time. Likely, only the date is stored in most instances.

Race and Ethnicity

Both race and ethnicity are stored in the personal data section of the schema, and a system of codes is used to record the specifics of both race and ethnicity. The documentation states that this may be different by the needs of the country, noting that Department of Labor rules in the U.S. require the recording of Hispanic or Latino ethnicity, in addition to race information.

[17] Enumerated is used by programmers to mean a fixed list of types. For example [personal, work, other]

Gender

Gender is stored in a fashion similar to ISO5218, allowing male, female, unreported and unknown.

Name

Although a stated intention of the PersonName field of HR-XML is to "be useful across many cultural contexts", the details do not go far beyond any of the other standards. While allowing for storage of the name in different scripts, as well as providing many different options for names (Given Name, Preferred Name, Former Family Name, etc.) the schema still basically adopts the First/ Middle/ Last/ Generation pattern common in the United States. While going rather further to give a name record that can be used for names that do not follow the U.S. pattern, the system still does not fully address the need for a flexible naming element that can represent names from different cultures.

Postal Address

By far the most agnostic address field, the standard allows for a great variety of different countries' addresses to be stored without changing the format of the field. Each field of the address is stored atomically, and it is left to the HR information system to format the address correctly if mail is produced. The standard defines CountryCode (following ISO 3166-1), PostalCode, Region, Municipality, DeliveryAddress (composed of AddressLine, StreetName, BuildingNumber, Unit and PostOfficeBox fields), Recipient and Organization fields. In addition, there is a data element describing the type of address (PO Box, street address, military address).

Telephone Number

As with most of the other standards, the telephone field is divided into the country code, area code, number and extension. It also allows for an access code.

Email Address

The schema provides for a URI [Uniform Resource Idenitifer][18] to be added to the communication block, hence an email address in the format mailto:foo@bar.com would be valid here. The schema also provides the option to record the email preference (HTML of plain text) of the recipient. Each block of communication data (address, telephone number and email address) can also be categorized as either personal or business.

Residency Indicators

In addition to storing the CitizenshipCountry, the ResidentCountry can also be stored. This is stored as a country code, with valid from and to dates. The system does not allow for the representation of residency within a territory, for example, residency within a state.

[18] Many people are acquainted with the term URL which is used in reference to web addresses (http://www.sigmasys.com) a URI is a more general term for addressing resources on the Internet, in addition to web pages. Email addresses can be written as URIs simply by placing mailto: before them, as shown in the example given in the text.

MedBiquitous (v1.0)

Founded by Johns Hopkins Medicine and leading professional medical societies, MedBiquitous is a not-for-profit, international group of professional medical and healthcare associations, universities, commercial, and governmental organizations dedicated to advancing healthcare education through technology standards that promote professional competence, collaboration, and better patient care[19].

Identifier

A UniqueID field is allowed by the standard. More than one may be defined for each member. It is bound by a domain which states the issuing organization. Given the medial nature of this standard, it is assumed that the domain would most likely be the name of the institution that issued the credentials. While this permits different identifiers, it does not solve the question of unique references, nor does it have any way of standardizing the name of the issuing organization. It would be possible for one version of a record to show an identifier from "Georgetown University" whereas another might be from "Georgetown". There is also the ability to store the "Tax Number" in the record. As with the UniqueID, there is an attribute describing the type of tax number. However, there is no standard enumeration of the types of numbers, meaning that one rendering might have the attribute "U.S. Social Security Number" whereas another might have the attribute "SSN".

Date of Birth

The date of birth is stored in a simple Date datatype. The date can also be attributed with an accuracy flag. This defines which parts of the date of birth can be considered accurate. Values can be DayMonthYear, MonthYear or Year. There is also a restriction attribute, stating whether or not this data may be published. This is common to several elements within the standard. The standard states that institutions must have their own rules pertaining to the sharing of limited information.

Race and Ethnicity

Both race and ethnicity may be defined as simple strings. This makes the system more flexible, as it does not proscribe the categories of race or ethnicity that might be defined, however, it does make exploitation of the data by an electronic system somewhat more difficult.

Gender

Gender of the individual is has the allowed values of male or female.

Name

The name schema is defined in an external schema, and is designed for reuse. It allows for title, given name(s), family names, generational identifiers, degrees, aliases, and former names. It also

[19] www.medbiq.org/about_us/mission/index.html

allows for a formatted name field, which attempts to account for “cultural issues regarding [name] order”.

Postal Address

As with the name schema, the address schema is externally defined and is designed for reuse and importation into other schema.

The address is stored in a format that is mostly agnostic to the type of address being stored, allowing easier storage of foreign addresses. The street address lines are non-atomic forcing several data elements into a single container in certain cases (for example, street number, street and road category would all be stored in a single data element). It also allows for organizational names to be added to the address field, and a restriction category is used to decide if an address may be published or not.

Telephone Number

Numbers are stored as a combination of country code, telephone number, extension and a freeform description (such as Fax, Home, etc.) There is also a restriction attribute.

Email Address

Email addresses are stored as a freeform string conforming to IETF RFC 2822[20]. They also have a restriction attribute.

Residency Indicator

Residency is defined as a simple 2-digit country code. Residency within a territory is not possible.

[20] The Internet Engineering Task Force publishes technical documents to improve the way people use the Internet. 2822 defines the Internet messaging format.

Problems and Potential Solutions

None of the current standards are adequate for transferring identity information which is a crucial first step in facilitating the exchange of electronic records. A strong standard will pull together the strengths of the different systems, while reducing the problems inherent in them.

Identifier

In a globally-integrating world, the use of the U.S. social security number as a primary identifier is not an acceptable way of providing a unique identifier for users. None of the educational standards successfully address the problem of uniquely identifying an individual in a way that is both secure and gives a very high level of uniqueness.

Most of the systems do acknowledge the need to be able to record multiple forms of identifiers. Whether it be a UK national insurance number, a university issued ID or a Canadian student number, most of the systems allow some flexibility in the storage of different identifiers, and most allow for the storage of more than one identifier.

The availability of multiple unique identifiers brings up the interesting question of "how do you identify the identifier?" It is not enough to state that an identifier is a SSN, as the United States is not the only country to have a social security number. The rather more verbose system preferred by MedBiquitous is a better solution, but suffers from the lack of standardization of the types of identifiers to be stored.

The RS3G (Rome Student Systems Group) mobility pilot developed at the University of Warsaw anticipated the ambiguity of student identifiers. Student identifier in Europe are often issued nationally. In his design Rafal Nagrodzki[21] prefixed the identifier with the two-character ISO 3166-2 country code. A similar problem will arise in the U.S. when identifiers are issued by the states. In this case the prefix of the ISO 3166-2 specification. "ISO 3166-2 contains a complete breakdown into a relevant level of administrative subdivisions of all countries listed in ISO 3166-1. The code elements used consist of the alpha-2 code element from ISO 3166-1 followed by a separator and a further string of up to three alphanumeric characters e. g. DK-025 for the Danish county Roskilde, IT-MI for the Italian province of Milano, [and] MG-T for the Antananarivo province in Madagascar."[22] While starting to address the question of multiple identifiers, this system relies on each nation, or subdivision only issuing a single type of identifier, which is often not the case[23].

Of all the standards, few recognize the expiration date of certain identifiers. This is perhaps unsurprising, as most of the systems are designed to only accept those forms of identification which will not change over time (such as professional registration numbers, National student numbers, etc.) In order to improve document matching, the ability to specify validity dates is useful, especially if those identifiers are reused by the issuing institution.

---

[21] See "The Mobility Project"

[22] More information can be found at http://www.iso.org/iso/country_codes/background_on_iso_3166/iso_3166-2.htm

[23] The standard proposed in the mobility pilot used elements derived from SCHAC (SCHema for ACademia) which provides for enumerations of identity documentation which can be stored on external servers. We try to propose a solution that relies as little on external directory-type services as possible.

Validity dates are also very important in the case of some very common documents. In many countries, the serial of a passport changes every time the document is reissued. Some countries also change the driver's license number upon reissue, whereas others continue to use the same number once issued. Within the U.S. the situation is inconsistent when a person moves between the states. Sometimes the receiving state will duplicate the number from the original driver's license, other times, a new identifier will be issued. This situation is further complicated when a person moves again back to their original state, which may reissue a new number, retrieve the old number, or duplicate the number from the other state. Needless to say, without the noted validity dates, it could be very difficult to match serial numbers in certain situations.

One way to solve the identifier problems is to recognize the two different forms of identifiers that exist: commonplace government identifiers, and institution specific identifiers.

Government-issued identifiers

Many countries issue some or all of the following types of numbers: driver's license numbers (DL), taxation identifiers (TAX), national identification numbers (ID), birth certificates (BC) and passport numbers (P). It should be noticed that not all of these documents can be used in all cases. For example, not all birth certificates carry a serial number, or the serial number on the document cannot be assumed to be unique within its category

In order to make these numbers unique for the purposes of automated recognition, the following attributes would need to be known:

- Type of identifier (DL, TAX, ID, BC, P)
- Issuing Country
- Issuing Region (for example, in the case of a U.S. driver's license) (optional in most cases)
- Validity from (optional)
- Validity to (optional)
- Versioning information
- Serial Number of the Document (the identifier itself)

Note that in the proposed schema, the issuing country and region elements were merged into a single element, allowing for the use of either an ISO 3166-1 or ISO 3166-2 code.

Institutional Identifiers

Institutional identifiers essentially those identifiers that do not fall into the categories above,are those issued by schools and universities, school districts, or government bodies that do not fall into the above identification categories.

One simple way to make a unique identifier here is to leverage the already existing unique database of identifiers- the institution's URL. This method does mean than any institution that does

not have a URL would not be able to provide a uniquely identifying number in the identity record, however, as more and more institutions gain a web presence, this seems the best way to enforce uniqueness without increasing the reporting burden on any institution, or having to create a separate identification service.

In order for this type of identifier to be unique, the following data elements would have to be recorded:

- Base URL of the issuing institution/ government body
- Validity from (optional)
- Validity to (optional)
- Versioning information
- Serial Number of the Document (the identifier itself)

Protection of personal information

In many cases, it is preferable not to transmit the actual serial number in order to reduce the value of the document to identity thieves, while still retaining the ability to match together documents in an automated fashion. This could be achieved by providing a hash of the serial number of the document, together with other pieces of data such as the family name of the individual in order to reduce the likelihood of reverse attacks.

In order to implement this, a type attribute could be added to the serial number, detailing the type of hashing algorithm used (for example, MD6). The serial number could then be the hash of a salt value followed by the serial number in question. In this way, an institution would be able to match together documents coming from two sources, so long as the hashing function were the same, or so long as either institution had access to the un-hashed version of the serial number. In order to promote inter-institution matching of serials, it would be necessary that the salt value be a pre-determined part of the identity, or that the unencrypted version of the number be available to at least one of the institutions.

The restricted tag, common in the Medbiquitous standard could also be employed here to restrict access to certain identifiers. For example, a Canadian social security could be stored in a database for use internally within a school, but be tagged as restricted, and therefore not allowed to be transmitted if records are transmitted between institutions.

While this method of storing identifiers is rather more complex than most of the systems thatwere analyzed, it allows for a great deal of flexibility, and is more likely to lead to a positive match between different records coming from disparate sources.

See Appendix B for a proposed XML schema for the identifier.

Date of Birth

The date of birth field is well defined across all of the standards. The XML format of yyyy-mm-dd is almost universally employed. The optional attributes of restricted and accuracy from the MedBiquitous standard are powerful additions. Both SIF and the ED standard store information relating to the verification of the document. While this is useful, it is U.S. specific, therefore it would be better if the verification code were coupled with a URL to identify the standards being applied. In this way, individual countries are free to set their own standards on what they consider acceptable documentation.

See Appendix C for a proposed XML schema for the date of birth record.

Race, Ethnicity and Gender

All three of these attributes are far more powerful from the point of view of statistics rather than identity. All three are very difficult to capture electronically (especially in the case of race and ethnicity) and legally (in the case of all three). As such, they should be avoided as part of the identification information of an individual. It is expected that institutions and regulatory bodies will still wish or require that this information be collected, however, it should more logically be moved to a separate part of the specification relating to biographical information. That block of information can be referenced to an individual country's requirements and definitions. It would likely be a restricted block of information, and would therefore not be transmitted between institutions that are in regions that use different classifications.

Name

The capturing of a person's name is important for a number of technical and social reasons. Names are the primary identifiers of people in human discourse, and on the surface, it does not seem that it should be difficult to encapsulate them in a computer-based format. Names, however, are very culturally dependent. Take, for example, a standard American name. Most people's names are composed of a first name, a middle name and a last name. Generally speaking, the middle name can be reduced to an initial because it is not given the same social weight as the first name. However, it would be uncommon to use an initial for the first name and keep the middle name written out in full, unless a person uses their middle name as their common form of address. How the system then deals with multiple middle names, hyphenated last names, etc. all further complicates this situation. Now, take for example a common Korean name; often Korean's have two first names, and they are of equal importance. Therefore, where Joseph Andrew Doe may equally be referred to Joseph A. Doe, Hae In Lim cannot be referred to as Hae I. Lim. The problem of the storage of names becomes even more complicated when other naming conventions, including other Asian names, Arabic names and Hispanic names.

Some of the systems attempt to gloss over these problems entirely storing only names that conform to a single format. Some attempt to give other options, such as storing the name in a non-

atomic[24] field, so as to permit freeform entry of names. The HR-XML standard takes some steps towards solving the problem, but does not address enough problems to be truly flexible.

In order to better capture different types of names, a format that is open to extension is the most appropriate route forwards. Each part of the name can be individually defined with an identifier, and certain attributes (such as whether or not it can be reduced to an initial, whether or not its order with respect to another name is relevant, whether or not it is hyphenated with a certain name, etc.) The format of certain forms of address can then be stored within the dataset itself, and the names can easily be built from the data. So, for example, the data might define an informal name, a formal name and a full name. This allows the system the flexibility to produce names with wildly varying formats, while at the same time, keeping the data format consistent, and keeping breaches of normal form to a minimum.

A proposed schema for the encapsulation of names is proposed in appendix D.

Postal Address

Of all the standards looked at, the HR-XML standard takes the most pragmatic approach to postal addresses. A comparison of the OECD[25] countries address formats, plus most middle-Eastern addresses shows that most addresses can be stored using this format with very little tweaking. A simple XSLT would allow the data to be extracted in the correct format to print an address label, while at the same time, providing atomic data that can easily be searched.

Appendix E contains a proposed schema for the postal address. This schema is based on information derived from online repositories of address formats, and takes into account the formats used in all of the OECD countries, except for Ireland, Korea and Israel, and also includes India, Pakistan, China, Jordan, the UAE and Hong Kong.

Telephone Number and Email Address

Due to their already internationalized formats, there is a large amount of consensus between the systems relating to the storage of both telephone numbers and email addresses. The main area of divergence relates to the ways in which these contact mechanisms are classified (home, personal, work, etc.) The use of the restricted attribute mimics that of the unlisted attribute in the ED specification.

Appendix F contains a proposed schema for electronic contact information.

[24] Storing something in atomic form means that the individual parts of it are stored in separate fields, so the first, middle and last names are stored as three different strings. To store information non-atomically means to store multiple piece of information in a single string, therefore a name might be stored as "John Andrew Doe". While storing information in this format removes some of the problems of how to store information, it does make leveraging information somewhat more difficult. As a very basic example, if we wanted to mail out a letter to a person in a database, it is much easier to produce a line that reads "Dear Mr. Doe" if the information is stored atomically compared to if the name is stored non-atomically. There are also many more technical reasons for avoiding non-atomic fields in databases, but this is beyond the scope of this document.

[25] Organization for Economic Cooperation and Development. The OECD was chosen for this comparison simply to limit the number of addresses to be compared, and to provide the most likely set of address data for a system of this type at this point in time.

Residency

Although important to funding in certain educational environments, the legal residency of an individual is not a useful part of a person's identity. Moreover, residency can sometimes be very difficult to define, and sometimes changes depending on the specific question that is being asked. As such, data pertaining to residency, citizenship, visa status etc. should be moved to a block of data that does not form part of the core identity service. In this way, it can more easily be linked to a specification that can be more pertinent to the legal framework in question.

Appendix A

Original Breakdown of Standards (unedited)

Electronic Student Data Exchange: Student Identity

	PESC Standard	SIF Standard	ED Standard	Kuali Standard
Information Source	*Taken from XML Standard Format for College Transcript 1.1.1A [STUDENT.PERSON]*	*Taken from Specification 2.3 StudentPersonal*	*Taken from Student Object (Early Childhood) description Element A – Personal Information*[26]	*Taken from RICE KualiIdentityManagement specification.*[27],[28]
Identifiers[29]	**School Assigned ID** **SIN** [Not recommended] **NSN**[Not recommended for US transfer] **Agency Assigned ID** [suggestion that states prefix with state code] **Recipient Assigned ID** **SSN** [Not recommended]	**RefID** **LocalID** **State/ProvinceID** **ElectronicIDList** **OtherIDList**	**Identification Number with enumerated types of number** [NCES0140]	**Internal object keys** **External_ID** which can be of numerous types, all defined in the type table of the database. These do not appear to be standardized.
Biographic and Demographic Data	**Date of Birth** [yyyy-mm-dd] **Birthday** [mm-dd] **Gender [**Male, Female, Unreported]	Demographic Breakdown includes the following: **Race list**: NCES0849 Race code, other code, proportion (decimal) **HispanicLatino** [yes/no] **Gender** [m/f/u] **Birthdate** [yyyy-mm-dd] **Birthdate Verification** [NCES031] **Place of Birth** [freeform] **County of Birth** [freeform] **StateofBirth** [US and CA state/province codeenumerations] **Country of Birth** [ISO31661/ISO31663 country name] **Country Arrival Date** [yyyy-mm-dd] **Citizenship Status** [based on NCES0322] **English Proficiency** [NCES0585 or other code] **Language List** [repeatable list composed of ISOZ3953 language codes, other code, NES0327 language type, freeform dialect string] **Dwelling Arrangements** [NCES0600 code, other code] **Marital Status** [NCES0330]	**Hispanic or Latino Ethnicity** [0149] **Race** [0150] **National/Ethnic Origin** [0160] **Sex** [01- Female, 02- Male] **Birthdate** [yyyy-mm-dd] **Birthdate** Verification [NCES0190] **Birth Place** [City, Country, State] **Citizenship Status** **Country of Citizenship** **Large quantity of migrant** data NCES0320-0362]	**Ethnicity** (defined as a code) Only one ethnicity can be specified. **Date of Birth** (datetime) **Gender Code** **Residency** (freeform) with references to determination method and if the student is defined as “in state”. Citizenship [postal country, “start and end dates”, status] **Visa Status** [mostly freeform]
Name	**Atomic Name Breakdown conforming to US naming conventions** Prefix	**Atomic Name Breakdown conforming to basic US naming convention** Prefix	**Extended Atomic Name Breakdown** First Name Middle Name	**Atomic name breakdown** First, Middle

[26] ED Standard specifically separates data into 9 areas, Personal information, Enrollment, School participation and activities, Non-school and post-school experience, Assessment, Transportation, Health conditions, Special program participation and student support services and Discipline. Breakdown only looks at area A.

[27] Description has been taken from a system-wide ERD, therefore certain aspects have been ignored to simplify the definition for this purpose.

[28] KIM keeps very close track of the last update of almost all of its entities, along with version numbers for each of the entities. These have not been recorded to simplify the table.

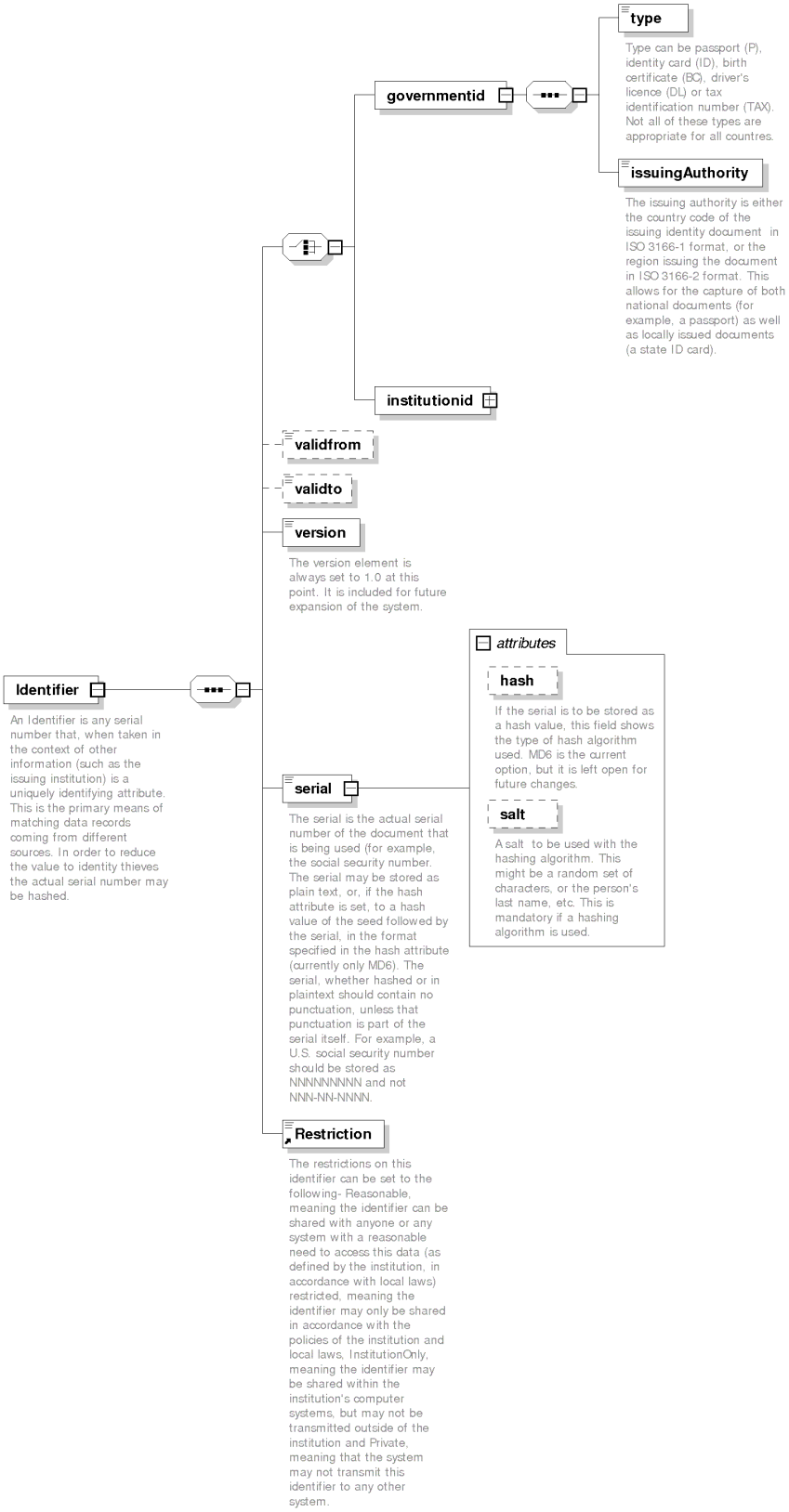
[29] (Internal, local, state or national identification numbers or codes used to disambiguate students)Note that the US Department of Education advises against the use of the SSN as an identifier whereas the Department of Commerce states their utility in longitudinal data systems.

	FirstName MiddleName LastName Suffix Title Composite	Last First Middle Preferred plus enumerated name type.[30]	Last Name Generation Code Title Alias Former Legal Name Last Name at Birth Nickname Tribal or Clan Name Name of Individual [composite name]	Last Suffix Title name type (internally defined)
High school	Freeform name followed by optional ID Group Number	Not declared	Defined in Element B and therefore not enumerated	Ignored (see note 2).
Address and other contact information	**"Contacts" – address format as "domestic" or "international"** All addresses have core of Address Line 1..3 City US Addresses have StateCode, Postal Code International addresses have StateProvince, postal code, country code **Telephone** Country Code Area Number Extension **Email** **URL**	**Defined address list, with atomic and freeform address areas.** Duplicated information implied in specification. Address type numbered according to NCE0025 standard. Geographical grid reference as x,y coordinates on globe **Phone number list**, with NCES0280 telephone number type, freeform number, freeform extension and enumerated listed status[31]. **Email list**: Freeform email address plus enumerated type field[32].	**Street addressed defined as atomic list following US standard with a country code added to the end, and enumerated address type** [0369] **Email address defined as type, and freeform address** **URL with no type**	**Line Broken Address Type** Lines 1..3 City, State, Postal Code Country Internally Defined Address Type **Email address** (freeform) with **type indicators** (internally defined) **Telephone number** Number Extension Country code type
Residency	Grouping of state, territory and country codes with residency classification as: Resident Nonresident Notreported	**Countries of Citizenship** [list based on ISO31661/ISO31663 country name] **Countries of Residency** [list based on ISO31661/ISO31663 country name]	Residency status with respect to a school district defined.	Purely defines if the student is "in state" – determination code, and in state flag
Vitality	**Deceased** (true/false)	Not declared	Under Enrollment (part B) 1070 **Death Date** 1080 **Death Cause**	Not specifically Defined
Other		**Alert Message** **Medical Alert Message** **Projected Graduation Year** **Graduation Date** **Acceptable Use** **IDEA** **Migrant** **Title1**	Large quantities of socioeconomic data captured, including family income, etc. Mostly in the 600-700 range.	

[30] 01- Given, 02- Current legal, 03- Legal, 04- Name of Record, 05-Previous Name, 07- Married Name, 08- Professional Name
[31] Listed, Unlisted, Unknown
[32] Primary, Alternate1..4

	Gifted and Talented **Economic Disadvantage** **ELL** **Homeless** **504** **Vocational Concentrator** **Immigrant** **Neglected Delinquent** **Metadata** **Extended Elements**

Appendix B: Proposed Schema for the Identifier


type
Type can be passport (P), identity card (ID), birth certificate (BC), driver's licence (DL) or tax identification number (TAX). Not all of these types are appropriate for all countres.
governmentid
issuingAuthority
The issuing authority is either the country code of the issuing identity document in ISO 3166-1 format, or the region issuing the document in ISO 3166-2 format. This allows for the capture of both national documents (for example, a passport) as well as locally issued documents (a state ID card).
institutionid
validfrom
validto
version
The version element is always set to 1.0 at this point. It is included for future expansion of the system.
attributes
hash
If the serial is to be stored as a hash value, this field shows the type of hash algorithm used. MD6 is the current option, but it is left open for future changes.
salt
A salt to be used with the hashing algorithm. This might be a random set of characters, or the person's last name, etc. This is mandatory if a hashing algorithm is used.
Identifier
An Identifier is any serial number that, when taken in the context of other information (such as the issuing institution) is a uniquely identifying attribute. This is the primary means of matching data records coming from different sources. In order to reduce the value to identity thieves the actual serial number may be hashed.
serial
The serial is the actual serial number of the document that is being used (for example, the social security number. The serial may be stored as plain text, or, if the hash attribute is set, to a hash value of the seed followed by the serial, in the format specified in the hash attribute (currently only MD6). The serial, whether hashed or in plaintext should contain no punctuation, unless that punctuation is part of the serial itself. For example, a U.S. social security number should be stored as NNNNNNNNN and not NNN-NN-NNNN.
Restriction
The restrictions on this identifier can be set to the following- Reasonable, meaning the identifier can be shared with anyone or any system with a reasonable need to access this data (as defined by the institution, in accordance with local laws) restricted, meaning the identifier may only be shared in accordance with the policies of the institution and local laws, InstitutionOnly, meaning the identifier may be shared within the institution's computer systems, but may not be transmitted outside of the institution and Private, meaning that the system may not transmit this identifier to any other system.

Element	Description	Required	Multiplicity	Datatype
Governmentid	This is the container that defines a standard form of identification issued by a government. Any identifier must have a governmentid or a institititionid container.	Either governmentid or institutionid container is a required field.	0..1	Container
Type	Sub-element of governmentid, this defines one of the standard forms of government issued identifiers.	Required	1	XS:String Possible values are DL (Driver's License) P (Passport) ID (Identity card) BC (Birth Certificate) TAX (Tax ID)
issuingAuthority	Government responsible for issuing the ID. Can either be a country code, or a region code.	Required	1	XS: String. Values are limited to ISO 3166-1 codes for countries, or ISO 3166-2 for regions.
institutionID	Container for institutionally issued identifiers, such as student identifiers, or other government identifiers that do not fall into one of the five common government identifiers.	Either governmentid or institutionid container is a required field.	0..1	Container
IssuerIdentifier	Base URL for the institution that has issued an identifier. Examples would include Georgetown.edu, keele.ac.uk, paris-sorbonne.fr	Required	1	XS:String
Validfrom	Date from which an identifier is valid. Can be used to resolve conflicts of identifiers, for example, where a passport number changes when it is re-issued after expiration.	Optional	0..1	XS:date
Validto	Date to which an identifier is valid. Can be used to resolve conflicts of identifiers, for example, where a passport number changes when it is re-issued after expiration.	Optional	0..1	XS:date
Version	Versioning information for the serial	Required	1	XS:string

	number. At this point, the version will be set to 1.0, but this element may be changed in future versions of the system.			
Serial	This is the actual serial number that is being used as an identifier (the passport number, the driver's license number, the student identifier, etc.) Only the code itself should be stored, not any readability punctuation, unless that punctuation forms part of the serial itself. For example, a U.S. social security number would be stored as NNNNNNNNN and not NNN-NN-NNNN. Serial can have the optional attributes of hash(can equal MD6) and seed (mandatory if hash is set). In this case, the serial is stored as an MD6 hash, with the seed concatenated on to the front of the serial.	Required	1	XS:String
Hash	Attribute of serial. If set, the serial is stored as a hashed version of the serial, with the attribute "seed" concatenated onto the front of the serial.	Optional	0..1	XS:String Permitted value is MD6
Salt	The salt is any collection of alphanumeric characters which is concatenated onto the front of the serial before the value is hashed. In order for institutions to be able to match on this field, one of them must have access to the original serial, or there needs to be agreement on the value of the salt. It could, for example, be set to the person's date of birth, though this might cause some problems with numbers like the United States Social Security Number, which are generally issued with a sequential pattern based on time of issue. Another option would be some element of the person's name.	Mandatory if hash is set.	0..1	XS:String
Restriction	The restrictions on this identifier can be set to the following- Reasonable, meaning the identifier can be shared	Mandatory	1	XS:String. Permitted values are

	with anyone or any system with a reasonable need to access this data (as defined by the institution, in accordance with local laws) restricted, meaning the identifier may only be shared in accordance with the policies of the institution and local laws, InstitutionOnly, meaning the identifier may be shared within the institution's computer systems, but may not be transmitted outside of the institution, and Private, meaning that the system may not transmit this identifier to any other system.	Reasonable Restricted InstitutionOnly Private

Examples

For a U.K. National Insurance number.

```
<Identifier>
<governmentid>
      <type>TAX</type
      <issuingAuthority>GB</issuingAuthority>
</governmentid>
<version>1.0</version>
<serial>TN019281M</serial>
<Restriction>Reasonable</Restriction>
</Identifier>
```

For a U.S. driver's license issued in the state of California.

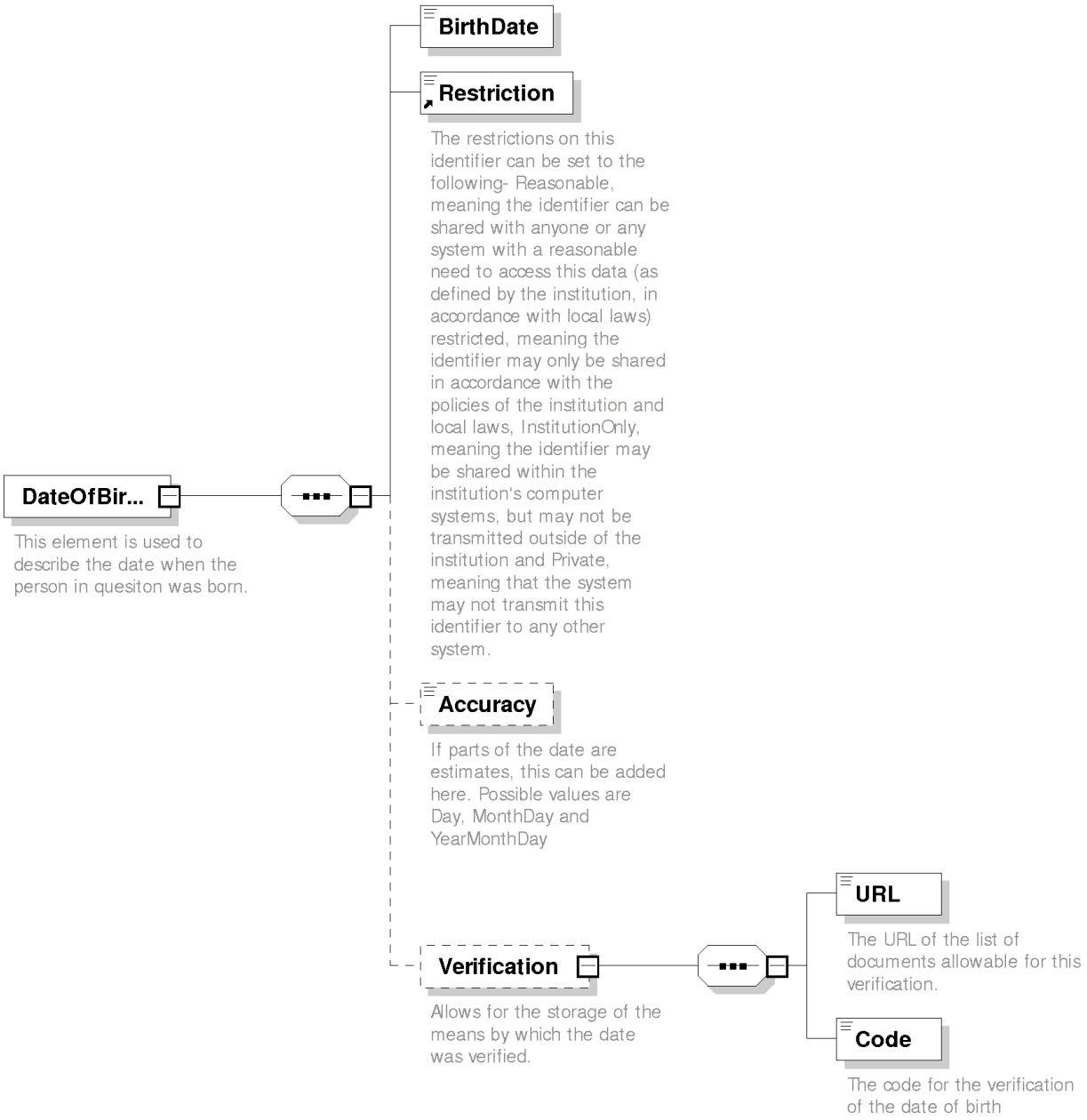
```
<Identifier>
<governmentid>
      <type>DL</type>
      <issuingAuthority>US-CA</issuingAuthority>
</governmentid>
<validfrom>2000-09-01</validfrom>
<validto>2005-09-01</validto>
<version>1.0</version>
<serial hash="MD6" seed="EJKRIU">
38a96ee214464c59f750ebc0e7c229d7</serial>
<Restriction>Restricted</Restriction>
</Identifier>
```

For a student ID card issued by the George Washington University.

```
<Identifier>
<institutionid>
      <issueridentifier>gwu.edu</type
</instititutionid>
<version>1.0</version>
```

```
<serial hash=”MD6” seed=”LKSI”> 63cfcf63820ae5ca7cccc8158141eae3</serial>
<Restriction>Reasonable</Restriction>
</Identifier>
```

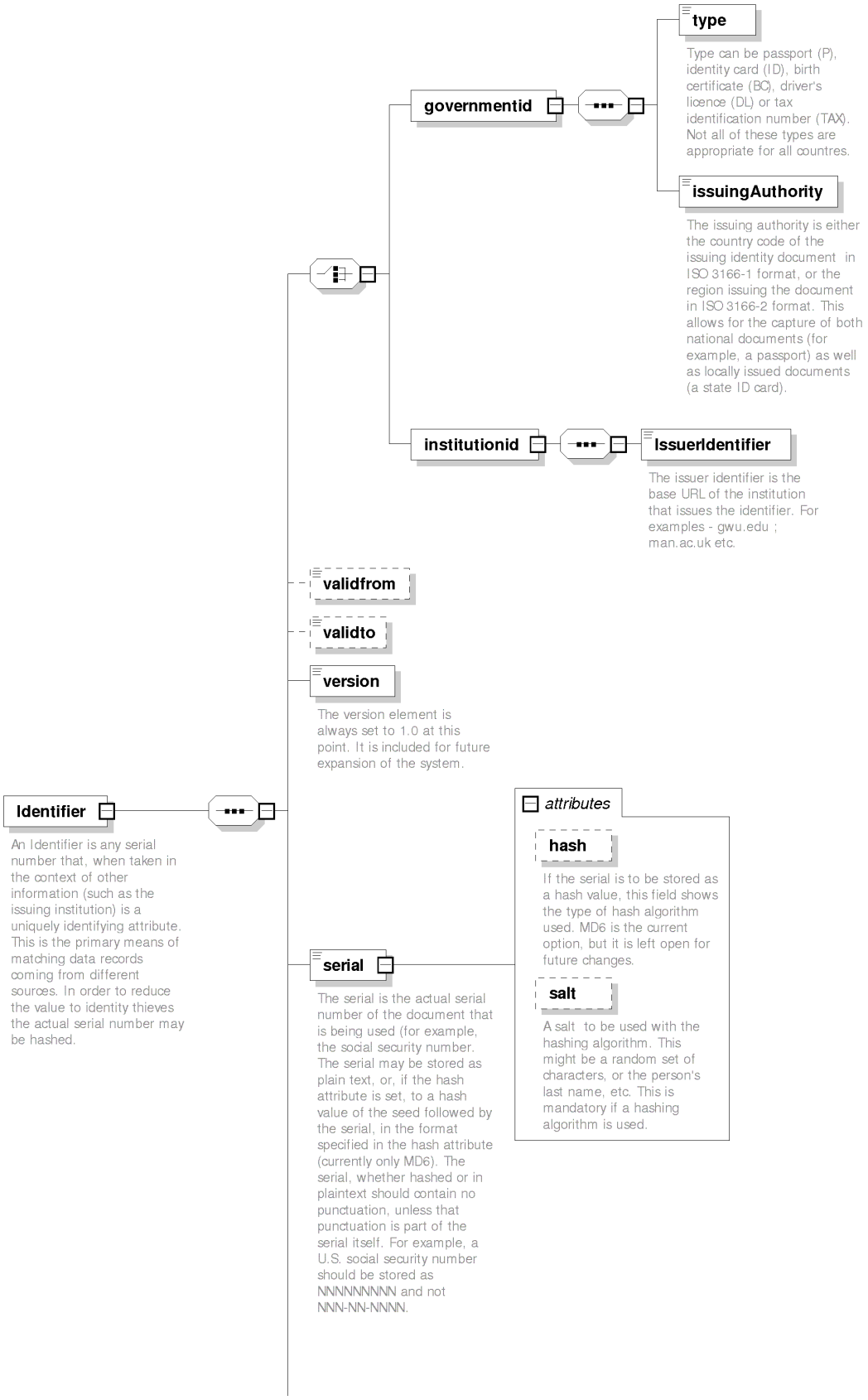
Appendix C: Proposed Schema for the Date of Birth

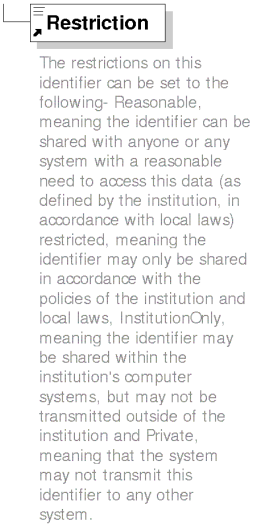


Element	Description	Required	Multiplicity	Datatype
Birthdate	This is the actual date of birth to be stored by the system.	Required	1	Date
Type	Sub-element of governmentid, this defines one of the standard forms of government issued identifiers.	Required	1	XS:String Possible values are DL (Driver's License) P (Passport) ID (Identity card) BC (Birth Certificate) TAX (Tax ID)
Restriction	The restrictions on this identifier can be set to the following- Reasonable, meaning the identifier can be shared with anyone or any system with a reasonable need to access this data (as defined by the institution, in accordance with local laws) restricted, meaning the identifier may only be shared in accordance with the policies of the institution and local laws, InstitutionOnly, meaning the identifier may be shared within the institution's computer systems, but may not be transmitted outside of the institution, and Private, meaning that the system may not transmit this identifier to any other system.	Mandatory	1	XS:String. Permitted values are Reasonable Restricted InstitutionOnly Private
Accuracy	The accuracy field allows the system to mark out those dates which might not be accurate in all fields to assist matching of partially incomplete records. The field marks which fields may not be accurate.	Optional	0..1	XS:string. Possible values are Day, MonthDay and YearMonthDay
Verification	This container allows for the storage of a verification element. This allows institutions to show the type of document they checked in order to verify the birth date.	Optional	0..1	Container
URL	This URL references the document that is lists the codes and acceptable documents for the verification of the birth date.	Mandatory (if verification is to be used)	0..1	XS:string in RFC 1783 format.
Code	The code is the specified documents code from the document referenced in	Mandatory (if verification is		XS:String

the URL field.	to be used)

Appendix D: Proposed Schema for the Name


type
Type can be passport (P), identity card (ID), birth certificate (BC), driver's licence (DL) or tax identification number (TAX). Not all of these types are appropriate for all countres.
governmentid
issuingAuthority
The issuing authority is either the country code of the issuing identity document in ISO 3166-1 format, or the region issuing the document in ISO 3166-2 format. This allows for the capture of both national documents (for example, a passport) as well as locally issued documents (a state ID card).
institutionid
IssuerIdentifier
The issuer identifier is the base URL of the institution that issues the identifier. For examples - gwu.edu ; man.ac.uk etc.
validfrom
validto
version
The version element is always set to 1.0 at this point. It is included for future expansion of the system.
Identifier
An Identifier is any serial number that, when taken in the context of other information (such as the issuing institution) is a uniquely identifying attribute. This is the primary means of matching data records coming from different sources. In order to reduce the value to identity thieves the actual serial number may be hashed.
attributes
hash
If the serial is to be stored as a hash value, this field shows the type of hash algorithm used. MD6 is the current option, but it is left open for future changes.
salt
A salt to be used with the hashing algorithm. This might be a random set of characters, or the person's last name, etc. This is mandatory if a hashing algorithm is used.
serial
The serial is the actual serial number of the document that is being used (for example, the social security number. The serial may be stored as plain text, or, if the hash attribute is set, to a hash value of the seed followed by the serial, in the format specified in the hash attribute (currently only MD6). The serial, whether hashed or in plaintext should contain no punctuation, unless that punctuation is part of the serial itself. For example, a U.S. social security number should be stored as NNNNNNNNN and not NNN-NN-NNNN.

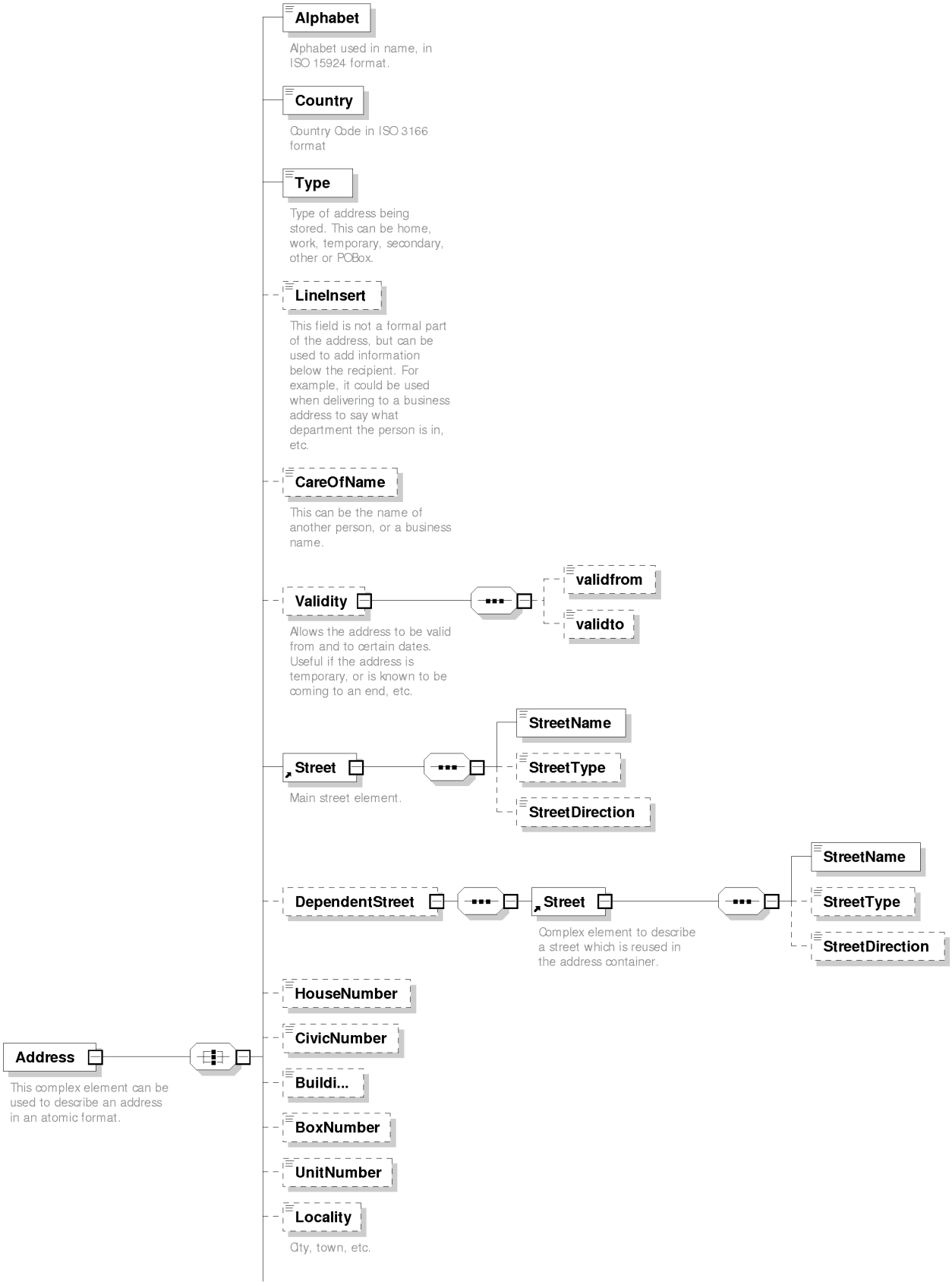

Restriction
The restrictions on this identifier can be set to the following- Reasonable, meaning the identifier can be shared with anyone or any system with a reasonable need to access this data (as defined by the institution, in accordance with local laws) restricted, meaning the identifier may only be shared in accordance with the policies of the institution and local laws, InstitutionOnly, meaning the identifier may be shared within the institution's computer systems, but may not be transmitted outside of the institution and Private, meaning that the system may not transmit this identifier to any other system.

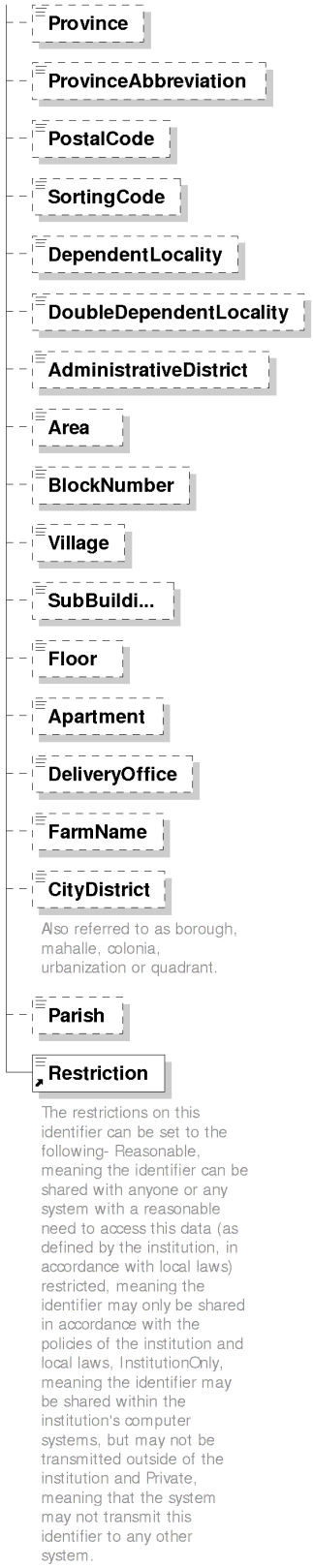
Element	Description	Required	Multiplicity	Datatype
PersonName	This is the container that is used to define a person's name.			Container
Type	This element describes the type of name being stored.	Mandatory	1	XS:String. Allowed values are Current Former Alias Other
Alphabet	Alphabet used in name, in ISO 15924 format.	Mandatory	1	XS:Sting
GivenName	Container for a person's given name(s)	Mandatory	1..*	Container
ID (attribute of GivenName)	Each name element must be assigned a uniquely identifying number.	Mandatory	1	XS:Integer
Optionality (Attribute of GivenName)	Is the name an optional name. For example, English middle names, or commonly omitted parts of names (such as "bin" in Arabic names).	Optional	0..1	XS:String. Allowed values are mandatory or optional.
Initialable (Attribute of GivenName)	Is the name commonly reduced to a single initial, for example, a middle name.	Optional	0..1	XS:String. Allowed values are yes and no.
Uncapitalized (Attribute of GivenName)	Is this name un-capitalized? Names without this flag are assumed to have the first letter capitalized.	Optional	0..1	XS:String. Allowed values are yes and no.
Linktoid (Attribute of GivenName)	ID of another name to which this name is linked. This value is used in conjunction with the next three attributes.	Optional, unless any of the following three values are given, in which case it is mandatory.	0..1	XS:Integer. Allowed values are the IDs of other names within this PersonalName container.
mandatorywithlink(Attribute of GivenName)	If the name referenced in linktoid is present, then this name must also be present. For example, in an Asian name such as In Wook, Wook cannot occur without In, and vice versa.	Optional	0..1	XS:String. Allowed values are yes and no.
orderirrelevantwithlink(Attribute of GivenName)	If the order with respect to the name in linktoid is irrelevant, this flag can be set. Common with hyphenated last names, where no specific order is preferred.	Optional	0..1	XS:String. Allowed values are yes and no.
linkcharacter(Attribute of GivenName)	How are this name, and the name refenced in linktoid joined together? If this option is omitted then a space is assumed.	Optional	0..1	XS:String. Allowed values are hyphenated or nospace
Nickname (Attribute	Is this name a nickname (i.e. a	Optional	0..1	XS:String.

of GivenName)	contraction or expansion of a legal name, or a name by which a person goes that is not really part of their name).			Allowed values are yes and no.
FamilyName	Container for a person's family name(s)	Mandatory	1..*	Container
ID (attribute of FamilyName)	Each name element must be assigned a uniquely identifying number.	Mandatory	1	XS:Integer
Optionality (Attribute of FamilyName)	Is the name an optional name. For example, English middle names, or commonly omitted parts of names (such as "bin" in Arabic names).	Optional	0..1	XS:String. Allowed values are mandatory or optional.
Initialable (Attribute of FamilyName)	Is the name commonly reduced to a single initial, for example, a middle name.	Optional	0..1	XS:String. Allowed values are yes and no.
Uncapitalized (Attribute of FamilyName)	Is this name un-capitalized? Names without this flag are assumed to have the first letter capitalized.	Optional	0..1	XS:String. Allowed values are yes and no.
Linktoid (Attribute of FamilyName)	ID of another name to which this name is linked. This value is used in conjunction with the next three attributes.	Optional, unless any of the following three values are given, in which case it is mandatory.	0..1	XS:Integer. Allowed values are the IDs of other names within this PersonalName container.
mandatorywithlink(Attribute of FamilyName)	If the name referenced in linktoid is present, then this name must also be present. For example, in an Asian name such as In Wook, Wook cannot occur without In, and vice versa.	Optional	0..1	XS:String. Allowed values are yes and no.
orderirrelevantwithlink(Attribute of FamilyName)	If the order with respect to the name in linktoid is irrelevant, this flag can be set. Common with hyphenated last names, where no specific order is preferred.	Optional	0..1	XS:String. Allowed values are yes and no.
linkcharacter(Attribute of FamilyName)	How are this name, and the name refenced in linktoid joined together? If this option is omitted then a space is assumed.	Optional	0..1	XS:String. Allowed values are hyphenated or nospace
Nickname (Attribute of FamilyName)	Is this name a nickname (i.e. a contraction or expansion of a legal name, or a name by which a person goes that is not really part of their name).	Optional	0..1	XS:String. Allowed values are yes and no.
Title	Title that can be added to the name. Can be a simple title (Mister, Madam, etc.) an academic title (Doctor) or any other preceding title.	Optional	0..1	XS:String

ID (attribute of Title)	Each name element must be assigned a uniquely identifying number.	Mandatory	1	XS:Integer
Suffix	Any part of the name which is added to the end. Examples would be generation codes (Junior, Senior, III) Academic degrees, etc.	Optional	0..*	XS:String
ID (attribute of Suffix)	Each name element must be assigned a uniquely identifying number.	Mandatory	1	XS:Integer
FormatShortName	List of IDs to form the person's shortname. Example for most English names would be their first name.	Mandatory	1	XS:String
FormatPreferredFormalName	List of IDs to form the person's preferred formal name. An example might be the person's first and family name, or their nickname and their family name.	Mandatory	1	XS:String
FormatFullName	List of IDs to form a person's complete name. Will likely include all elements of the name.	Mandatory	1	XS:String
FormatFamilyName	List of IDs to form a person's entire family name.	Mandatory	1	XS:String

Appendix E: Proposed Schema for the Address

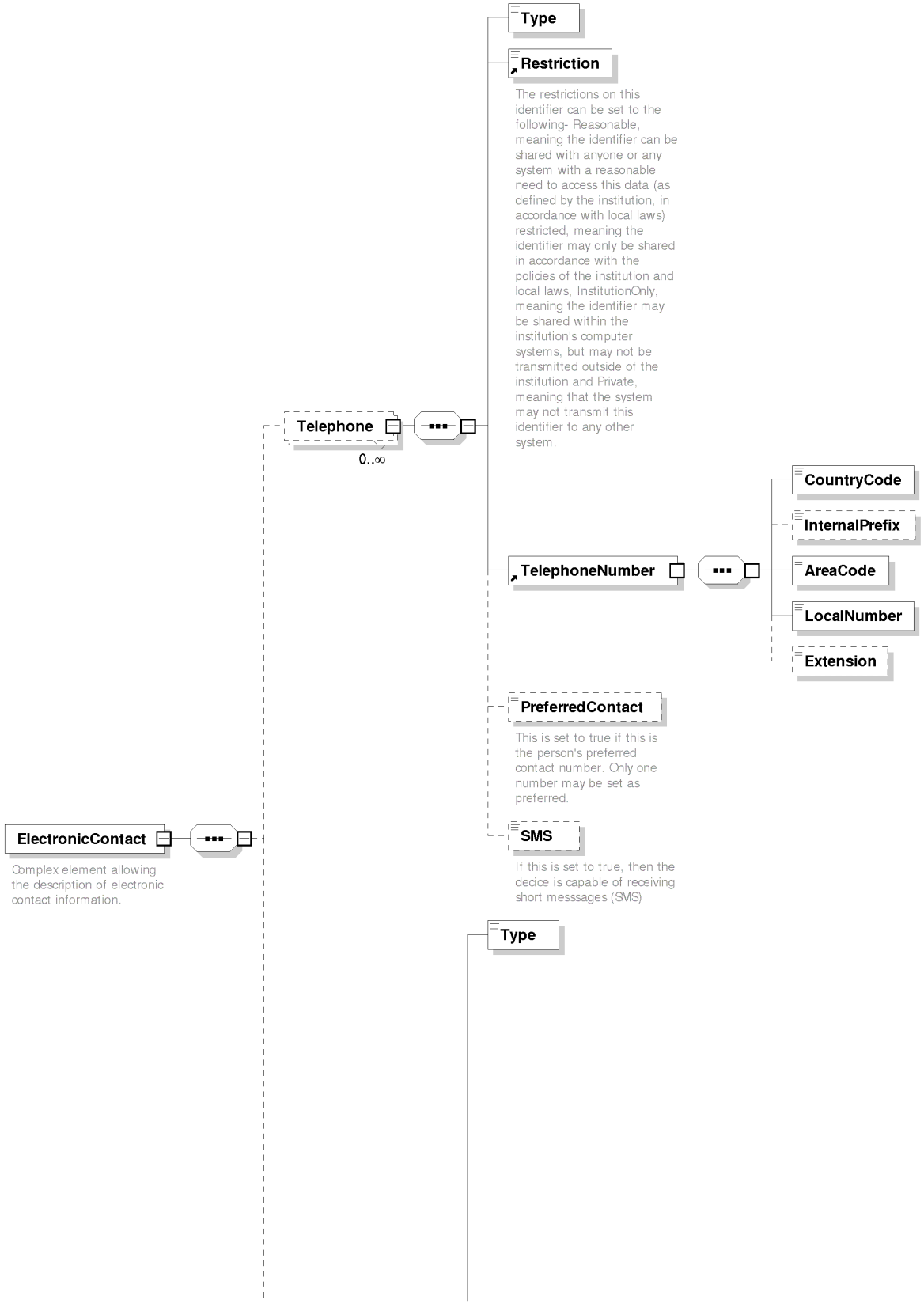

Alphabet
Alphabet used in name, in ISO 15924 format.
Country
Country Code in ISO 3166 format
Type
Type of address being stored. This can be home, work, temporary, secondary, other or POBox.
LineInsert
This field is not a formal part of the address, but can be used to add information below the recipient. For example, it could be used when delivering to a business address to say what department the person is in, etc.
CareOfName
This can be the name of another person, or a business name.
Validity
Allows the address to be valid from and to certain dates. Useful if the address is temporary, or is known to be coming to an end, etc.
validfrom
validto
Street
Main street element.
StreetName
StreetType
StreetDirection
DependentStreet
Street
Complex element to describe a street which is reused in the address container.
StreetName
StreetType
StreetDirection
HouseNumber
CivicNumber
Buildi...
BoxNumber
UnitNumber
Locality
City, town, etc.
Address
This complex element can be used to describe an address in an atomic format.

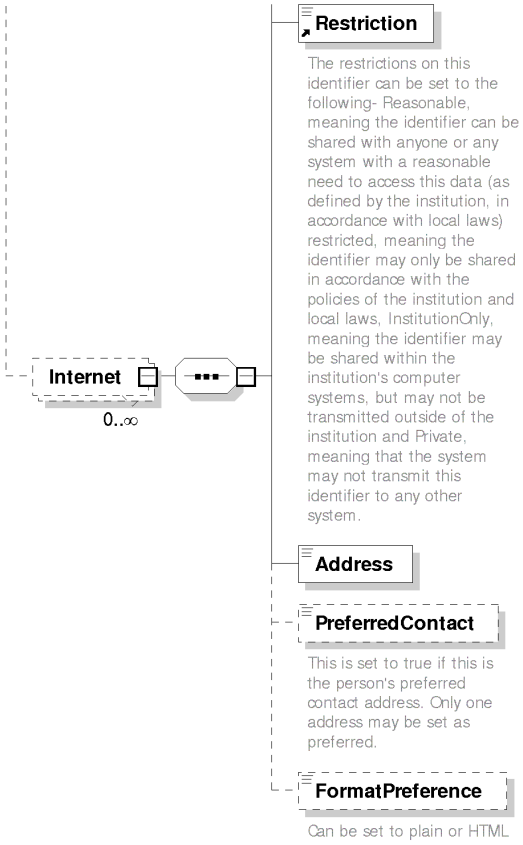

Province
ProvinceAbbreviation
PostalCode
SortingCode
DependentLocality
DoubleDependentLocality
AdministrativeDistrict
Area
BlockNumber
Village
SubBuildi...
Floor
Apartment
DeliveryOffice
FarmName
CityDistrict
Also referred to as borough, mahalle, colonia, urbanization or quadrant.
Parish
Restriction
The restrictions on this identifier can be set to the following- Reasonable, meaning the identifier can be shared with anyone or any system with a reasonable need to access this data (as defined by the institution, in accordance with local laws) restricted, meaning the identifier may only be shared in accordance with the policies of the institution and local laws, InstitutionOnly, meaning the identifier may be shared within the institution's computer systems, but may not be transmitted outside of the institution and Private, meaning that the system may not transmit this identifier to any other system.

Element	Description	Required	Multiplicity	Datatype
Address	Complex element type that defines a postal address for the majority of countries.			Complex
Alphabet	The alphabet being used to define the address in ISO 15924 format.	Mandatory	1	XS:String
Restriction	The restrictions on this address can be set to the following- Reasonable, meaning the identifier can be shared with anyone or any system with a reasonable need to access this data (as defined by the institution, in accordance with local laws) restricted, meaning the identifier may only be shared in accordance with the policies of the institution and local laws, InstitutionOnly, meaning the identifier may be shared within the institution's computer systems, but may not be transmitted outside of the institution, and Private, meaning that the system may not transmit this identifier to any other system.	Mandatory	1	XS:String. Permitted values are Reasonable Restricted InstitutionOnly Private
Country	Country of address, using ISO 3166 codes.	Mandatory	1	XS:String
Type	Type of address being stored. Can be Home, Work, Temporary, Secondary, POBox or Other.	Mandatory	1	XS:String
LineInsert	This field is not a formal part of the address, but can be used to add information below the recipient. For example, it could be used when delivering to a business address to say what department the person is in, etc.	Optional	0..1	XS:String
CareOfName	This can be either another person's name, or a business name.	Optional	0..1	XS:String
Validity	Allows the address to be valid from and to certain dates. Useful if the address is temporary, or is known to be coming to an end, etc.	Optional	0..1	Complex
Validfrom	The date from which the address is valid.	Optional	0..1	XS:Date
Validto	The date to which the address is valid.	Optional	0..1	XS:Date
Street	Container for street description	Optional	0..1	Street

StreetName	Name of the Street	Mandatory in a Street element.	1	XS:String
StreetType	Type of the street, for example Road, Avenue, Alley, etc.	Optional	0..1	XS:String
StreetDirection	Geographical direction of the street, for example North, South, etc.	Optional	0..1	XS:String
DependentStreet	Some addresses require a dependent street. A dependent street is stored in the same container as the main street.	Optional	0..1	Street
HouseNumber	The number of the house on the street. If the house has a name, it can also be stored in this element.	Optional	0..1	XS:String
CivicNumber		Optional	0..1	XS:String
BuildingNumber		Optional	0..1	XS:String
BoxNumber		Optional	0..1	XS:String
UnitNumber		Optional	0..1	XS:String
Locality	City, town, etc.	Optional	0..1	XS:String
Province	State, Province, major county, etc.	Optional	0..1	XS:String
ProvinceAbbreviation	State or province abbreviation.	Optional	0..1	XS:String
PostalCode	Post Code, Zip, etc.	Optional	0..1	XS:String
SortingCode		Optional	0..1	XS:String
DependentLocality		Optional	0..1	XS:String
DoubleDependentLocality		Optional	0..1	XS:String
AdministrativeDistrict		Optional	0..1	XS:String
Area		Optional	0..1	XS:String
BlockNumber		Optional	0..1	XS:String
Village		Optional	0..1	XS:String
SubBuilding		Optional	0..1	XS:String
Floor		Optional	0..1	XS:String
Apartment		Optional	0..1	XS:String
DeliveryOffice		Optional	0..1	XS:String
FarmName		Optional	0..1	XS:String
CityDistrict	May also be referred to as Mahalle, Colonia, Borough, Urbanization or quadrant.	Optional	0..1	XS:String
Parish		Optional	0..1	XS:String

Appendix F: Proposed Schema for the Electronic Contact Information

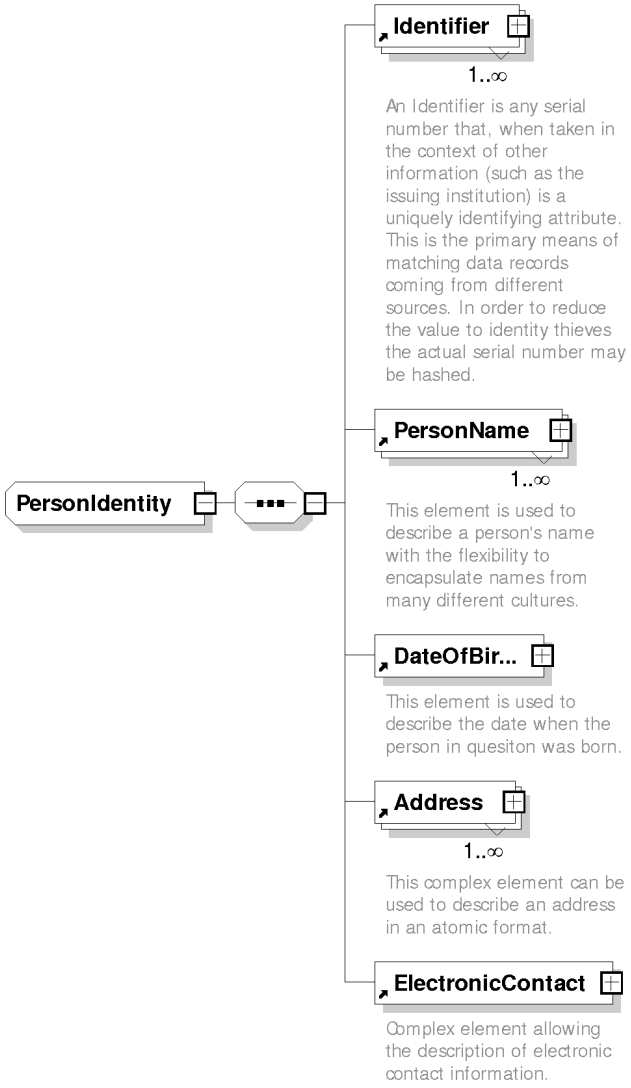



Restriction
The restrictions on this identifier can be set to the following- Reasonable, meaning the identifier can be shared with anyone or any system with a reasonable need to access this data (as defined by the institution, in accordance with local laws) restricted, meaning the identifier may only be shared in accordance with the policies of the institution and local laws, InstitutionOnly, meaning the identifier may be shared within the institution's computer systems, but may not be transmitted outside of the institution and Private, meaning that the system may not transmit this identifier to any other system.
Internet
0..∞
Address
PreferredContact
This is set to true if this is the person's preferred contact address. Only one address may be set as preferred.
FormatPreference
Can be set to plain or HTML

Element	Description	Required	Multiplicity	Datatype
ElectronicContact	Container element containing any forms of electronic contact information. This field is designed to evolve with new methods of contact.			Container
Telephone	Container for the storage of different telephone numbers.	Optional	0..*	Container
Type	Enumerated types of telephone number.	Mandatory	1	XS:String Enumerations available: Home Work Mobile Facsimile Other
Restricted	Restricted flag as already defined.	Mandatory	1	XS:String
TelephoneNumber	Container to store the numerical part of the number.	Mandatory	1	Container
CountryCode	Dialing code for the country.	Mandatory	1	XS:String Allowable values are those defined by the ITU's E.164 and E.123 standards.
InternalPrefix	The number that must be dialed before the number if the call is internal to a country. For most European nations, this is 0.	Optional	0..1	XS:String
AreaCode	The area code for the number, not including the internal prefix. For example, the code for London would be 20, not 020. The code for Washington, DC is 202, not 1202.	Mandatory	1	XS:String
LocalNumber	The local part of the number.	Mandatory	1	XS:String. Spaces may be included, though any interpreting software should ignore them.
Extension	If an extension is required, it is listed here	Optional	0..1	XS:String
PreferredContact	Is this is the number on which a person prefers to be contacted by the organization, then this value is set to	Optional	0..1	XS:Boolean

	true. If omitted, it is assumed that it is false.			
SMS	If this value is set to true, then the telephone number is able to accept short messages (SMS). If it is omitted, then it is assumed to be false.	Optional	0..1	XS:Boolean
Internet	Container type to allow the storage of different forms of addresses on the Internet.	Optional	0..*	Container
Type	Type of address being stored	Mandatory	1	XS:String PersonalEmail WorkEmail OtherEmail Website Other
Restricted	Restriction flag as defined elsewhere.	Mandatory	1	XS:String
Address	The actual address.	Mandatory	1	XS:String This should be formatted as a URI, including the protocol (http:// , mailto:// etc.
PreferredContact	If this is the address that the person wishes to be contact on then this flag will be set to true. If omitted, it is assumed that it is false.	Optional	0..1	XS:Boolean
FormatPreference	Allows the preferred form of communication to be set.	Optional	0..1	XS:String Can be set to Plaintext or HTML

Appendix G: Proposed Schema for the Overall Identifier



Appendix H: Proposed XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
	<xs:element name="Restriction" default="Public">
		<xs:annotation>
			<xs:documentation>The restrictions on this identifier can be set to the following- Reasonable, meaning the
identifier can be shared with anyone or any system with a reasonable need to access this data (as defined by the institution, in
accordance with local laws) restricted, meaning the identifier may only be shared in accordance with the policies of the institution and
local laws, InstitutionOnly, meaning the identifier may be shared within the institution's computer systems, but may not be transmitted
outside of the institution and Private, meaning that the system may not transmit this identifier to any other
system.</xs:documentation>
		</xs:annotation>
		<xs:simpleType>
			<xs:restriction base="xs:string">
				<xs:enumeration value="Public"/>
				<xs:enumeration value="Restricted"/>
				<xs:enumeration value="InstitutionOnly"/>
				<xs:enumeration value="Private"/>
			</xs:restriction>
		</xs:simpleType>
	</xs:element>
	<xs:element name="Street">
		<xs:annotation>
			<xs:documentation>Complex element to describe a street which is reused in the address
container.</xs:documentation>
		</xs:annotation>
		<xs:complexType>
			<xs:sequence>
				<xs:element name="StreetName"/>
				<xs:element name="StreetType" minOccurs="0"/>
				<xs:element name="StreetDirection" minOccurs="0"/>
```

```
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="Address">
                <xs:annotation>
                    <xs:documentation>This complex element can be used to describe an address in an atomic format.</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:all>
                        <xs:element name="Alphabet">
                            <xs:annotation>
                                <xs:documentation>Alphabet used in name, in ISO 15924 format.</xs:documentation>
                            </xs:annotation>
                        </xs:element>
                        <xs:element name="Country">
                            <xs:annotation>
                                <xs:documentation>Country Code in ISO 3166 format</xs:documentation>
                            </xs:annotation>
                        </xs:element>
                        <xs:element name="Type">
                            <xs:annotation>
                                <xs:documentation>Type of address being stored. This can be home, work, temporary, secondary, other or POBox.</xs:documentation>
                            </xs:annotation>
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:enumeration value="Home"/>
                                    <xs:enumeration value="Work"/>
                                    <xs:enumeration value="Temporary"/>
                                    <xs:enumeration value="Secondary"/>
                                    <xs:enumeration value="POBox"/>
                                    <xs:enumeration value="Other"/>
                                </xs:restriction>
```

```
                    </xs:simpleType>
                </xs:element>
                <xs:element name="LineInsert" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>This field is not a formal part of the address, but can be used to add
information below the recipient. For example, it could be used when delivering to a business address to say what department the
person is in, etc.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="CareOfName" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>This can be the name of another person, or a business
name.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="Validity" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>Allows the address to be valid from and to certain dates. Useful if
the address is temporary, or is known to be coming to an end, etc.</xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="validfrom" type="xs:date" minOccurs="0"/>
                            <xs:element name="validto" type="xs:date" minOccurs="0"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element ref="Street">
                    <xs:annotation>
                        <xs:documentation>Main street element.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="DependentStreet" minOccurs="0">
                    <xs:complexType>
```

```
                        <xs:sequence>
                            <xs:element ref="Street"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="HouseNumber" type="xs:string" minOccurs="0"/>
                <xs:element name="CivicNumber" type="xs:string" minOccurs="0"/>
                <xs:element name="Building" type="xs:string" minOccurs="0"/>
                <xs:element name="BoxNumber" type="xs:string" minOccurs="0"/>
                <xs:element name="UnitNumber" type="xs:string" minOccurs="0"/>
                <xs:element name="Locality" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>City, town, etc.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="Province" minOccurs="0"/>
                <xs:element name="ProvinceAbbreviation" minOccurs="0"/>
                <xs:element name="PostalCode" minOccurs="0"/>
                <xs:element name="SortingCode" minOccurs="0"/>
                <xs:element name="DependentLocality" minOccurs="0"/>
                <xs:element name="DoubleDependentLocality" minOccurs="0"/>
                <xs:element name="AdministrativeDistrict" minOccurs="0"/>
                <xs:element name="Area" minOccurs="0"/>
                <xs:element name="BlockNumber" minOccurs="0"/>
                <xs:element name="Village" minOccurs="0"/>
                <xs:element name="SubBuilding" minOccurs="0"/>
                <xs:element name="Floor" minOccurs="0"/>
                <xs:element name="Apartment" minOccurs="0"/>
                <xs:element name="DeliveryOffice" minOccurs="0"/>
                <xs:element name="FarmName" minOccurs="0"/>
                <xs:element name="CityDistrict" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>Also referred to as borough, mahalle, colonia, urbanization or
quadrant.</xs:documentation>
```

```
                                    </xs:annotation>
                                </xs:element>
                                <xs:element name="Parish" minOccurs="0"/>
                                <xs:element ref="Restriction"/>
                            </xs:all>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="TelephoneNumber">
                        <xs:annotation>
                            <xs:documentation> This element is used to define the standard format for telephone numbers which is part of the electronic contact information.</xs:documentation>
                        </xs:annotation>
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="CountryCode"/>
                                <xs:element name="InternalPrefix" minOccurs="0"/>
                                <xs:element name="AreaCode"/>
                                <xs:element name="LocalNumber"/>
                                <xs:element name="Extension" minOccurs="0"/>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="ElectronicContact">
                        <xs:annotation>
                            <xs:documentation>Complex element allowing the description of electronic contact information.</xs:documentation>
                        </xs:annotation>
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="Telephone" minOccurs="0" maxOccurs="unbounded">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element name="Type">
                                                <xs:simpleType>
```

```
                                        <xs:restriction base="xs:string">
                                            <xs:enumeration value="Home"/>
                                            <xs:enumeration value="Mobile"/>
                                            <xs:enumeration value="Work"/>
                                            <xs:enumeration value="Facsimilie"/>
                                            <xs:enumeration value="Other"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                                <xs:element ref="Restriction"/>
                                <xs:element ref="TelephoneNumber"/>
                                <xs:element name="PreferredContact" type="xs:boolean" minOccurs="0">
                                    <xs:annotation>
                                        <xs:documentation>This is set to true if this is the person's
preferred contact number. Only one number may be set as preferred.</xs:documentation>
                                    </xs:annotation>
                                </xs:element>
                                <xs:element name="SMS" type="xs:boolean" minOccurs="0">
                                    <xs:annotation>
                                        <xs:documentation>If this is set to true, then the decice is capable
of receiving short messsages (SMS)</xs:documentation>
                                    </xs:annotation>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="Internet" minOccurs="0" maxOccurs="unbounded">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="Type">
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:enumeration value="PersonalEmail"/>
                                            <xs:enumeration value="WorkEmail"/>
```

```
                                                    <xs:enumeration value="OtherEmail"/>
                                                    <xs:enumeration value="Website"/>
                                                    <xs:enumeration value="Other"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:element>
                                        <xs:element ref="Restriction"/>
                                        <xs:element name="Address" type="xs:string"/>
                                        <xs:element name="PreferredContact" type="xs:boolean" minOccurs="0">
                                            <xs:annotation>
                                                <xs:documentation>This is set to true if this is the person's
preferred contact address. Only one address may be set as preferred.</xs:documentation>
                                            </xs:annotation>
                                        </xs:element>
                                        <xs:element name="FormatPreference" minOccurs="0">
                                            <xs:annotation>
                                                <xs:documentation>Can be set to plain or
HTML</xs:documentation>
                                            </xs:annotation>
                                            <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                    <xs:enumeration value="Plaintext"/>
                                                    <xs:enumeration value="HTML"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="DateOfBirth">
                    <xs:annotation>
```

```
                    <xs:documentation>This element is used to describe the date when the person in quesiton was born.
</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="BirthDate" type="xs:string"/>
                        <xs:element ref="Restriction"/>
                        <xs:element name="Accuracy" minOccurs="0">
                            <xs:annotation>
                                <xs:documentation>If parts of the date are estimates, this can be added here. Possible
values are Day, MonthDay and YearMonthDay</xs:documentation>
                            </xs:annotation>
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:enumeration value="Day"/>
                                    <xs:enumeration value="MonthDay"/>
                                    <xs:enumeration value="YearMonthDay"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                        <xs:element name="Verification" minOccurs="0">
                            <xs:annotation>
                                <xs:documentation>Allows for the storage of the means by which the date was verified.
</xs:documentation>
                            </xs:annotation>
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="URL" type="xs:string">
                                        <xs:annotation>
                                            <xs:documentation>The URL of the list of documents allowable
for this verification.</xs:documentation>
                                        </xs:annotation>
                                    </xs:element>
                                    <xs:element name="Code" type="xs:string">
```

```
                                                    <xs:annotation>
                                                        <xs:documentation>The code for the verification of the date of
birth</xs:documentation>
                                                    </xs:annotation>
                                                </xs:element>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="PersonName">
                            <xs:annotation>
                                <xs:documentation>This element is used to describe a person's name with the flexibility to encapsulate names
from many different cultures. </xs:documentation>
                            </xs:annotation>
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="Type">
                                        <xs:annotation>
                                            <xs:documentation>Type of name can be current, former, alias or
other.</xs:documentation>
                                        </xs:annotation>
                                        <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                                <xs:enumeration value="Current"/>
                                                <xs:enumeration value="Alias"/>
                                                <xs:enumeration value="Former"/>
                                                <xs:enumeration value="Other"/>
                                            </xs:restriction>
                                        </xs:simpleType>
                                    </xs:element>
                                    <xs:element name="Alphabet" type="xs:string">
                                        <xs:annotation>
```

```
                                        <xs:documentation>Alphabet used in name, in ISO 15924 format.</xs:documentation>
                                    </xs:annotation>
                                </xs:element>
                                <xs:element name="GivenName" maxOccurs="unbounded">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element name="Name">
                                                <xs:annotation>
                                                    <xs:documentation>The actual text of the name to be
stored.</xs:documentation>
                                                </xs:annotation>
                                            </xs:element>
                                        </xs:sequence>
                                        <xs:attribute name="ID" use="required">
                                            <xs:annotation>
                                                <xs:documentation>Each name element must be given an unique
identifier number.</xs:documentation>
                                            </xs:annotation>
                                            <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                    <xs:enumeration value="yes"/>
                                                    <xs:enumeration value="no"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:attribute>
                                        <xs:attribute name="optionality" default="mandatory">
                                            <xs:annotation>
                                                <xs:documentation>Is the name optional, as, for example, English middle
names, or parts of Arabic names, such as "bin"</xs:documentation>
                                            </xs:annotation>
                                            <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                    <xs:enumeration value="mandatory"/>
                                                    <xs:enumeration value="optional"/>
```

```
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:attribute>
                                <xs:attribute name="initialable" default="yes">
                                    <xs:annotation>
                                        <xs:documentation>Can the name commonly be reduced to an
Initial.</xs:documentation>
                                    </xs:annotation>
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:enumeration value="yes"/>
                                            <xs:enumeration value="no"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:attribute>
                                <xs:attribute name="uncapitalized" type="xs:string">
                                    <xs:annotation>
                                        <xs:documentation>Is this part of the name written with an uncapitalized
letter?</xs:documentation>
                                    </xs:annotation>
                                </xs:attribute>
                                <xs:attribute name="linktoid" type="xs:integer">
                                    <xs:annotation>
                                        <xs:documentation>ID number of a name to which this name is linked.
This link is referenced in the following three attributes.</xs:documentation>
                                    </xs:annotation>
                                </xs:attribute>
                                <xs:attribute name="mandatorywithlink">
                                    <xs:annotation>
                                        <xs:documentation>with respect to the name referenced in linktoid, if the
referenced name is used, is this name also mandatory with it?</xs:documentation>
                                    </xs:annotation>
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
```

```
                                        <xs:enumeration value="yes"/>
                                        <xs:enumeration value="no"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:attribute>
                            <xs:attribute name="orderirrelevantwithlink">
                                <xs:annotation>
                                    <xs:documentation>with respect to the name referenced in linktoid, is the
ordering of this name the referenced name irrelevant. For example, in a hyphenated last name, is the order
important?</xs:documentation>
                                </xs:annotation>
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:enumeration value="yes"/>
                                        <xs:enumeration value="no"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:attribute>
                            <xs:attribute name="linkcharacter">
                                <xs:annotation>
                                    <xs:documentation>with respect to the name referenced in linktoid, is a
special character used to link the two names together? can be hyphenated, or no space. With no linkcharacter, a space is assumed.
</xs:documentation>
                                </xs:annotation>
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:enumeration value="hypenate"/>
                                        <xs:enumeration value="nospace"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:attribute>
                            <xs:attribute name="nickname" type="xs:string">
                                <xs:annotation>
```

```
                                        <xs:documentation>Is this name a nickname (i.e. no a legal name, but a name by which a person is commonly called.)</xs:documentation>
                                    </xs:annotation>
                                </xs:attribute>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="FamilyName" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="Name" type="xs:string"/>
                                </xs:sequence>
                                <xs:attribute name="ID" use="required">
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:enumeration value="yes"/>
                                            <xs:enumeration value="no"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:attribute>
                                <xs:attribute name="optionality" default="mandatory">
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:enumeration value="mandatory"/>
                                            <xs:enumeration value="optional"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:attribute>
                                <xs:attribute name="initialable" default="yes">
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:enumeration value="yes"/>
                                            <xs:enumeration value="no"/>
                                        </xs:restriction>
                                    </xs:simpleType>
```

```
                    </xs:attribute>
                    <xs:attribute name="uncapitalized" type="xs:string"/>
                    <xs:attribute name="linktoid" type="xs:integer"/>
                    <xs:attribute name="mandatorywithlink">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="yes"/>
                                <xs:enumeration value="no"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                    <xs:attribute name="orderirrelevantwithlink">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="yes"/>
                                <xs:enumeration value="no"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                    <xs:attribute name="linkcharacter">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="hypenate"/>
                                <xs:enumeration value="nospace"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                </xs:complexType>
            </xs:element>
            <xs:element name="Title" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Preceding title of name, for example Madam, Doctor,
etc.</xs:documentation>
                </xs:annotation>
```

```
                    <xs:complexType>
                        <xs:simpleContent>
                            <xs:extension base="xs:string">
                                <xs:attribute name="ID" type="xs:integer" use="required"/>
                            </xs:extension>
                        </xs:simpleContent>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Suffix" minOccurs="0" maxOccurs="unbounded">
                    <xs:annotation>
                        <xs:documentation>Suffix of name. Can include generational codes, (Junior, III)
academic qualifications (M.S.) or other name qualifiers.</xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                        <xs:simpleContent>
                            <xs:extension base="xs:string">
                                <xs:attribute name="ID" type="xs:integer" use="required"/>
                            </xs:extension>
                        </xs:simpleContent>
                    </xs:complexType>
                </xs:element>
                <xs:element name="FormatShortName">
                    <xs:annotation>
                        <xs:documentation>ID elements used to create the person's short name. Example for
most English-speaking people would be their first given name.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="FormatPreferredFormalName">
                    <xs:annotation>
                        <xs:documentation>ID elements used to create the person's preferrend (full) name.
Example may be Given Name + Last Name.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="FormatFullName">
```

```
                                        <xs:annotation>
                                                <xs:documentation>List of all elements that form a person's entire
name.</xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                                <xs:element name="FormatFamilyName">
                                        <xs:annotation>
                                                <xs:documentation>List of all elements that form a person's entire family
name.</xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="Identifier">
                <xs:annotation>
                        <xs:documentation>An Identifier is any serial number that, when taken in the context of other information
(such as the issuing institution) is a uniquely identifying attribute. This is the primary means of matching data records coming from
different sources. In order to reduce the value to identity thieves the actual serial number may be hashed.</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                        <xs:sequence>
                                <xs:choice>
                                        <xs:element name="governmentid">
                                                <xs:complexType>
                                                        <xs:sequence>
                                                                <xs:element name="type">
                                                                        <xs:annotation>
                                                                                <xs:documentation>Type can be passport (P), identity card
(ID), birth certificate (BC), driver's licence (DL) or tax identification number (TAX). Not all of these types are appropriate for all
countres. </xs:documentation>
                                                                        </xs:annotation>
                                                                        <xs:simpleType>
                                                                                <xs:restriction base="xs:string">
```

```
                                                    <xs:enumeration value="BC"/>
                                                    <xs:enumeration value="DL"/>
                                                    <xs:enumeration value="P"/>
                                                    <xs:enumeration value="TAX"/>
                                                    <xs:enumeration value="ID"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:element>
                                        <xs:element name="issuingAuthority">
                                            <xs:annotation>
                                                <xs:documentation>The issuing authority is either the country code of the issuing identity document  in ISO 3166-1 format, or the region issuing the document in ISO 3166-2 format. This allows for the capture of both national documents (for example, a passport) as well as locally issued documents (a state ID card).</xs:documentation>
                                            </xs:annotation>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="institutionid">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="IssuerIdentifier" type="xs:string">
                                            <xs:annotation>
                                                <xs:documentation>The issuer identifier is the base URL of the institution that issues the identifier. For examples - gwu.edu ; man.ac.uk etc.</xs:documentation>
                                            </xs:annotation>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:choice>
                        <xs:element name="validfrom" type="xs:date" minOccurs="0"/>
                        <xs:element name="validto" type="xs:date" minOccurs="0"/>
```

```
                    <xs:element name="version" type="xs:string">
                        <xs:annotation>
                            <xs:documentation>The version element is always set to 1.0 at this point. It is included for future expansion of the system.</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element name="serial">
                        <xs:annotation>
                            <xs:documentation>The serial is the actual serial number of the document that is being used (for example, the social security number. The serial may be stored as plain text, or, if the hash attribute is set, to a hash value of the seed followed by the serial, in the format specified in the hash attribute (currently only MD6). The serial, whether hashed or in plaintext should contain no punctuation, unless that punctuation is part of the serial itself. For example, a U.S. social security number should be stored as NNNNNNNNN and not NNN-NN-NNNN. </xs:documentation>
                        </xs:annotation>
                        <xs:complexType>
                            <xs:simpleContent>
                                <xs:extension base="xs:string">
                                    <xs:attribute name="hash" use="optional">
                                        <xs:annotation>
                                            <xs:documentation>If the serial is to be stored as a hash value, this field shows the type of hash algorithm used. MD6 is the current option, but it is left open for future changes.</xs:documentation>
                                        </xs:annotation>
                                        <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                                <xs:enumeration value="MD6"/>
                                            </xs:restriction>
                                        </xs:simpleType>
                                    </xs:attribute>
                                    <xs:attribute name="salt" type="xs:string" use="optional">
                                        <xs:annotation>
                                            <xs:documentation>A salt  to be used with the hashing algorithm. This might be a random set of characters, or the person's last name, etc. This is mandatory if a hashing algorithm is used.</xs:documentation>
```

```
                                                                    </xs:annotation>
                                                                </xs:attribute>
                                                            </xs:extension>
                                                        </xs:simpleContent>
                                                    </xs:complexType>
                                                </xs:element>
                                                <xs:element ref="Restriction"/>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                    <xs:complexType name="PersonIdentity">
                                        <xs:sequence>
                                            <xs:element ref="Identifier" maxOccurs="unbounded"/>
                                            <xs:element ref="PersonName" maxOccurs="unbounded"/>
                                            <xs:element ref="DateOfBirth"/>
                                            <xs:element ref="Address" maxOccurs="unbounded"/>
                                            <xs:element ref="ElectronicContact"/>
                                        </xs:sequence>
                                    </xs:complexType>
</xs:schema>
```

Bibliography

Farmer, Jim. (**2010: 12 April**). *Report from a Meeting of Chief Technologists: Harmonizing Web Services Data Transport Standards for Teaching, Learning, and Support of Students*. Washington DC: Georgetown University. Available from www.immagic.com/eLibrary/ARCHIVES/GENERAL/IMM/I100412F.pdf

Dematteis, Kerry. Lutz, Richard. McCallum-Bayliss, Heather (**1998**) *Whose Name Is It: Names, Ownership and Databases.*Herndon: VA. Available from http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/LAS_US/L980000A.pdf

U.S. Department of Education. National Center for Education Statistics. (**2001**) *Student Data Handbook: Elementary, Secondary, and Early Childhood Education: 2001 Update*. NCES 2000-343 (Revised). Project Officer: Beth Young. Washington, DC.

Postsecondary Electronic Standards Council (**2009**) *Implementation Guide to the Postsecondary Electronic Standards Council XML Standard Format for the College Transcript Version 1.1.1.A* Washington, DC. Available from http://www.pesc.org/library/docs/standards/College%20Transcript/XML%20College%20Trsc%20Impl%20Guide%202009%2010%2027%20V%201.1.1.A.pdf

Schools Interoperability Framework Association (**2009**) *Schools Interoperability Framework™ Implementation Specification 2.3*. Washington D.C. Available From http://specification.sifinfo.org/Implementation/2.3/

Kuali Foundation (**2010**) *Kuali Rice Global Technical Reference Guide*. Available from http://kuali.org/rice/documentation/1.0.1.1/TRG_Global/

International Organization for Standardization(**2006**) *Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes.*Genève, CH.

International Organization for Standardization **(2007)** *Codes for the representation of names of countries and their subdivisions -- Part 2: Country subdivision code.* Genève, CH.

International Organization for Standardization **(2004)** *Information technology -- Codes for the representation of human sexes*. Genève, CH.

Data Quality Campaign (**2006: 7 November**). *Creating a Longitudinal Data System: Using Data To Improve Student Achievement*. Washington DC USA. Available from www.immagic.com/eLibrary/ARCHIVES/GENERAL/DATAQ_US/D061107C.pdf

International Telecommunication Union (**2005**) *ITU-T Recommendation E.164*. Genève, CH.

Internet Engineering Task Force **(2008)** RFC5321 : *Simple Mail Transfer Protocol*. Fremont, CA.

Internet Engineering Task Force (**2001**) RFC2822 : *Internet Message Format*. Fremont, CA.

Arizona State Legislature (**2007**) *Identification Numbers, Social Security Numbers*, ARS 15-1823. Phoenix, AZ

Arizona State Legislature (**2003**) *Confidentiality of Social Security Numbers, ARS 44-1376*. Phoenix, AZ

NIEM Technical Architecture Committee (**2008**) *NIEM High-Level Version Architecture*. Washington, DC

HR-XML Consortium (**2008**) *HR-XML 3.0 Library*. Chicago, IL

MedBiquitous Professional Profile Working Group (**2008**) *MedBiquitous Healthcare Professional Profile Implementation Guidelines Version 1.0.*Baltimore, MD

PESC Postsecondary Electronics Standards Council. (**2002:26 February**). *Agreement on the Use of Standards in Electronic Exchanges Support Federal Financial Aid Between the US Department of Education's Office of Student Financial Assistance and the Postsecondary Electronic Standards Council.* Washington DC USA: Postsecondary Electronics Standards Council. www.immagic.com/eLibrary/ARCHIVES/GENERAL/PESC/P020226.pdf

Masa, Javi (**2009**) *Schema for Academia: Attribute Definitions for Individual Data*. Amsterdam, NL

Nagrodzki, Rafał (**2009**) *The Mobility Project*. Warsaw, PL:University of Warsaw. Available at:http://usos.edu.pl/Mobility/nagrodzki2009-12-03.pdf