Customer Information Quality (CIQ) Specifications Version 3.0 – Technical Overview

Public Review Draft 02

15 June 2007

Specification URIs:

This Version:
docs.oasis-open.org/ciq/v3.0/prd02/supp/ciq-technical-overview-v3-prd2.html
docs.oasis-open.org/ciq/v3.0/prd02/supp/ciq-technical-overview-v3-prd2.doc
docs.oasis-open.org/ciq/v3.0/prd02/supp/ciq-technical-overview-v3-prd2.pdf

Previous Version:
N/A

Latest Version:
docs.oasis-open.org/ciq/v3.0/supp/ciq-technical-overview-v3.html
docs.oasis-open.org/ciq/v3.0/supp/ciq-technical-overview-v3.doc
docs.oasis-open.org/ciq/v3.0/supp/ciq-technical-overview-v3.pdf

Technical Committee:
OASIS Customer Information Quality TC

Chair(s):
Ram Kumar (kumar.sydney@gmail.com)

Editor(s):
Ram Kumar (kumar.sydney@gmail.com)

Related work:
This version of the CIQ specifications replaces or supercedes:

- OASIS CIQ extensible Name Language (xNL) V2.0 Committee Specification
- OASIS CIQ extensible Address Language (xAL) V2.0 Committee Specification
- OASIS CIQ extensible Name and Address Language (xNAL) V2.0 Committee Specification
- OASIS CIQ extensible Customer Information Language (xCIL) V2.0 Committee Specification

Abstract:

This technical overview document provides a quick practical introduction into high level technical details of CIQ TC specification family version 3.0.

Status:

This document was last revised or approved by the OASIS CIQ TC on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at www.oasis-open.org/committees/ciq.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (www.oasis-open.org/committees/ciq/ipr.php.

The non-normative errata page for this specification is located at www.oasis-open.org/committees/ciq.

Notices

Table of Contents

1 Introduction

This document is a brief technical overview of version 3.0 of OASIS CIQ TC specifications family namely:

- **xNL** : extensible Name Language
- **xAL**: extensible Address Language
- **xNAL**: extensible Name and Address Language (combines xNL and xAL)
- **xPIL:** extensible Party Information Language (formerly known as extensible Customer Information language (**xCIL**)
- **xPRL**: extensible Party Relationships Language (formerly known as extensible Customer Relationships Language (**xPRL**) – Release data for this specification not set yet

The purpose of this document also is to give software developers and solution architects a quick snapshot of CIQ TC specifications and help decide if the specifications are suitable for a particular application.

2 CIQ TC Family Version 3.0

2.1 The Need for a New Version

The CIQ TC's XML Name and Address languages define universal structures for name and address entities.

It is a trivial exercise to define name and address structures for a particular locale, but on the international scale it is much harder due to cultural and lingual differences. Previous versions of xNAL defined the name and address structures to a great level of detail providing very hierarchical XML structures to express names and addresses in a consistent way.

However, the previous versions were:

- ambiguous by providing multiple options for representing the same information
- offering a complex model for simple representation of name and address data
- difficult to implement as an object model
- perceived as being complex for many applications that required minimal representation
- semantically incorrect for many country name and address data that are bound by its culture and geographical boundaries

In many cases the xNAL family of specifications were used as a basis for a localized standards that were much simpler, but not truly interoperable on a global scale. The derived standards were mainly about scaling it down to a simpler and lighter version that would meet the local requirements.

CIQ TC recognized the need for simplifying the specifications while keeping them locale-independent and interoperable on a global scale, and importantly, ensuring that the capabilities of the earlier versions are not compromised.

2.2 What is in scope in this version

- Ensure all the overall expressive power of version 2.0 is not lost
- The specification will include W3C XML schemas
- All examples defined using version 2.0 will be represented in version 3.0
- High level UML models of the schemas

2.3 What is out of scope in this version

- DTDs
- Privacy and security issues connected to exchanging and storing personal information
- Data exchange methods and procedures for party information
- Messaging protocol for exchange of party information
- Validation/verification of party information
- Formatting, labeling, or sorting of party information
- API specifications
- Backward compatibility with previous versions

3 CIQ TC Specifications Version 3.0

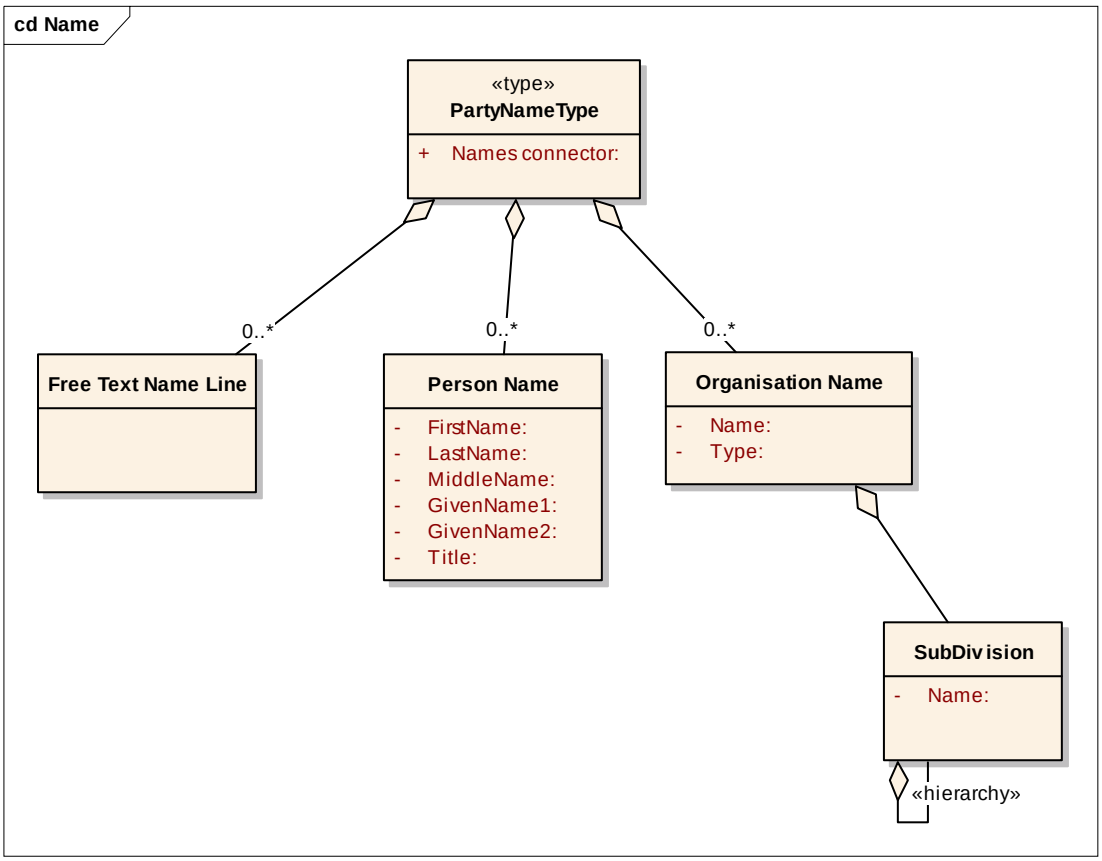
This section provides a brief overview of the CIQ TC specifications (Version 3.0).

3.1 Extensible Name Language (xNL)

xNL defines an XML structure to represent party name data. An example of a Party is “customer”. A party could be a “Person” or an “Organization”. An “Organization” could be educational institutions namely, school, university or college, clubs, associations, industry groups, not-for-profit bodies, consortiums, etc.

xNL was designed to handle international name data that is culturally and geographically specific. For example, the concept of given names and family names do not exist in some cultures, e.g. in some regions of India.

xNL can represent names in over 36 formats and it is extendable. The diagram below illustrates a high level UML model of xNL.



3.1.1 Example – Simple Person Name Representation

Dr Jeremy Apatuta Johnson III PhD

```
<n:PartyName>
   <n:PersonName>
          <n:NameLine>Dr Jeremy Apatuta Johnson III PhD</n:NameLine>
   </n:PersonName>
</n:PartyName>
```

3.1.2 Example – Complex Person Name Representation

Dr Jeremy Apatuta Johnson III Phd

```
<n:PartyName>
   <n:PersonName>
          <n:NameElement Abbreviation="true" ElementType="Title">Dr</n:NameElement>
          <n:NameElement ElementType="FirstName">Jeremy</n:NameElement>
          <n:NameElement ElementType="MiddleName">Apatuta</n:NameElement>
          <n:NameElement ElementType="LastName">Johnson</n:NameElement>
          <n:NameElement ElementType="GenerationIdentifier">III</n:NameElement>
          <n:NameElement ElementType="Title">PhD</n:NameElement>
   </n:PersonName>
</n:PartyName>
```

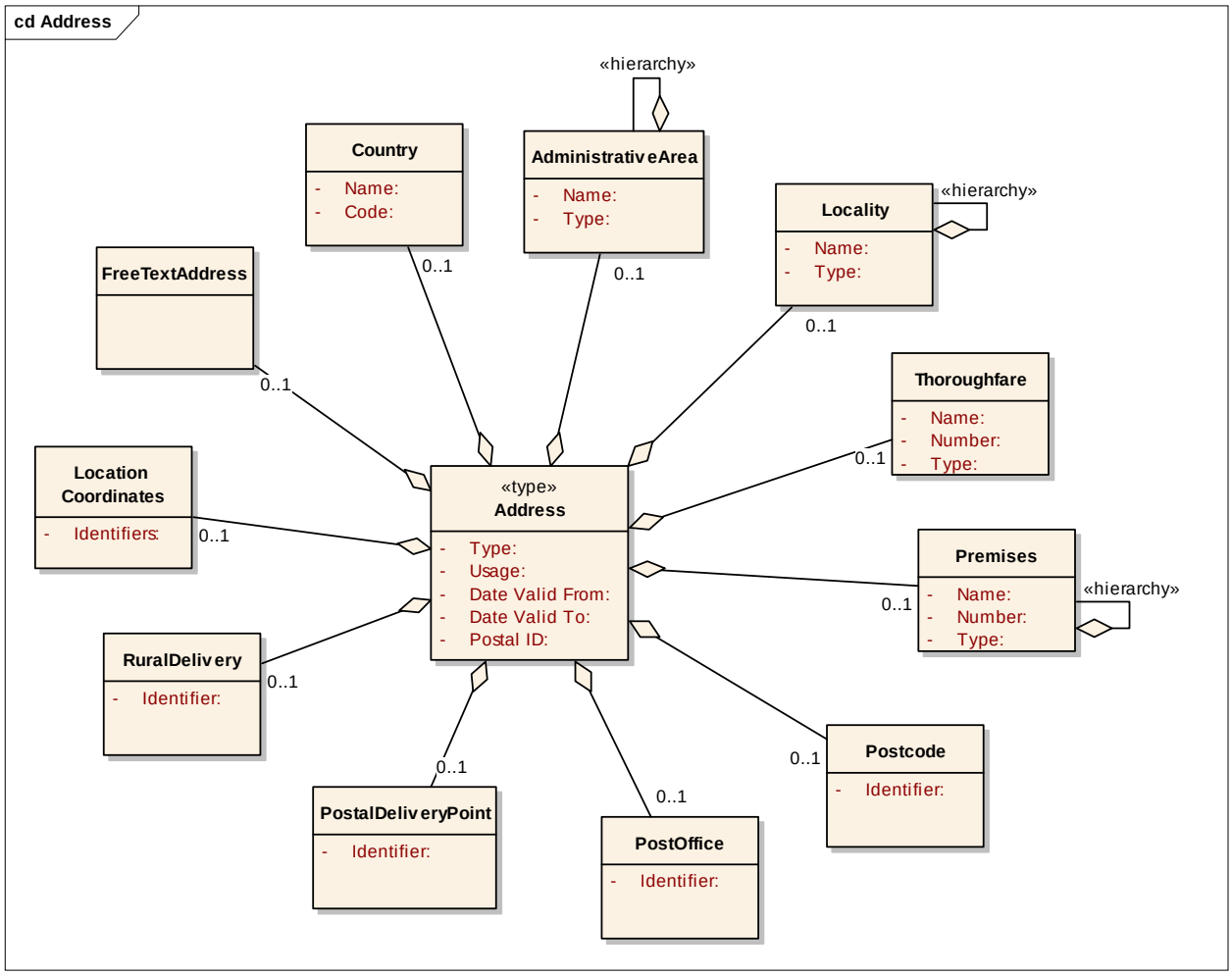
3.2 Extensible Address Language (xAL)

xAL defines an XML structure to represent address data. An address could include but not limited to any of the following types that are supported by xAL:

- Airport
- Business/Commercial Parks
- Caravan Parks
- Community Developments
- Dual (Primary and Secondary)
- Educational institutions
- Entertainment/Recreation Parks
- Hospitals
- Large Mail Users
- Marinas
- Military
- Ports
- Retirement Villages
- Resorts
- Royal Highness
- Rural (with land, air and water access)
- Sporting Venues
- Territories
- Tribal
- Simple Urban
- Complex Urban
- Utility Urban
- Ranged Urban
- Villages
- Canals
- Banks

- etc

xAL can represent addresses of 245+ countries in over 130 formats. The diagram below illustrates a high level UML model of xAL.



3.2.1 Example – Simple Address Representation

16 Patterson Street, OCEAN REEF, WA

```
<a:Address>
   <a:FreeTextAddress>
      <a:AddressLine>16 Patterson Street</a:AddressLine>
      <a:AddressLine>OCEAN REEF</a:AddressLine>
      <a:AddressLine>WA</a:AddressLine>
   </a:FreeTextAddress>
</a:Address>
```

3.2.2 Example – Semi Complex Address Representation

16 Patterson Street, OCEAN REEF, WA

```
<a:Address>
   <a:FreeTextAddress>
      <a:AddressLine>16 Patterson Street</a:AddressLine>
   </a:FreeTextAddress>
   <a:AdministrativeArea a:Type="State">
      <a:NameElement>WA</a:NameElement>
   </a:AdministrativeArea>
   <a:Locality a:Type="Suburb">
      <a:NameElement>OCEAN REEF</a:NameElement>
   </a:Locality
</a:Address>
```

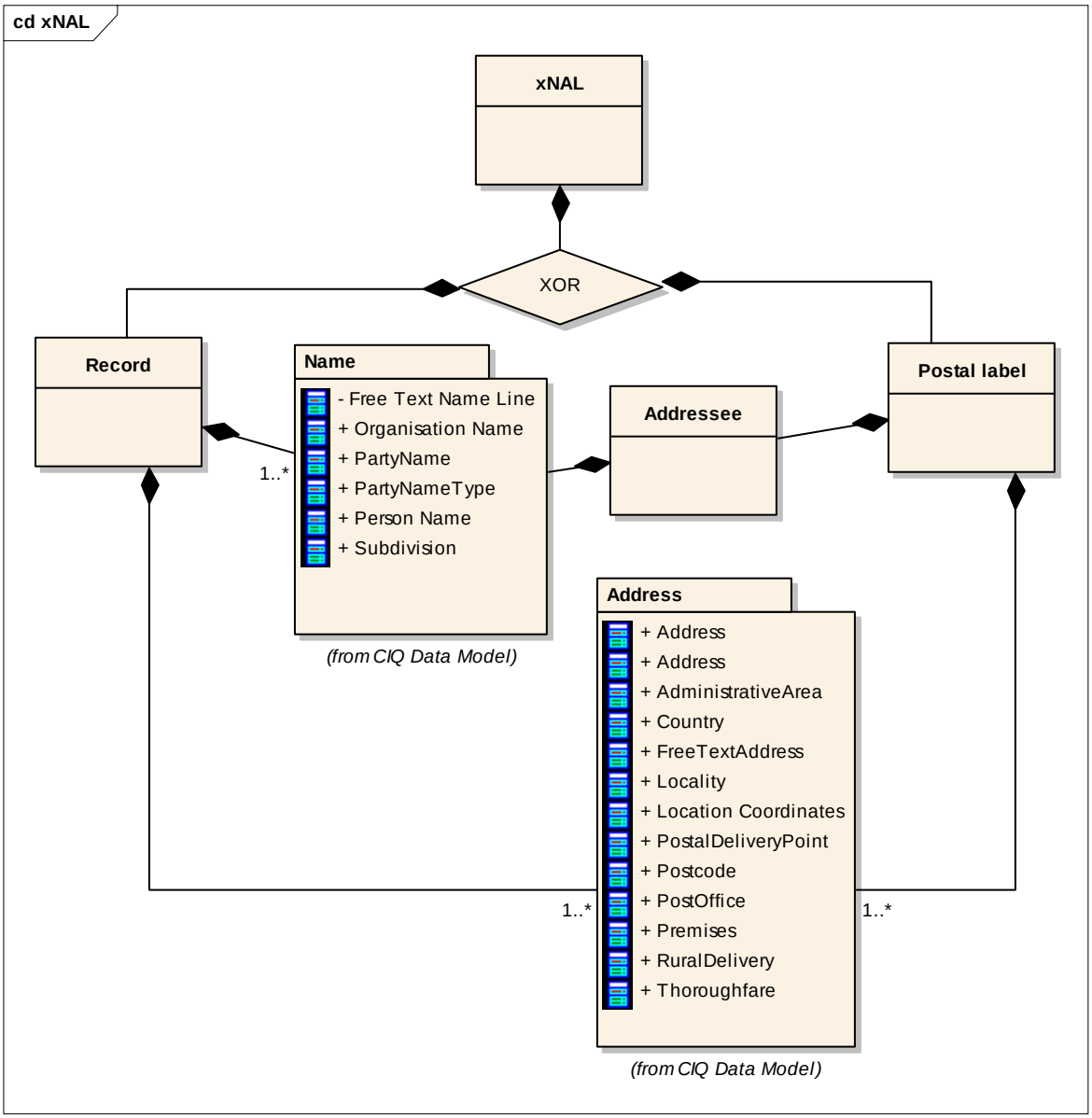
3.2.3 Example – Complex Address Representation

16 Patterson Street, OCEAN REEF, WA

```
<a:Address>
   <a:AdministrativeArea a:Type="State">
          <a:NameElement>WA</a:NameElement>
   </a:AdministrativeArea>
   <a:Locality a:Type="Suburb">
          <a:NameElement>OCEAN REEF</a:NameElement>
   </a:Locality>
   <a:Thoroughfare a:Type="Street">
          <a:Name>Patterson</a:Name>
          <a:Number>16</a:Number>
   </a:Thoroughfare>
</a:Address>
```

3.3 Extensible Name and Address Language (xNAL) Version 3.0

xNAL defines an XML strucutre to represent name and address data bound together. xNAL utilizes XML structures from xNL and xAL specifications. The diagram below illustrates a high level UML model of xNAL version 3.0.



3.3.1 Example Simple Name and Address Representation

Mr H G Guy, 9 Uxbridge Street, Redwood, Christchurch

```
<nal:Record>
   <n:PartyName>
      <n:NameLine>Mr H G Guy</n:NameLine>
   </n:PartyName>
   <a:Address>
       <a:FreeTextAddress>
          <a:AddressLine>9 Uxbridge Street</a:AddressLine>
          <a:AddressLine>Redwood</a:AddressLine>
          <a:AddressLine>Christchurch</a:AddressLine>
       </a:FreeTextAddress>
   </a:Address>
</nal:Record>
```

3.3.2 Example – Complex Name and Address Representation

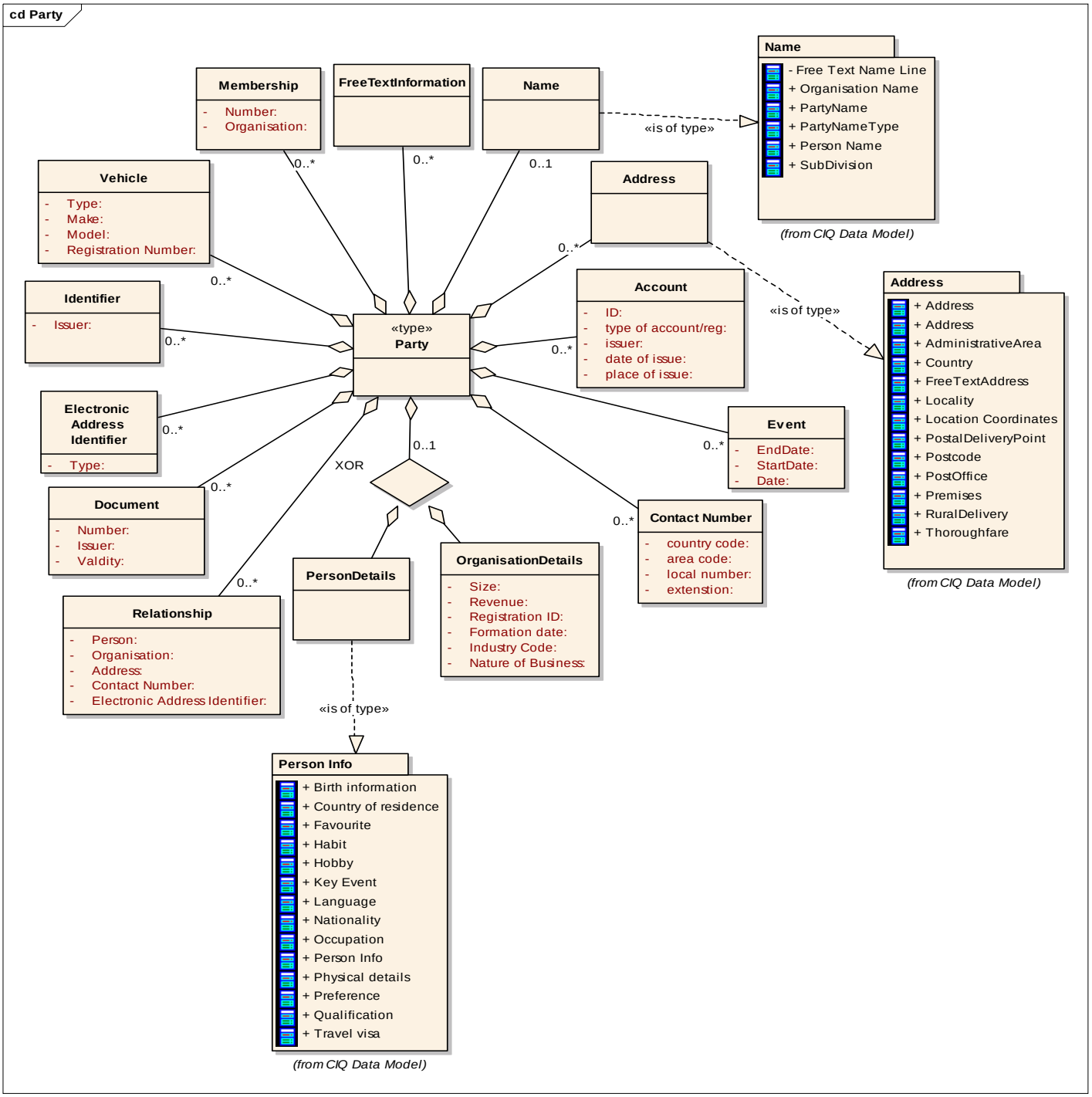
Mr H G Guy, 9 Uxbridge Street, Redwood, Christchurch

```
<xnal:Record>
   <n:PartyName>
      <n:PersonName>Mr H G Guy
          <n:NameElement n:ElementType=”Title>Mr</n:NameElement>
          <n:NameElement n:ElementType=”FirstNameInitial”>H</n:NameElement>
          <n:NameElement n:ElementType=”MiddleNameInitial”>G</n:NameElement>
          <n:NameElement n:ElementType=”LastName”>Guy</n:NameElement>
       </n:PersonName>
   </n:PartyName>
   <a:Address>
       <a:AdministrativeArea>
          <a:NameElement>Christchurch</a:NameElement>
       </a:AdministrativeArea>
       <a:Locality>
          <a:NameElement>Redwood</a:NameElement>
       </a:Locality>
       <a:Thoroughfare>>
          <a:Name>Uxbridge Street </a:Name>
          <a:Number>9</a:Number>
       </a:Thoroughfare>
   </a:Address>
</xnal:Record>
```

3.4 Extensible Party Information Language (xPIL)

xPIL defines an XML structure to represent party-centric data. Party-centric data includes name, address, e-mail address, telephone numbers, identification details (e.g. passport, license number, identification card, etc), vehicle details, account details, etc. These unique attributes of a party assist in uniquely identifying a party. The diagram below illustrates a high-level UML view of xPIL version 3.0

.



3.5 Extensible Party Relationships Language (xPRL)

xPRL defines a consistent way of using xLink to represent party relationships. Party relationships could be:

- Person to Person relationships
- Person to Organization relationships, and
- Organization to Organization relationships

Release date for version 3.0 of this specification not set yet.

4 Implementing CIQ TC specifications – Practical Guidelines

Some readers may find it hard to get to grips with the CIQ TC specifications family. This section is an informative guide to help you get started.

4.1 Where to Start

Consider doing the following:

- Clearly define your requirements and goals of using CIQ Specifications
- Complete reading this document (15 minutes)
- Study the XML examples of the schemas (30 minutes). Examples are provided in the same download as the schemas.
- Study the schema diagrams (15 minutes). You can browse the schemas using an XML editor or use HTML documentation provided as part of every CIQ TC specification
- Try to build valid structures you need using the schemas and your sample data (20 minutes). You may want to use an XML editor that provides information from schema xs:annotation elements to help you understand the meaning of the elements and attributes.
- Understanding the OASIS Codelist specification and how to use the specification as part of CIQ Specifications (provided as an option) could be time consuming, but a worthy exercise. Enough work has already been done by the TC to keep this process simple by providing all required files and test cases in the CIQ Specification package.
- If you want to customize the base CIQ schemas without touching/modifying them to meet your application specific requirements, use Schematron patterns as part of the UMCLVV approach used by OASIS Code List Specification. To be able to use this option, you need to have some basic knowledge of xPath and Schematron languages.

4.2 Don't get confused – keep it simple

xNL, xAL and other CIQ TC specifications provide the flexibility to deal with different types of applications. Flexibility could lead to breaking interoperability unless the implementation is managed effectively. If you are interoperating the data with other parties, ensure that you and the other parties implement the specifications in identical fashion agreement is managed.

Version 3.0 allows you to customize the specifications to meet your requirements without affecting the structure of the schemas through enumeration lists. However, please ensure that what you have customized is agreeable with the other party that exchanges data with you (e.g. applications, end users, external parties) to achieve interoperability between parties involved in data exchange.

4.3 Data Exchange

CIQ TC specifications can be used to organize data exchange of party information or just names and addresses. It is likely that CIQ TC specifications on their own are not enough to organize such an exchange as it requires some messaging mechanisms and additional information such as metadata.

CIQ TC recommends that reusable elements from the CIQ TC schemas are used inside other namespaces or wrappers. This will ensure that the original namespaces remain intact while additional information is still provided.

CIQ TC re-iterates here that agreements should be in place between parties involved in the data exchange process on how the specifications will be implemented to ensure consistency in implementations and how the agreement will be managed/ governed. This is very important to achieve interoperability of data between parties involved in data exchange.

Given that CIQ Specifications provide many optional elements and attributes, implementation of the specifications for data exchange require agreement in place between parties that use the CIQ specifications based data formats to ensure interoperability.

4.4 Output Formatting

CIQ TC specifications do not have any means to specify the formatting of the data. It is up to the application to decide which formatting suits best. It is recommended to preserve the original order of elements to assist with correct output formatting. Remember, that addresses, for example, may begin with the finest details (e.g. flat number) in some locales or with country name in the other. Preserving the original order is important.

4.5 Customizing Schema

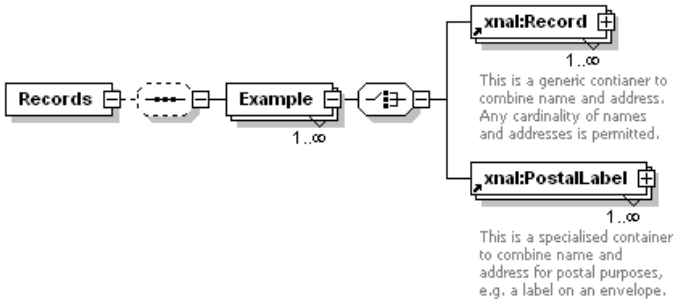
CIQ XML Schemas (*xNL.xsd, xAL.xsd, xNAL.xsd*, and *xPIL.xsd*) have been designed to be application and industry independent thereby allowing different applications to use them. Users have been provided with the following choices to customize CIQ Schema to meet their specific application requirements.

Further details on this subject are described in “Name, Address and Party Specifications Document” of CIQ TC.

4.5.1 Schema Extensions

It is possible to extend CIQ XML schemas within some allocated boundaries to meet specific application or locale requirements. The extensions can be of four types:

- Any element can have any number of attributes from the non-target namespace, which means you can include some other attributes not specified by the schema.
- Enumerations can be changed and they are intentionally placed in a separate “include” xml schema file.
- Enumerations can also be changed with genericode approach from OASIS Code List Representation Technical Committee and the enumeration lists are placed in separate files (.gc extension)
- Adding new elements to the schema is not permitted to ensure interoperability– use wrappers instead. This is shown in the figure below the elements of xNAL are wrapped using an XML Schema that has “Records” as its root element.



4.5.2 Restricting Schema

Restricting the use of the CIQ XML Schemas as part of implementation can be done by two ways:

- All elements and attributes in the CIQ XML Schemas are optional. This provides users the flexibility to customise the schemas to meet their application specific requirements.
- Deleting new elements in the CIQ XML schemas are not permitted to ensure interoperability – use UMLCVV (approach from OASIS Code List Representation TC and OASIS UBL TC) that is proposed to restrict the schema without modifying it. This allows customisation of the schema by defining business rules using Schematron language, an open industry standard, to meet application specific requirements, but at the same time ensures that the XML document is compatible with the base/core

schema. This capability for example, allows *xAL* schema to be customised to meet country specific address structure requirements. An example would be a country like *Singapore* where there are no states, cities, post towns and Rural Areas. In this case, a business rule can be written not to use *AdministrativeArea, SubAdministrativeArea, PostTown*, and *RuralDelivery* elements of *xAL.xsd* schema.

A working example of this is provided as part of the CIQ V3.0 package.

4.6 Data Mapping Challenges

The main challenge in standardising name and address and even party data structures is in a potentially infinite number of ways they can be presented for different applications, different cultures and locales.

4.6.1 Application Diversity

For example a simple e-commerce database may have name as one field, address as a free-text 3-field set and other party information in a dozen of other fields. It may be sufficient for that particular data usage scenario.

A larger bank may be interested in a more detailed name and address structure to allow business intelligence applications to do their analysis.

The differences in complexity between these two examples present a great challenge finding a common form of representing the data so that it is attractive to all parties participating in data exchange.

4.6.2 Cultural Diversity

Name and address presentation formats vary between cultures elevating the importance of breaking down the structure and preserving the original meaning of the elements so that the name or address can be correctly restored at a later time. It is virtually impossible to fit all these diverse views into a single name and address specification that is also specific to a particular culture. Some balanced approach is required to meet the semantic and presentation variations and requirements in one specification. It is the goal of CIQ TC to achieve such a balance.

India is a good example of cultural diversity with people from different ethnic backgrounds, languages (officially 14 national languages) and religions. In some Indian locales there is no concept of family name or given name or surname or first name or middle name or last name. They have the following name types that can be used as part of a person's name:

> *Grand father name, Great grand father name, Father's name, Mother's name, Native Place name, Tribal name, Caste name, Husband's name, Birth name*, etc.

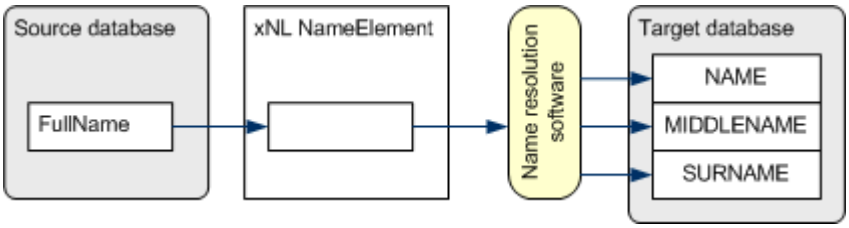
Addresses are culture and locale specific too. There is typically a great degree of freedom as to how one writes an address with information that is specific to the geographic location/locale. Yet it still reaches the destination. For example, in countries like Thailand, addresses include the names of the river banks, or canals instead of streets. The concept of neither the postal code nor the locality applies to some countries. In certain countries an address is attached to the number of a postal van that delivers the mail to the destination as the van is responsible for delivering mail to a certain area/streets in an area.

4.6.3 CIQ TC Solution

CIQ TC provides a solution that can absorb and persist with the information in the form it was originally provided without any loss of semantics. The information can then be mapped to some target structure with a minimal effort.

However, CIQ TC does not provide a solution for mapping a simple source structure to a more complex target as it would require parsing and "understanding" the information carried in the structure itself. Any solution to this problem is out of scope for CIQ TC.

The diagram below shows how a simple one-field data model can be mapped to another complex data model through xNL, but with help of "smart name parsing/scrubbing data quality software" (and there are plenty in the market) to separate a *full name* into *name*, *middle name* and *surname*:



A. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

John Glaubitz	Vertex, Inc	Member, CIQ TC
Max Voskob	Individual	Former Member, CIQ TC
Hidajet Hasimbegovic	Individual	Member, CIQ TC
Robert James	Individual	Member, CIQ TC
Joe Lubenow	Individual	Member, CIQ TC
Mark Meadows	Microsoft Corporation	Former Member, CIQ TC
John Putman	Individual	Former Member, CIQ TC
Michael Roytman	Vertex, Inc	Member, CIQ TC
Colin Wallis	New Zealand Government	Member, CIQ TC
David Webber	Individual	Member, CIQ TC
Graham Lobsey	Individual	Member, CIQ TC
George Farkas	XBI Software, Inc	Member, CIQ TC

OASIS CIQ Technical Committee (TC) also wishes to acknowledge contributions from former members of the TC since its inception in 2000. Also, the TC would like to express its sincere thanks to the public in general (this includes other standard groups, organizations and end users) for their feedback and comments that helped the TC to improve the CIQ specifications.

Special thanks to Mr.Hugh Wallis, Director of Standards Development of extensible Business Reporting Language (xBRL) International Standards Group (http://www.xbrl.org) for working closely with the CIQ TC in jointly implementing W3C xLink specification that is now used by both xBRL and CIQ Specifications to enable interoperability between the two specifications.

Special thanks to Mr.Carl Reed, Chief Technology Officer of Open Geospatial Consortium (OGC – http://www.opengeospatial.org) for his guidance and assistance to the TC in referencing the work of OGC on GeoRSS and Geo-Coordinates for addresses/locations as part of CIQ Address Specifications.

Special thanks to Mr.Ken Holman, Chair of OASIS Code List TC (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=codelist) for his assistance to the TC in releasing the OASIS Code List version of CIQ V3.0 XML Schemas.

Last but not least, the TC thanks all users of the CIQ TC specifications in real world and for their continuous feedback and support.

B. Intellectual Property Rights, Patents, Licenses and Royalties

CIQ TC Specifications (includes documents, schemas and examples[1 and 2]) are free of any Intellectual Property Rights, Patents, Licenses or Royalties. Public is free to download and implement the specifications free of charge.

[1]**xAL-Australia.XML**

Address examples come from AS/NZ 4819:2003 standard of Standards Australia and are subject to copyright

[2]**xAL-International.xml**

Address examples come from a variety of sources including Universal Postal Union (UPU) website and the UPU address examples are subject to copyright.

xLink-2003-12-31.xsd

This schema was provided by the xBRL group in December 2006.

C. Revision History

Revision	Date	Editor	Changes Made
V3.0 PRD 01	13 April 2006	Ram Kumar and Max Voskob	Prepared 60 days public review draft from Committee Draft 01
V3.0 PRD 02	15 June 2007	Ram Kumar	Prepared second round of 60 days public review draft from Committee Draft 02 by including all public review comments from PRD 01. Also included is implementation of OASIS Code list specification