Election Results Common Data Format Specification

Revision 2.0

Mark Cwalinski
Sam Dana
John Dziurlaj
Jared Marcotte
Justin Moore
John Wack
Sarah Whitt

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

NIST Special Publication 1500-100

Election Results Common Data Format Specification

Revision 2.0

Mark Cwalinksi
Justin Moore
Google, Inc.

Sam Dana
Prometheus Computing

John Dziurlaj
Democracy Fund

Jared Marcotte
The Turnout

John Wack
Software and Systems Division
Information Technology Laboratory, NIST

Sarah Whitt
Wisconsin Elections Commission

U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Undersecretary of Commerce for Standards and Technology

National Institute of Standards and Technology (NIST) Special Publication 1500-100,
Revision 2.0
146 pages (December 2019)

NIST Special Publication series 1500 is intended to capture external perspectives related to NIST standards, measurement, and testing-related efforts. These external perspectives can come from industry, academia, government, and others. These reports are intended to document external perspectives and do not necessarily represent official NIST positions.

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. This document reports on ITL's research, guidance, and outreach efforts in Information Technology and its collaborative activities with industry, government, and academic organizations.

Abstract

This publication describes an election results common data format specification for pre-election setup information and post-election results reporting. It contains a UML (Unified Modeling Language) model of the election data and an XML (eXtensible Markup Language) and JSON (JavaScript Object Notation) format derived from the UML model. It also contains background information regarding how geopolitical geography is structured and used in the model and schema. The XML format is comprehensive and at the same time very flexible, able to accommodate election scenarios used throughout the U.S. It is part of a series of planned common data format specifications for voting equipment.

Keywords

Ballot; common data format; contest; district; election results; jurisdiction; overvote; political office; political party; precinct; referendum; undervote, voting.

Acknowledgements

Executive Summary

This publication presents a common data format (CDF) for pre-election setup information and post-election results reporting. The format, known as the Election Results Common Data Format Specification Version 2.0, is comprehensive and detailed in its coverage of election results-related data and at the same time very flexible, able to accommodate election scenarios used throughout the United States. This publication describes:

- a UML (Unified Modeling Language) model [1],
- derived XML (eXtensible Markup Language) and JSON (JavaScript Object Notation) schemas [2][3],
- usage information and guidance, and
- background information.

This specification provides a common data interchange format for election data used in voting systems across U.S. jurisdictions. Using this specification, pre-election and post-election data can be published in a common, well-understood format. The format accommodates highly detailed election results data and is sufficiently flexible to accommodate many different types of contests and political structures.

This specification provides manufacturers of election management systems (EMS) and managers of election jurisdictions with standard methods for importing and exporting election data, thereby increasing interoperability among election devices and reducing the need to create software to translate between proprietary formats. Interoperable data will reduce costs to election jurisdictions by reducing the complexity in election management and offering jurisdictions more choice in election equipment.

This specification is geared towards the following audiences:

- Election officials
- Voting equipment manufacturers
- Election-affiliated organizations and
- Election analysts and the general public

The format accommodates three different election scenarios:

Pre-election. The period prior to an election, for reporting pre-election data from a jurisdiction but not yet complete information about any election.

Election. The period during which an election is being conducted and election results reports are produced. The reports include aggregated results data or more detailed, precinct-level reporting, depending on the capabilities of the reporting jurisdiction.

Post-election. The period after the polls close when more detailed election results reports are produced with options for precinct reporting, type of ballot, and type of device.

The XML and JSON schemas associated with this specification are derived from the UML model, which defines the types, structure, and interrelationships of geopolitical geography across the United States. The model was designed to accommodate multiple types of contests and their many variations.

It also provides the capability to report on these contests from higher aggregate levels down to very fine levels of detail, including:

- reporting by precincts and split precincts;
- reporting by ballot type, for example, absentee and election day; and
- reporting by device type and specific voting device.

The UML model can be re-used and modified to meet the needs of other planned common data format specifications for voting devices such as electronic pollbooks and ballot marking devices.

Notice of Revision 2.0

This document is a version 2.0 revision to the NIST Special Publication 1500-100, Election Results Common Data Format Specification, Version 1. Changes were made to the version 1 UML model to add additional information and make the model easier to implement and use. Additionally, the XML schema was updated accordingly and a new JSON schema was generated.

The complete list of changes is contained in Appendix E. Following is a summary:

- Various revisions were made to the Executive Summary and Sections 1, 2, and 3 to improve the presentation and to accomodate version 2 changes.
- Section 4, XML Schema documentation was replaced with overview documentation for the version 2 UML model and how the model is mapped to XML and JSON.
- Section 5 now contains the version 2 UML model class and enumeration documentation.
- Section 6 now contains the version 2 XML schema aspects and usage documentation.
- Appendix E, XML Schema, and Appendix F, UML Class Diagram, were eliminated; these items are available at the locations specified in Appendix D.
- Appendix E now contains a detailed list of changes for version 2.

Table of Contents

List of Appendices

1   Introduction

This document is a revised 2.0 specification for the UML-based (Unified Modeling Language) [1] common data format (CDF) for exchanging pre-election and post-election data from voting systems used for managing elections and tabulating election results across states and territories of the United States. The format serves as a basic way to export election information from an election management system (EMS) and as a means for combining election data from different EMSs or transferring election data between EMSs. It defines common exchange methods between distributed voting places and central offices and from election offices to news media and the general public. These common exchange methods promote interoperability and eliminate the need for proprietary formats.

This specification includes the data model in UML that specifies and defines the data involved in pre-election setup and post-election results reporting. It also includes discussion and documentation for XML (eXtensible Markup Language) [2] and JSON (JavaScript Object Notation) [3] schemas that were derived from the UML model

The primary features of this specification are:

- major data elements and their attributes and associations are fully defined in a UML data model;
- the data model can be used to generate data formats (for example XML, JSON);
- election data and results can be reported at flexible levels from highly aggregated to very detailed;
- detailed reporting includes by device type, by type of ballot, and by geopolitical geographies including precinct and split precinct;
- geopolitical units of geography can be specified in a flexible manner to mirror reporting structures used across states, counties, and cities;
- major elements such as contests, geopolitical units, and parties include the capability to support multiple types of identifiers and cross-references; and
- detailed instructions for implementing and using the XML schema are included.

1.1   Why this specification is needed

The purpose of this specification is to provide a comprehensive, flexible, and interoperable pre-election setup and post-election results reporting XML and JSON format for manufacturers to integrate into their voting equipment and for election offices, the media, and other groups to use in their own software. Some advantages of using this specification are that:

- election results can be reported directly from election offices in this format regardless of voting system manufacturer, thus enabling interoperability;
- the need for custom software and custom reporting formats is greatly reduced;
- jurisdictions that use multiple versions of EMSs and tabulators can more easily combine and transfer information between systems; and

- consistency in election results reports across different voting systems, jurisdictions, and states, will make reporting on election analysis and performance surveys easier and more accurate, for example, the election administration and voting survey (EAVS) [4].

1.2 Intended Audience

The intended audience of this specification includes election officials, manufacturers and developers, and others in the election community, including the general public. Election results reporting is deceptively complex, thus some background in election administration or technology is useful in understanding the material in this specification.

1.3 Motivation and methodology

This specification was motivated primarily by the need to reduce the inherent complexity for U.S. election officials in collecting and publishing election data, especially on election night when time frames are tight and there are more opportunities for error and a greater need for automation. The process of reporting election results is a highly complicated activity that occurs over several different time frames and in multiple scenarios. The equipment involved and data produced often do not interoperate, adding more complexity to the process. Additionally, there are sometimes significant variations among different jurisdictions within a state as well as among the states in the way they report election results.

NIST and a community of U.S. election officials, analysts, and voting system manufacturers investigated reporting scenarios and their associated geopolitical geographies throughout the United States. They also evaluated other XML schemas associated with U.S. elections, including:

- the State of Florida XML schema for election results reporting [5],
- the Pew Voting Information Project XML schema [6],
- the OASIS Election Markup Language (EML) XML schema version 7.0 [7], and
- schemas created by the Associated Press[1].

From this analysis, three use cases were developed for election results reporting:

- **Pre-election** – election data that is known ahead of the election; basically, an export from an EMS of the contests, candidates, ballot initiatives, information about offices, and the geopolitical geographies associated with the reporting jurisdiction.

- **Election night** –
  - reporting of election results either summarized by contest and jurisdiction or broken down by individual reporting units such as precincts, and
  - associated formats, either as updates or corrections to previous reports or as internal intermediate reports within a state or county.

---

[1] The Associated Press does not make these schemas available to the public.

- **Post-election** – updates and the final results compiled during the post-election canvass.

A UML data model was subsequently generated to define the data associated with the use cases and to show the relationship and organization of the data elements. Finally, XML and JSON schemas were generated from the UML data model. The schemas define the rules of the respective format.

The advantages of using a UML data model as an intermediate step to generating the XML and JSON schemas include that:

1.   the model is independent of the concrete format or other potential formats that could be derived, and
2.   relationships between data elements are easier to correctly define and visualize when they are independent of any specific data format.

If changes are needed to the format, one can make changes to the UML model and generate a new version of the format.

Note that this specification addresses U.S. governmental elections and is not intended for use "as is" in other types of elections or in other countries. However, the specification was written with the intention that it be adaptable to other election environments.

1.4   Revision 2 Changes

As a result of usage in state elections, the initial version was revised, mainly in three ways:

a) by eliminating wrapper elements whose purpose was to make XML files easier to read but were implemented inconsistently,
b) by restructuring contest and ballot counts to make their usage consistent, and
c) by generating a JSON schema.

The complete list of changes is listed in Appendix E.

1.5   Document Structure

Section 2 starts with an overview of geopolitical geographies such as counties, districts, and precincts, describing how they are categorized, how they interrelate, and how election results are tied to them.
Section 3 contains an overview of the three use cases for election results reporting and the UML data model that implements the use cases.
Section 4 contains overview documentation for the UML model and explains how it is mapped to XML and JSON.
Section 5 contains documentation for the UML model.
Section 6 describes how to use the major features of the XML schema.

The appendices include references, definitions, acronyms, and instructions for downloading the files associated with this specification.

2   Background: Geopolitical Geography

This section provides an overview of the geopolitical geography in the United States as it relates to elections and election results reporting and provides background for how geopolitical geography is implemented in the UML model and XML/JSON schemas that are described in the remaining sections. Knowing what constitutes geopolitical geography and how it is interrelated and used in elections provides the underpinning for understanding the complexities of election results reporting.

2.1   The Primary Types of Geopolitical Geography

The primary types of geopolitical geography include those that run elections such as states, counties, and cities, as well as the many types of election districts that are tied to contests, precincts, and various other geographical units associated with political boundaries. Generally, the media and election analysts wish to obtain voting results comprised of these units; thus, the process of running an election includes associating contests and vote counts with these units for reporting.

Ballot counts and vote counts for contests can be associated with different types of geopolitical geography, ranging from aggregated counts associated with a county or state down to more detailed counts associated with a precinct and breakdowns of a precinct. Precincts are generally the smallest unit of geopolitical geography, and in many states, there is generally one polling place per precinct. Precincts can be thought of as the bricks or building blocks that compose all other geopolitical geography.

Geopolitical geography can often be quite complex in that some are hierarchical, others overlap, and still others change their boundaries regularly, sometimes several times within a year. Changes to city and district boundaries affect precinct boundaries, splitting them into multiple parts (called split precincts), with each part requiring a distinct ballot style.

The following sections break down geopolitical geography into three primary types and show how the geographies interrelate. These three types are:

1. Governmental-based geography
2. Political-based geography
3. Administrative-based geography

2.1.1   Governmental-based Geography

Governmental-based geography refers to entities that run elections and are well-established and do not change over time, with the exception of some cities. For many states, the governmental-based geography is hierarchical, as shown in Figure 1. This can be categorized as follows:

- States
- Counties
- Cities
- Towns and Townships
- Other Civil Divisions.

American Indian reservations may span states and are sovereign bodies for the purposes of their internal governmental elections. They are not represented hierarchically under the state, however voters residing in reservations vote in the same federal, state, and local elections as other voting residents of the state.

All states have counties, although some use different words to describe them, such as, parishes for Louisiana and boroughs for Alaska. Townships occur in 13 states and adhere to county boundaries.   In the six New England states, townships run the election process and there is no county government; thus, election results are reported directly to the state. Municipalities (cities, towns, or villages) in Michigan, Minnesota, and Wisconsin also run their elections, but report their information to the county, which then reports to the state. Other civil divisions include boroughs as used in Connecticut, New Jersey, Pennsylvania, and other states; New York City's boroughs are treated as counties.
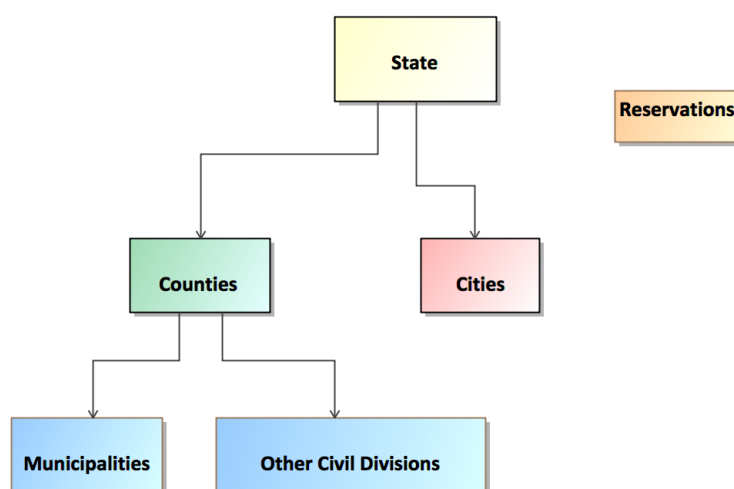
Figure 1 – Governmental-based Geographies

Governmental-based geographies are associated with offices that are elected jurisdiction-wide (such as for Governor, County Clerk, Supervisor, Treasurer, Assessor, and Highway Commissioner) and thus, do not require different ballot style areas within the geography for those offices, that is, all voters in the jurisdiction vote for the office.

Governmental-based geographies do not cross the lines of the precincts that compose them. However, cities can change their boundaries through annexations, and, in some states, city boundaries can also cross county boundaries. *Thus, changes to city boundaries may result in crossing the boundaries of one or more precincts, creating split precincts and requiring a distinct ballot style for each split precinct.*

2.1.2   Political-based Geography

Political-based geographies are those that tend to be population-based and therefore may change with each U.S. Census every 10 years in a process known as re-districting. Political-based geographies are also known as *election districts*, where people are elected to an office that has jurisdiction within a specific geography, for example, a U.S. Congressional district.
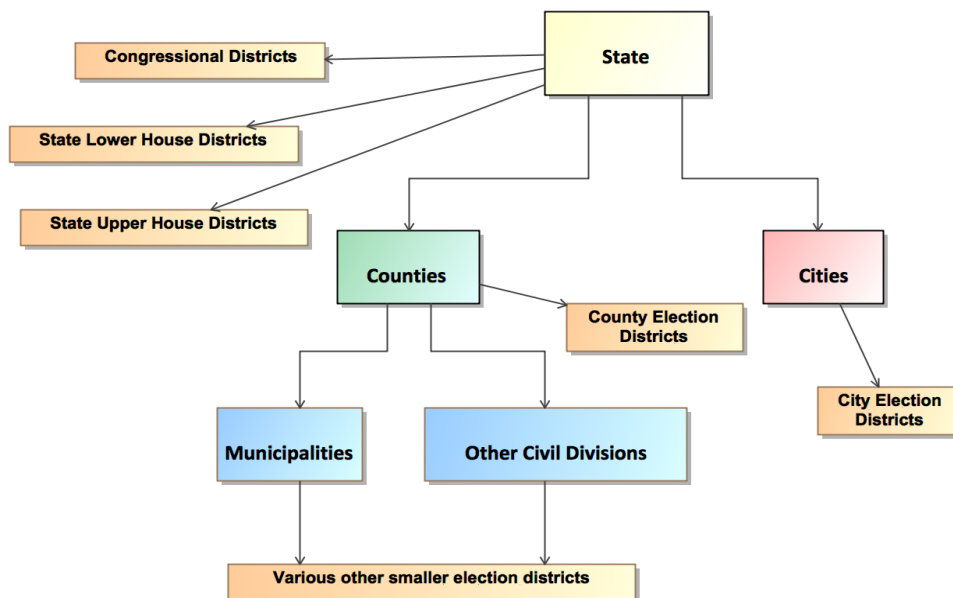
Figure 2 – Political-based Geographies

Figure 2 shows the most common political-based geographies as they interrelate with the governmental-based geographies. Political-based geographies can be categorized as follows:

- U.S. Congressional districts
- State senate or upper-house districts
- State house or lower-house districts (in some states, several state house districts combine to form a state senate district)
- County election districts
- City election districts (cities sometime call them Wards)
- Numerous other forms of election districts.

Because election districts can change as they are re-drawn, political-based geographies will often divide precincts, creating split precincts and requiring a distinct ballot style for each split precinct.

2.1.3   Administrative-based Geography

Administrative-based geographies are identified this way because their boundaries are determined by election or civil administration. Administrative-based geographies include precincts and their various types such as wards, combined precincts, and split precincts. They can be very small, sometimes only applying to several streets or houses or even only a single house along a street. They can involve territory that is non-contiguous in itself, for example, for some of the taxing and special districts. They can change a number of times throughout a given year, even daily in some cases. Figure 3 shows the basic administrative-based geographies, which can be categorized as follows:

- Election administrative areas
  - Precincts, split precincts, combined precincts, wards
  - Polling places, vote centers
  - Various other ballot style areas
- Taxing districts, such as fire, water, sewer, transit, school, police, hospital, utilities
- Special districts, that is, unique areas brought together for a referendum.
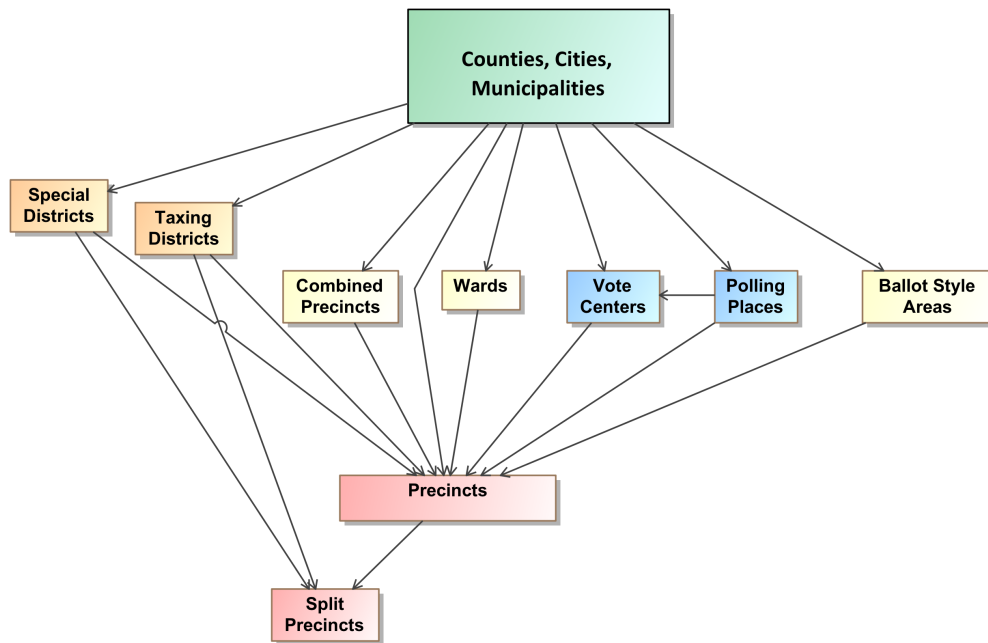
Figure 3 – Administrative-based Geographies

2.2   Linking the Geopolitical Geographies Together

As an example of administrative-based geographies and their relationship to political-based and governmental-based geographies, Figure 4 shows the wards and precincts that make up the city of Cambridge, MA, and Figure 5 shows how the wards and precincts in the city (where the precinct boundaries are delineated in red) compose the U.S. Congressional election districts [8]. The wards are implemented as collections of precincts and are shown in a distinct color in Figures 4 and 5.
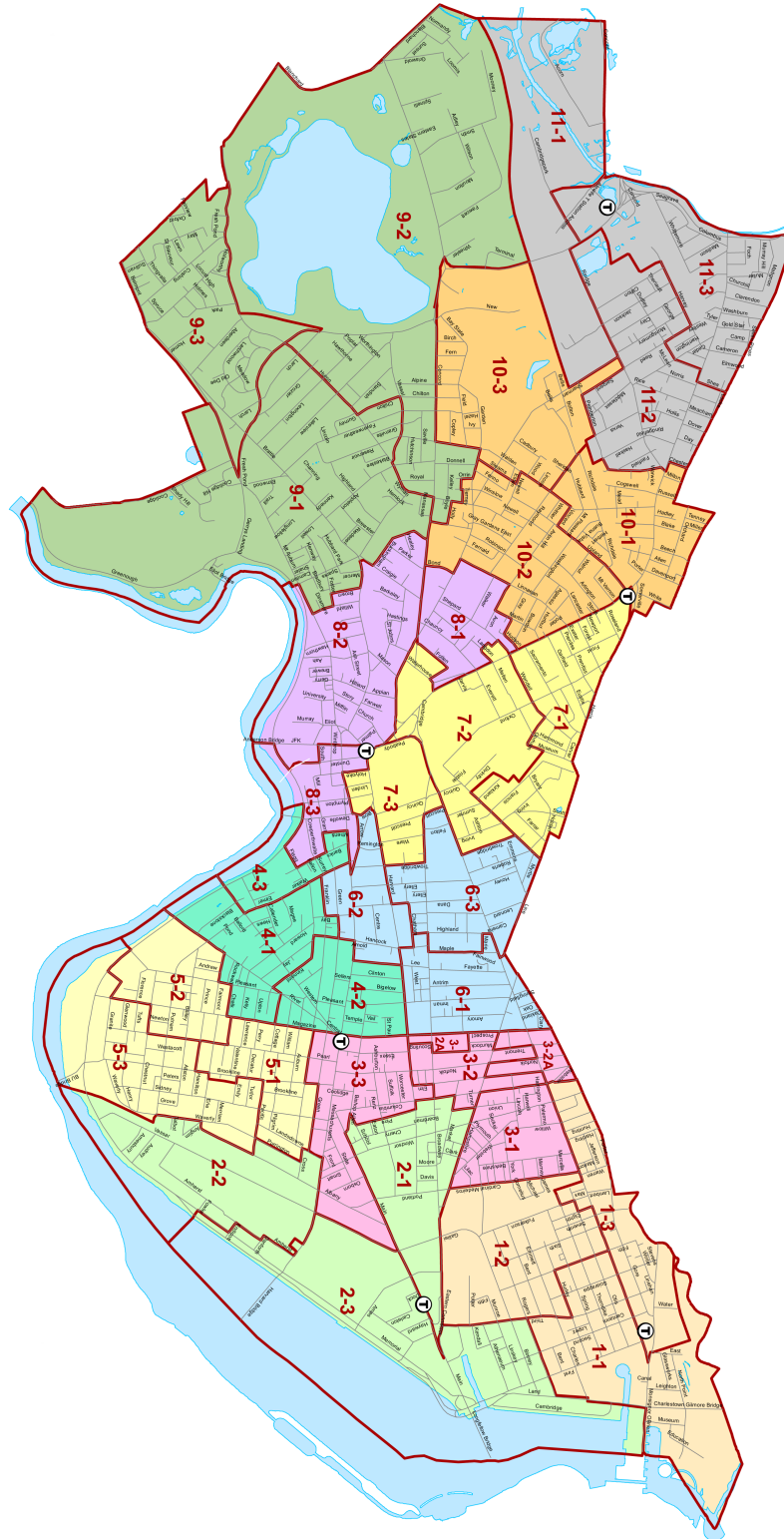
Figure 4 – Ward and Precincts in Cambridge, MA.
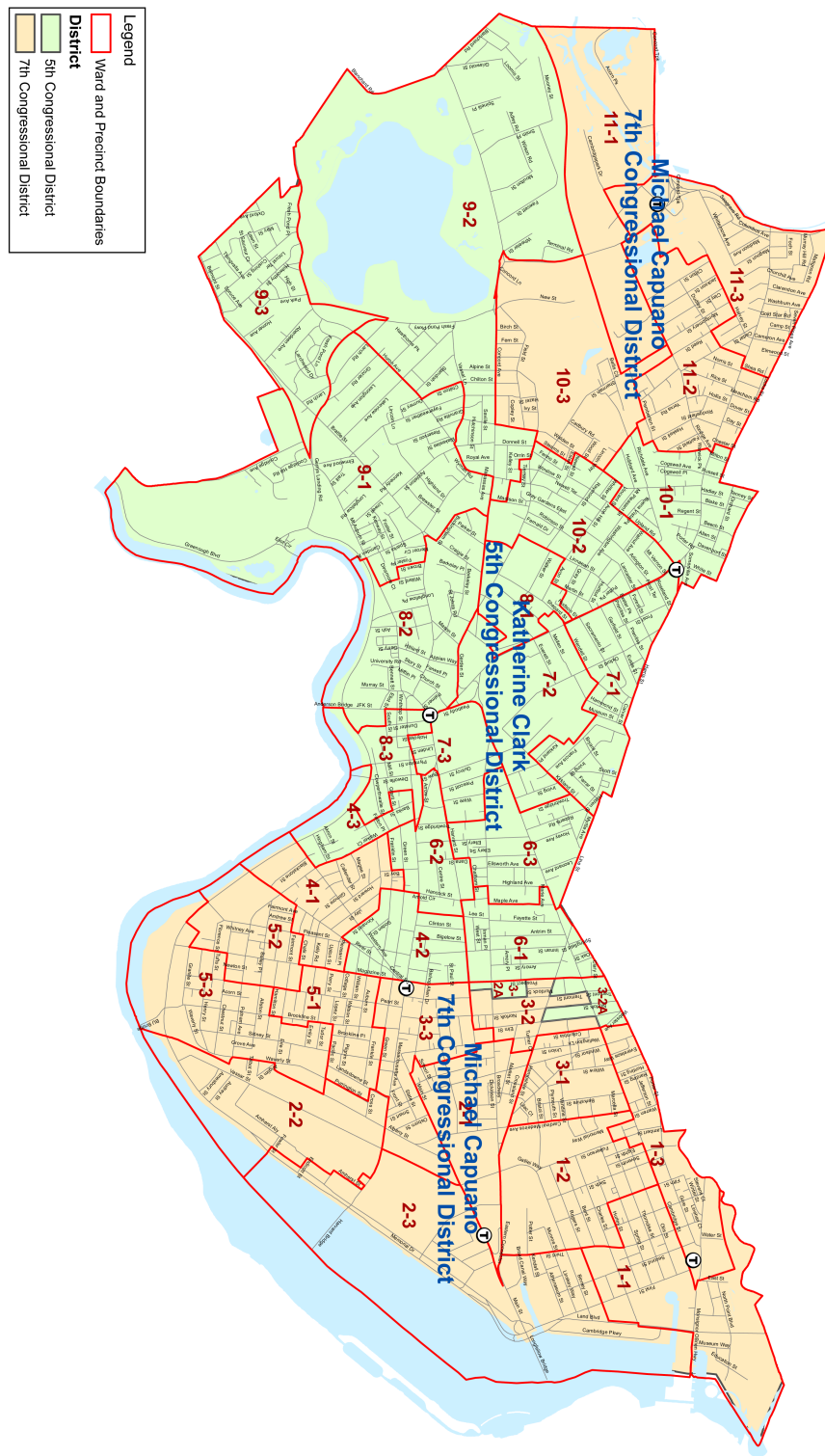
Figure 5 – Districts Overlaying Wards and Precincts in Cambridge, MA.

In many states, the boundaries of election districts may crisscross the precinct boundaries, creating one or more split precincts, with a distinct ballot style for each split precinct. Depending on the number of districts and how often they cross the precinct boundaries, the resulting number of ballot styles created could grow well beyond the number of whole precincts. It is possible that, despite best efforts, very low numbers of voters or even just one voter will require a district ballot style.
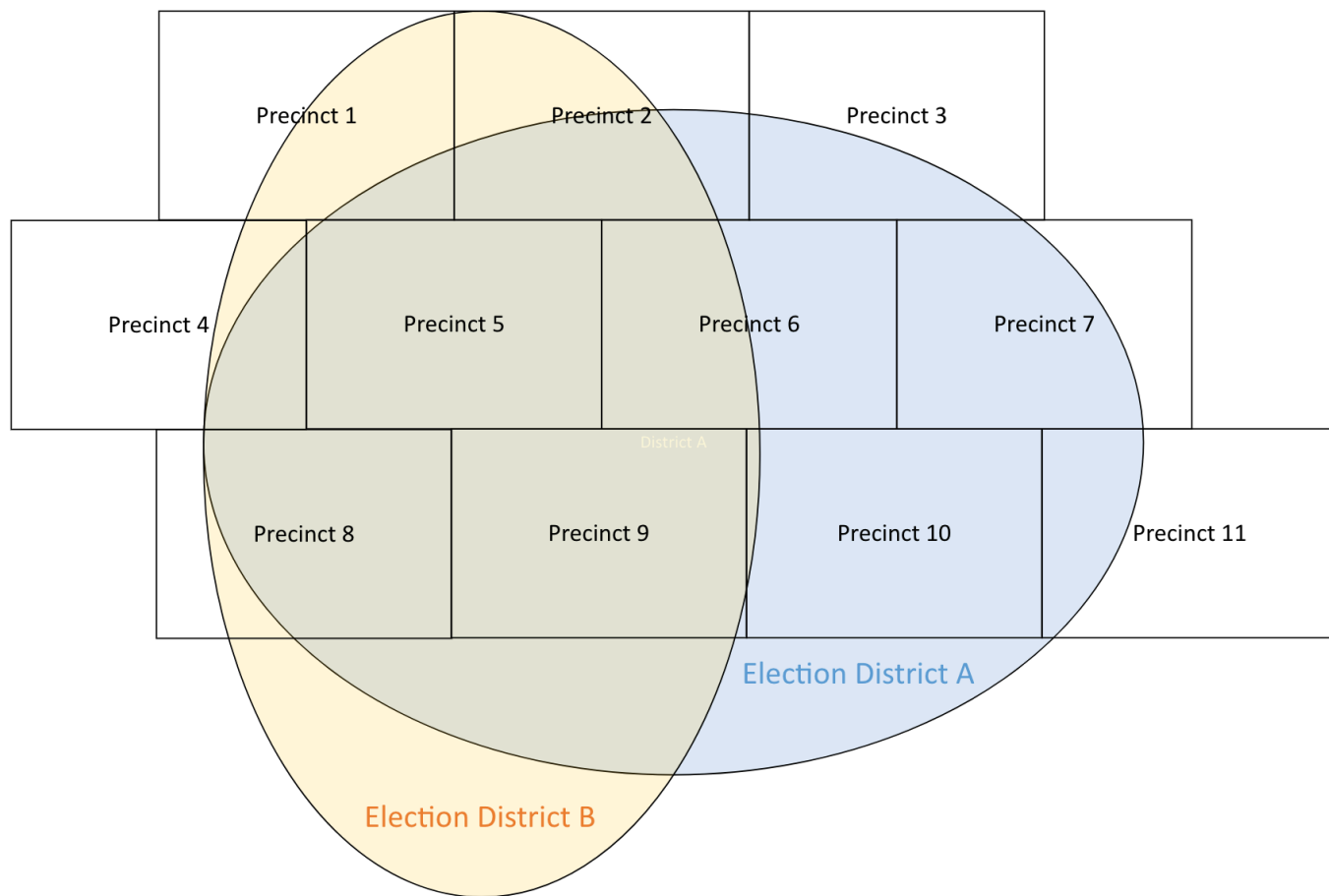
Figure 6 – Overlapping Non-hierarchical Election Districts

As an example, Figure 6 shows two election districts that overlay a number of precincts. Every time a precinct is not wholly contained within one or more districts, the precinct is split into however many pieces are necessary, each piece "belonging" to one of the districts. Figure 6 shows that a number of the precincts are split in different ways, for example, Precinct 6 is split into two pieces.

Precinct 6 will therefore require two distinct ballot styles, one for each of the splits. To correctly tabulate the votes in the different election districts, it will be necessary to know which split of Precinct 6 is contained within each of the two election districts.

Precincts can be split as well by changes to the other administrative-based geographies. Adding to the complexity, a number of states now use combined precincts and vote centers on election night, which associate multiple precincts with one polling place. This means that for a vote center handling multiple precincts that themselves may be split, there can be potentially many different ballot styles in use at the vote center, with each voting device needing to display or tabulate any one of the ballots. This adds further complexity and places additional demands on election jurisdictions on their ability to manage and report details of votes on election night and post-election [9].

To make this situation more manageable, some states and counties prefer over time to *heal* split precincts by combining them with other precincts or generally redrawing the precinct boundaries so that the number of ballot styles is reduced, and election management and reporting is less complicated.

2.3   Geopolitical Geography in the UML Model and XML Schema

The previous discussion demonstrated that there are different types of geopolitical geography that overlap each other or behave hierarchically, resulting sometimes in very complex maps and many small geopolitical units that require distinct, different ballot styles. Election officials may spend considerable time in managing this complexity.

Furthermore, each state and sometimes county or city will manage elections differently, using combinations of units such as combined precincts or wards, with specific rules about how the associated contests operate. When one combines the complexities of geopolitical geography with the different election rules employed in the U.S. states and territories, one sees that running an election can be an extremely complicated endeavor. Election results reporting mirrors this complexity.

Note that the different geographies form relationships much like a lattice, in which objects can be related in non-hierarchical ways. The UML model and XML schema implement geopolitical geography in this way using an object that can be linked with other objects depending on the type of geopolitical geography. In the UML model, this object is referred to as the GpUnit (short for 'Geopolitical Unit") class, and in the XML schema it is called the *<GpUnit>* element. GpUnits can model a district, or county, or precinct, and can be linked to each other to mirror the real-world geopolitical geography of the reporting jurisdiction.

GpUnits can be linked hierarchically when modeling jurisdictional geographies. To model a jurisdiction that runs or reports on elections, the lowest-level GpUnits, that is precincts, will be children of the election-running GpUnit, say a city or county or state.

District GpUnits need to be linked to the precinct or split precinct GpUnits that compose them. The precincts and split precincts thus link the jurisdictional and district GpUnits together, as shown below in Figure 7 (and described in greater detail in section 6.2). The wards in Figure 7 are synonymous with precincts in Wisconsin and are the children of the combined precincts, and so forth on up to the state. The precincts and split precincts are also the children of the districts that they compose.

Wisconsin
type = state

St. Croix County
type = county

Town of Richmond
type = municipality

State Senate District 10
type = state-senate

Wards 1 and 2
type = combined-precinct

Wards 3-5
type = combined-precinct

State Assembly District 29
type = state-house

State Assembly District 30
type = state-house

New Richmond School District
type = school

Ward 1
type = precinct

Ward 2
type = precinct

Ward 3
type = precinct

Ward 4
type = precinct

Ward 5
type = precinct

Ward 1 Split 1
type = split-precinct

Ward 1 Split 2
type = split-precinct

Congressional District 7
type = congressional

Somerset School District
type = school

Figure 7 - GpUnit Structural Hierarchies

3   Use Cases and UML Model

This section describes the three use cases for pre-election setup and post-election results reporting included in this specification, and then describes the UML model that implements the use cases and was used to generate the XML and JSON schemas. This section is provided for background and to assist in understanding how the model and schemas are structured and used.

3.1   Use Cases

This section describes the three use cases for the following phases of pre-election setup and post-election results reporting:

1. **Pre-election** – for reporting of election data about offices, political parties, and the geopolitical geographies associated with the reporting jurisdiction, as well as any election-specific information such as the contests, candidates, or ballot measures;
2. **Election night** – for reporting of election results to the public during the election or on election night and for internal state rolls-ups of election results between and among election management systems; and
3. **Post-election** – for reporting the final results compiled during the post-election canvass, that is, when all results are re-examined and re-tabulated to ensure correctness.

3.1.1   Pre-Election Use Case

The pre-election use case enables election officials to report on a variety of election-related data; the data could include information about a specific upcoming election or it could provide more general information about the reporting jurisdiction so as to determine whether the election data is accurate and organized correctly or to convey information to the general public about contests and ballot information in an upcoming election. Pre-election information may come from any of the databases or devices that an election jurisdiction uses to manage elections, including voter registration databases, ballot programming systems, candidate filing systems, EMSs, campaign finance systems, etc.

The data for the pre-election use case includes the following:

- Pre-election reports in a variety of formats, including:
    - As one complete file or a sequence of files;
    - As additions to previous reports; and
    - As corrections to previous reports;
- Jurisdictions, districts, offices, precincts and voting locations within the scope of the reporting jurisdiction and how they are organized;
- Political party information;
- Offices associated with contests and districts;
- Information about persons relevant to the jurisdiction such as authorities, candidates, office holders, etc.;

- Election type, date, place, jurisdiction, authority information, registered voters. The type of election could include:
    - Open and closed partisan primaries;
    - Runoffs;
    - Special elections; and
    - General elections;
- Ballot measures and contests, including retention contests;
- Candidates on the ballot and their associated party affiliations and contact information;
- Offices associated with the contests and election districts;
- Vote-capture device types or the specific devices associated with polling places; and
- Ballot styles containing contest and candidate information in the order as they appear on the ballot at specific precincts or split precincts.

3.1.2 Election Night Reporting Use Case

The election night reporting use case addresses aggregated, contest-level reporting of election results. Depending on a jurisdiction's capabilities, it could include further details such as precinct-level vote and ballot counts and breakdowns of counts by device type and ballot type.

The election night reporting use case involves election reporting managed in two primary ways:

1. By the counties (or cities, etc.) reporting directly to the public/media and also reporting upward to the state, or
2. By the counties reporting upward to the state and the state reporting directly to the public/media.

When reporting upward to the state, election results are reported from local jurisdictions to the city or county or state level, where the results continue to be aggregated in a process referred to as *rolling up the votes* and creating state or county *roll-ups*. As part of this process, internal, intermediate reports from different EMSs or databases need to be combined. The roll-ups are then formatted and released to the media and public during election night until all results from the local jurisdictions are accounted for. The results are, at this stage, considered as unofficial.

The data for election night reporting includes the following:

- Any information from the pre-election use case;
- Aggregated contest results for offices and ballot measures, including:
    - Votes reported in the contest;
    - Votes reported for each candidate(s) or selection(s) in the contest;
    - Overvotes reported for the contest; and
    - Undervotes reported for the contest;
- Further vote details as possible to report, including:
    - Breakdowns of votes by various reporting units such as precinct or split precinct;
    - Vote counts and ballot counts broken down by ballot type (absentee, early, etc.) and type of device (optical scan, etc.); and

14

      o   Additional counts as available, e.g., the number of votes cast by party or the number of straight party selections.

3.1.3 Post-Election Reporting Use Case

The post-election reporting use case addresses the updates to the election night unofficial results. Post-election results contain greater detail than election night results, including results by reporting unit, by counts of different types of ballots (early, absentee, provisional, etc.), and by the type of vote-capture device used by voters to cast their ballots. They may also include results that were not available for election night reporting, such as absentee ballots received too close to the election day, or provisional ballots. This use case is of interest especially by analysts and media performing detailed analysis of election results and other items of interest such as the methods people are choosing to vote (e.g., favoring absentee over in-person) or provisional voting rejection rates.

The data for post-election night reporting includes:

- Any information from the pre-election and election night reporting use cases;
- Ballot measure and contest vote counts, including:
  - By geopolitical units of geography (e.g., county, township, city, precinct, etc.);
  - By type of ballot/voting (e.g., absentee, early, in-person, provisional);
  - By type of voting equipment used (e.g., optical scan, etc.); and
  - By overvotes and undervotes for each contest;
- Ballot measure and contest summaries broken down as per the vote counts, but also including the number of ballots on which the ballot measure or contest appeared; and
- Precinct summaries broken down as per the vote counts.

4   UML Model Overview

This section presents the UML model that was structured to implement the use cases. It includes a class diagram that shows a picture of the model and an overview of how to read the relationships between the classes so as to understand how the model and related schemas are structured.

The UML model represents a format-independent description of the data required by the three use cases of the specification. Its primary benefit is that it unambiguously defines and describes the data elements and how they are related without requiring readers to know the technical details of any particular data format implementation, e.g., XML. By using a model-based approach, the resultant data format is more likely to be well-structured and more tolerant to modifications. The data format can be generated from the model using commercial tools, thus if changes need to be made to a format, the model can be changed, and the format can be re-generated.

Figure 8 shows a high-level view of the class diagram, minus its attributes and certain ancillary classes (the complete diagram is available for download, see Appendix D).
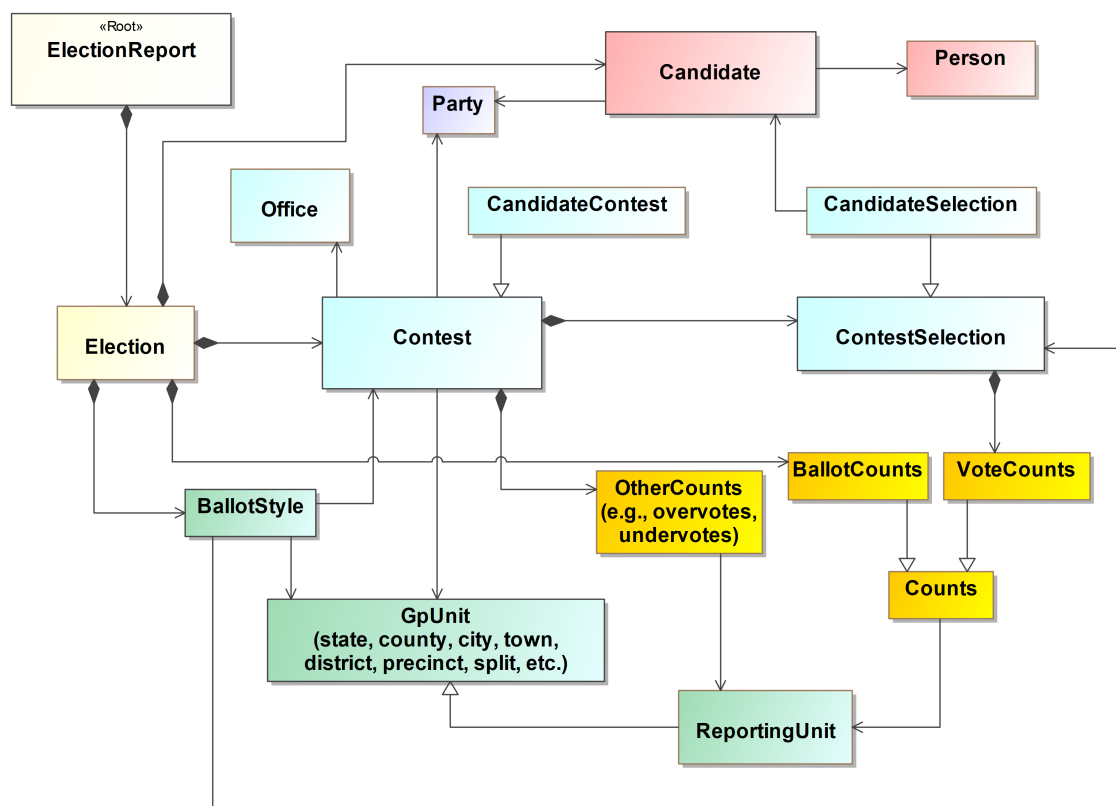
Figure 8 – Simplified UML Class Diagram

4.1   Major Classes

Each class represents a major data element, e.g., Contest, Candidate, or Party, and classes that are highly related to each other are shown in the same color. The major classes shown in Figure 8 are:

- **ElectionReport** – the root class, includes attributes describing the type of report, when generated, etc.;
- **Election** – for a specific election, includes attributes describing the type of election, date, etc.; it also links to ballot counts for the election;
- **GpUnit** – short for "Geopolitical Unit," for describing units of geopolitical geography so that they can be associated with contests, offices, ballot styles, and election results. There are two major types of GpUnits:
    1. *ReportingUnit* – for jurisdictions, districts, precincts, and other units for which election results can be associated; and
    2. *ReportingDevice* – for types of voting devices or specific voting devices for which election results can be associated (not shown in Figure 8);
- **Office** – for political offices that are associated with contests and election districts and current office holders;
- **Party** – for political parties associated with the reporting jurisdiction and for associating the parties with candidates and contests;
- **Person** – for address and contact information for persons associated with the reporting jurisdiction, including boards of authorities, candidates, current office holders, or other election-related officials;
- **Contest** – for contests and for linking together the major elements needed for contests such as candidates and contest selections and election results. There are four types of Contests:
    1. *CandidateContest* – for contests involving candidates;
    2. *BallotMeasureContest* – for contests involving ballot measures (not shown in Figure 8);
    3. *PartyContest* – for straight-party selection (not shown in Figure 8); and
    4. *RetentionContest* – for judicial or other types of retention contests (not shown in Figure 8);
- **Candidate** – for describing candidates so that they can be associated with contests;
- **ContestSelection** – for describing the types of contest selections in a contest and associating them with election results:
    1. *BallotMeasureSelection* – used if the contest selection is for a ballot measure (not shown in Figure 8);
    2. *CandidateSelection* – used if the contest selection is for a candidate; and
    3. *PartySelection* – used if the contest selection is for a party as in straight party selections (not shown in Figure 8);
- **Counts** – for various counts associated with the election; the counts can be linked to geopolitical units such as for counties, precincts, etc.:
    1. *VoteCounts* – used for vote counts associated with a contest selection;
    2. *BallotCounts* – used for types of ballot counts;
- **OtherCounts** – for describing summary counts of overvotes and undervotes and write-ins associated with a contest;
- **BallotStyle** – for describing the contests and contest selections on a ballot and linking them to geopolitical units such as precincts.

The attributes associated with the UML classes correspond closely to the XML and JSON schemas' attributes and sub-elements associated with the major elements, described in Section 5.

4.1.1   Understanding Relationships Between Classes

The major classes in the UML model result in major elements in the schemas, and the different types of relationships between the UML classes determine how the elements are structured (linked) in the schema. There are three types of relationships between the classes:

Directed Composition: see Figure 9, ElectionReport and Election, read as, "An election report is composed of elections." In the XML schema for example, the *<Election>* element will be generated as a sub-element of the *<ElectionReport>* element. A directed composition relationship has a closed diamond at one end and an arrow pointing to the composing class:

Figure 9 – Directed Composition Example

Is a Type of or Instance of (Generalization): see Figure 10, Contest and CandidateContest, read as, "A candidate contest is a type of contest." Contest is an abstract class; it is "implemented" by its concrete classes such as CandidateContest. In the XML schema, Contest will be generated as an abstract element and serve as an extension base to the *<CandidateContest>* element. An instance relationship has an open triangle at one end, pointing from the concrete class to the abstract class:

Figure 10 – "Is a Type of" Example

Directed Association: see Figure 11, Candidate and Party, read as, "A candidate is associated with or linked to a party." In the XML schema, the *<Candidate>* element will include a *<PartyId>* element, which will contain an identifier associated with a *<Party>* element. A directed association has an arrow at one end, goes in one direction, and serves to link the class to another associated class, e.g., the party linked to the candidate:

Figure 11 – Directed Association Example

For the directed composition associations in Figure 8, one can see that ElectionReport is composed of Election, GpUnit, Office, Party, and Person. Election is composed of BallotStyle, Candidate, and Contest. Contest is composed of ContestSelection and SummaryCounts. ContestSelection is composed of VoteCounts. GpUnit is also composed of SummaryCounts.

For the "is a type of" relationships in Figure 8, one can see that CandidateContest is a type of Contest. ReportingUnit is a type of GpUnit. CandidateSelection is a type of ContestSelection.

Lastly, for the directed associations in Figure 8, one can see that many of the classes are associated with each other, as one would expect. For example, Party is associated with (or linked to) Candidate just as a political party would be associated with or linked to a candidate. Candidate is associated or linked to CandidateSelection, just as a selection on the ballot in a candidate contest would be for a specific candidate.

4.2   Mapping the Model to XML and JSON

This section contains an overview of how the UML model is mapped to the XML and JSON data formats. The schemas are available for download; see Appendix D for file locations.

4.2.1   Root Element

The UML class model is a graph data structure in the sense that there is no hierarchy implied by the model. However, the two supported implementation formats, XML and JSON, are hierarchical in structure (also called a tree). All hierarchies must start with a root, which is indicated in the UML model as a class with the «root» stereotype applied. In Figure 8, the ElectionReport class will be generated in XML as the root element *<ElectionReport>*.

4.2.2   Elements and Attributes

In the UML model, the classes become major elements in the XML schema. The attributes of a class become XML sub-elements of the major elements.

JSON (as with UML) only provides a single data structure for presenting information, objects, whereas XML provides two data structures for presenting data, elements and attributes. If a UML attribute has the «xmlAttribute» stereotype or the class contains an attribute with the «simpleContent» stereotype, the attribute becomes an XML attribute. The «simpleContent» stereotype indicates that the attribute is the target for the character data of the XML element. For example, the following UML represents a class named FileValue with two attributes:

```
FileValue
«simpleContent»-Data : base64Binary [1]
-fileName : String [0..1]
-mimeType : String [0..1]
Table: UML Class File
```

An example using XML is:

```
<FileValue fileName="String"
mimeType="string">UjBsR09EbGhjZ0dTQUxNQUFBBUUNBRU1tQ1p0dU1GGUXhEUzhi</FileValue>
```

4.2.3   Data Type Mapping

The UML model uses a number of primitive and specialized types based on the XML schema, as shown in Table 1:

UML	XML	JSON	JSON Format
String	xsd:String	String	
date	xsd:date	String	date
Boolean	xsd:boolean	Boolean	boolean
base64Binary	xsd:base64Binary	String	byte
anyURI	xsd:anyURI	String	uri
float	xsd:float	number	

Table 1 - UML, XML, JSON type equivalents

4.2.4   Representing Multiplicities

Some attributes in the UML model can repeat. Repetition is represented by multiplicities whose lower or upper cardinality is greater than one, e.g., 1..2, 2..4, 0..*, etc. An asterisk (*) for the upper cardinality represents an unbounded number of repetitions. A "1" for the lower cardinality indicates that the attribute is required.

UML attributes with multiplicities greater than one are represented in XML as repeating elements. For example:

```
<Uri Annotation="mobile">http://mobile.samplesite.com/</Uri>
<Uri Annotation="desktop">http://www.samplesite.com/</Uri>
```

In JSON, UML attributes with multiplicities greater than one are represented as an array of objects. For example:

```
"Uri":
    [
        {
            "@type": "ElectionResults.AnnotatedUri",
            "Annotation": "mobile",
            "Content": "http://mobile.samplesite.com/"
        },
        {
            "@type": "ElectionResults.AnnotatedUri",
            "Annotation": "desktop",
            "Content": "http://www.samplesite.com/"
        }
    ]
```

Even if an implementer wants to provide a single occurrence of an attribute, it must be wrapped in an array:

```
"Uri":
    [
        {
            "@type": "ElectionResults.AnnotatedUri",
            "Annotation": "mobile",
            "Content": "http://mobile.samplesite.com/"
        }
    ]
```

4.2.5   References

Some classes of data may be referenced repeatedly, for example political parties or geopolitical units. It would make sense to define single instances of these classes and then reference them whenever they are needed as opposed to creating new instances. The UML model represents these references as directed associations between classes. Figure 12 shows a reusable ReportingUnit that is referenced by multiple Contests.
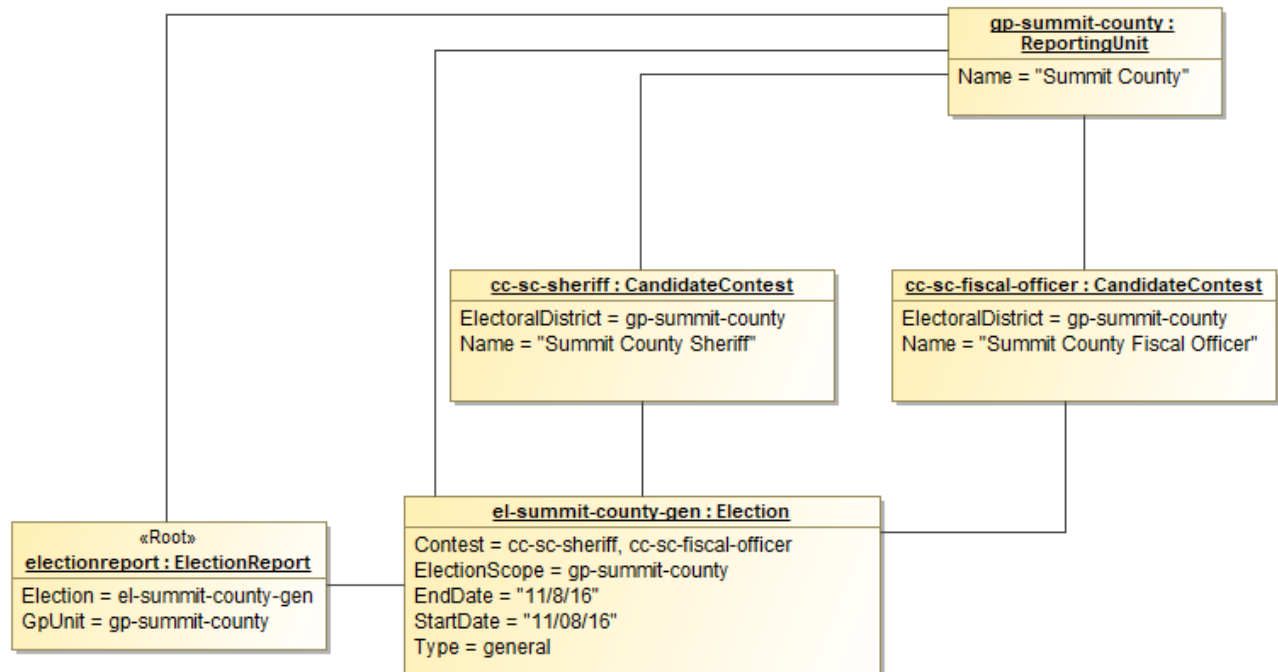
Figure 12 - UML Instance using references

In this figure, two contests are defined; they are of type CandidateContest and both have the same election district: Summit County. Only one reporting unit for Summit County needs to be defined, however, because the contests can reference the reporting unit as needed.

The XML representing the classes in Figure 12 is as follows:

```
<ElectionReport>
    <Election>
        <Contest ObjectId="cc-sc-sherif" xsi:type="CandidateContest">
            <ElectionDistrictId>gp-summit-county</ElectionDistrictId>
```

```
                    <Name>Summit County Sheriff</Name>
            </Contest>
            <Contest ObjectId="cc-sc-fiscal-officer"
             xsi:type="CandidateContest">
                    <ElectionDistrictId>gp-summit-county</ElectionDistrictId>
                    <Name>Summit County Fiscal Officer</Name>
            </Contest>
            <ElectionScopeId>gp-summit-county</ElectionScopeId>
            <Type>general</Type>
      </Election>
      <GpUnit ObjectId="gp-summit-county">
            <Name>Summit County</Name>
      </GpUnit>
</ElectionReport>
```

XML provides two built in types for handling references: one for establishing the reusable element (xsd:ID) and one for referencing it (xsd:IDREF). Identifiers are defined using the ObjectId attribute. The name of the identifier must be unique across the XML instance and conform to restrictions specified by the xsd:NCName datatype. *An xsd:NCName value must start with either a letter or underscore and may contain only letters, digits, underscores, hyphens, and periods.* JSON references are handled by the use of an @id property. An example follows:

```
{
      "@type": "ElectionResults.ElectionReport",
      "Election": [
            {
                    "@type": "ElectionResults.Election",
                    "Contest": [
                            {
                                    "@id": "cc-sc-sheriff",
                                    "@type": "ElectionResults.CandidateContest",
                                    "ElectionDistrict": "gp-summit-county",
                                    "Name": "Summit County Sheriff"
                            },
                            {
                                    "@id": "cc-sc-fiscal-officer",
                                    "@type": "ElectionResults.CandidateContest",
                                    "ElectionDistrict": "gp-summit-county",
                                    "Name": "Summit County Fiscal Officer"
                            }
                    ],
                    "Type": "general"
            }
      ],
      "GpUnit": [
            {
                    "@id": "gp-summit-county",
                    "@type": "ElectionResults.ReportingUnit",
                    "Name": "Summit County"
            }
      ]
}
```

5   UML Class Documentation

This section contains documentation for each of the UML classes and enumerations. In some cases, examples in XML are included to clarify the documentation. XML is used because the structure of the elements, sub-elements, and enumerations closely resembles that of the UML model.

Curly braces around an attribute indicate that the attribute is a reference to an instance of a different class. For example, the attribute {GpUnit} is a reference to an instance of the GpUnit class.

5.1   Class AnnotatedString

Used as a type for character strings; it adds a 16-character annotation to a character string. An example in XML is:

```
<Email annotation="Work">john.a.smith@26003.org</Email>
<Phone annotation="Cell">304-243-1234</Phone>
```
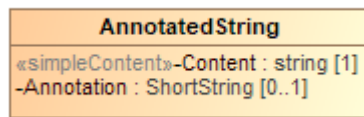
Figure 13 - AnnotatedString

Attribute	Multiplicity	Type	Attribute Description
Annotation	0..1	ShortString	An annotation of up to 16 characters associated with a character string.
Content	1	String	The string to be annotated.

5.2   Class AnnotatedUri

Used as a type for character strings that represent Uniform Resource Identifiers (URI); it adds a 16-character annotation to a character string.
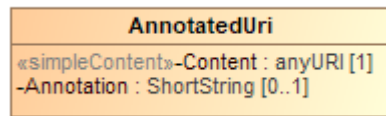
Figure 14 - AnnotatedUri

Attribute	Multiplicity	Type	Attribute Description
Annotation	0..1	ShortString	An annotation of up to 16 characters associated with a character string.
Content	1	anyURI	The URI to be annotated.

5.3   Class BallotCounts

Used for identifying various ballot counts. It inherits the attributes of Counts. Its syntax in XML is:

```
<Counts xsi:type="BallotCounts" ... />
```
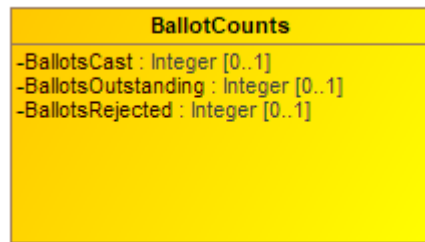

Figure 15 - BallotCounts

Attribute	Multiplicity	Type	Attribute Description
BallotsCast	0..1	Integer	Number of ballots cast.
BallotsOutstanding	0..1	Integer	Number of ballots not yet counted.
BallotsRejected	0..1	Integer	Number of ballots rejected.

5.4   Class BallotMeasureContest

For ballot measure (i.e., referenda or a tax measure) and judicial retention contests. It inherits the attributes of Contest. Its syntax in XML is:

```
<Contest xsi:type="BallotMeasureContest" ... />
```

If the type of ballot measure is not listed in enumeration BallotMeasureType, use "other" and include the type (that is not listed in the enumeration) in OtherType.

Figure 16 – BallotMeasureContest

Element	Multiplicity	Type	Element Description
ConStatement	0..1	InternationalizedText	For a statement on the ballot associated with a "no" vote.
EffectOfAbstain	0..1	InternationalizedText	For a statement on the ballot detailing the effect of abstaining from voting on the ballot measure.
FullText	0..1	InternationalizedText	For full text on the ballot of the ballot measure.
InfoUri	0..*	AnnotatedUri	For associating a URI with the ballot measure contest.
PassageThreshold	0..1	InternationalizedText	For a statement on the ballot of the number or percentage of votes needed to approve or pass the ballot measure.
ProStatement	0..1	InternationalizedText	For a statement on the ballot associated with a "yes" vote.
SummaryText	0..1	InternationalizedText	For a summary on the ballot of the ballot measure.
Type	0..1	BallotMeasureType	For indicating the type of ballot measure.

Element	Multiplicity	Type	Element Description
OtherType	0..1	String	Used when BallotMeasureType is other.

5.5   Class BallotMeasureSelection

For a contest selection in a ballot measure contest. Because judicial or other retention contests are often treated like ballot measure contests, this element can be used also for retention contests. It inherits the attributes of ContestSelection. Its syntax in XML is:

```
<ContestSelection xsi:type="BallotMeasureSelection" ... />
```


Figure 17 - BallotMeasureSelection

Attribute	Multiplicity	Type	Attribute Description
ExternalIdentifier	0..*	ExternalIdentifier	For associating an ID with the ballot measure selection.
Selection	1	InternationalizedText	Contains the text used to indicate a vote for or against the ballot measure, e.g., "yes", "no".

5.6   Class BallotStyle

For defining a ballot style composed of ordered content (i.e. Headers or Contests) and their contest selections, and associating the ballot style with a political party, a reference to an image of the ballot, and a reference to a precinct or other geopolitical unit that the ballot is unique to. Election includes BallotStyle.

BallotStyle references OrderedContent to include content that appears on that ballot style. To preserve any rotation associated with the ballot, it is expected that the generating application will list the occurrences of OrderedContest in the order as on the ballot for the associated geopolitical unit.

BallotStyle references one or more GpUnit instances defined for the associated precincts or split precincts. If the ballot style is associated with multiple precincts (or other geographies), multiple references to the precinct GpUnit instances can be included. An example in XML is:

```
<BallotStyle>
    <GpUnitIds>Precinct5 Precinct6 Precinct7</GpUnitIds>
    <OrderedContest ... />
    ...
</BallotStyle>
```

When including ExternalIdentifier, if the type is not listed in enumeration IdentifierType, use "other" and include the type (that is not listed in the enumeration) in OtherType.

```
                        BallotStyle
-ExternalIdentifier : ExternalIdentifier [0..*]
-ImageUri : AnnotatedUri [0..*]
```
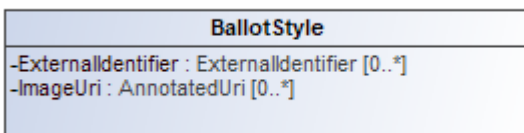
Figure 18 - BallotStyle

Attribute	Multiplicity	Type	Attribute Description
ExternalIdentifier	0..*	ExternalIdentifier	For associating an ID with the ballot style.
{GpUnit}	1..*	GpUnit	Unique identifier for one or more GpUnit instances. For associating specific geopolitical units with the ballot style.
ImageUri	0..*	AnnotatedUri	URI for a ballot image.
{OrderedContent}	0..*	OrderedContent	For associating a ballot style with ballot content, such as contests or headers.
{Party}	0..*	Party	Unique identifier for one or more Party instances. For associating one or more

Attribute	Multiplicity	Type	Attribute Description
			parties with the ballot style.

5.7   Class Candidate

For defining information about a candidate in a contest. CandidateSelection references Candidate instances to associate one or more candidates with a contest selection. Election includes Candidate.

Candidate uses the Party association to reference the candidate's political party. If the candidate is endorsed by other parties for a particular contest, the endorsing parties are referenced using the CandidateSelection attribute. ExternalIdentifier can be used to associate an ID with the candidate. If the type is not listed in enumeration IdentifierType, use "other" and include the type (that is not listed in the enumeration) in OtherType.

```
                        Candidate
-BallotName : InternationalizedText [1]
-ExternalIdentifier : ExternalIdentifier [0..*]
-FileDate : date [0..1]
-IsIncumbent : boolean [0..1]
-IsTopTicket : boolean [0..1]
-PostElectionStatus : CandidatePostElectionStatus [0..1]
-PreElectionStatus : CandidatePreElectionStatus [0..1]
```
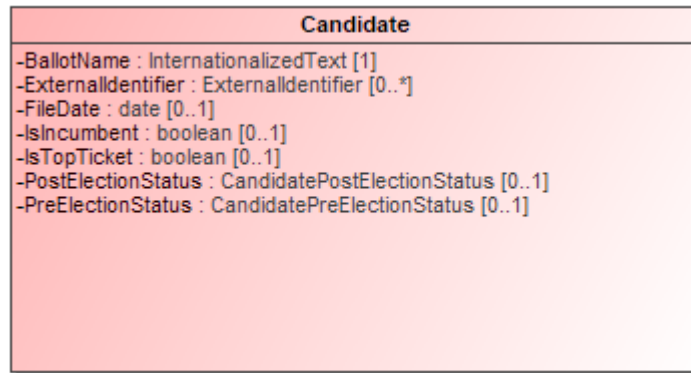
Figure 19 - Candidate

Attribute	Multiplicity	Type	Attribute Description
BallotName	1	InternationalizedText	For the candidate's name as listed on the ballot.
CampaignSlogan	0..1	InternationalizedText	The slogan or motto used by the candidate in their campaign.
{ContactInformation}	0..1	ContactInformation	For associating contact information for the candidate.
ExternalIdentifier	0..*	ExternalIdentifier	For associating codes with the candidate.
FileDate	0..1	date	Date when the candidate filed for the contest.
IsIncumbent	0..1	Boolean	Boolean to indicate whether the candidate is the incumbent for the office associated with the contest. Assumed to be "no" if not present.

31

Attribute	Multiplicity	Type	Attribute Description
IsTopTicket	0..1	Boolean	Boolean to indicate whether the candidate is the top of a ticket that includes multiple candidates. Assumed to be "no" if not present.
{Party}	0..1	Party	For associating a party with the candidate.
{Person}	0..1	Person	For associating more detailed information about the candidate.
PostElectionStatus	0..1	CandidatePostElectionStatus	Final status of the candidate, e.g., winner, withdrawn, etc.
PreElectionStatus	0..1	CandidatePreElectionStatus	Registration status of the candidate, e.g., filed, qualified, etc.

5.8 Class CandidateContest

For a contest that involves selecting one or more candidates. It inherits the attributes of Contest. Its syntax in XML is:

```
<Contest xsi:type="CandidateContest" ... />
```

This class optionally references Office and Party. If the candidate contest is associated with a ticket (of candidates) and each candidate in the ticket is associated with a separate office, the association to Office can reference each of the separate offices. For example, if the contest is for the state governor ticket but Governor and Lieutenant (Lt.) Governor are both separate offices, the association references first to the Office instance defined for the Governor's office and then to the Lt. Governor's office. In this case, it is expected that the generating application will list the multiple references according to a jurisdiction-defined ordering scheme, e.g., Governor first and Lt. Governor second.

Note that when using the CandidateSelection class to associate the candidates with a contest selection for the contest, the order of the candidates should match the order of offices. Again, using the example of the state governor ticket, if the offices are listed with Governor first and Lt. Governor second, then the order of the candidates in the ContestSelection instance should be identical, with the Governor candidate first and the Lt. Governor candidate second.

CandidateContest
-NumberElected : Integer [0..1]
-NumberRunoff : Integer [0..1]
-VotesAllowed : Integer [1]
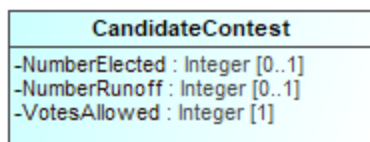
Figure 20 – CandidateContest

Attribute	Multiplicity	Type	Attribute Description
NumberElected	0..1	Integer	Number of candidates that are elected in the contest ("n" of n-of-m).
NumberRunoff	0..1	Integer	The number of candidates in a runoff contest.
{Office}	0..*	Office	For associating office descriptions.
{PrimaryParty}	0..*	Party	For associating parties with the contest.
VotesAllowed	1	Integer	Maximum number of votes per voter in this contest.

5.9   Class CandidateSelection

For the contest selections in a candidate contest, including for write-ins. It inherits the attributes of ContestSelection. References to multiple Candidate instances can be included if necessary, e.g., when the contest selection would be for a ticket of candidates (unless the ticket itself is defined as a candidate). Its syntax in XML is:

```
<ContestSelection xsi:type="CandidateSelection" ... />
```

EndorsementParty is used to reference any associated endorsement parties other than the specific party of the candidate (Candidate references Party for that purpose). For example, if a candidate of one party is also endorsed by a second party, use EndorsementParty to reference the second party. A second example would be for ballot fusion as used in some states, where the same candidate is listed multiple times in the same contest, but with different endorsement parties.

When multiple candidates are referenced for a ticket and the ordering of the candidates is important to preserve, it is expected that the generating application will list the references to Candidate instances according to the ordering scheme in place. For example, if the contest is for a ticket in which each candidate is associated with a different office, then the order of the candidates should match the same ordering of the <Office> element references within <OfficeIds> in the <Contest xsi:type="CandidateContest" ... /> element.
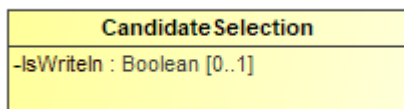
Figure 21 - CandidateSelection

Attribute	Multiplicity	Type	Attribute Description
{Candidate}	0..*	Candidate	For associating a candidate with the candidate selection on the ballot and for cases where the contest selection is for multiple candidates, e.g., a ticket.
{EndorsementParty}	0..*	Party	For associating one or more endorsing parties with the candidate selection.
IsWriteIn	0..1	Boolean	Indicates whether the candidate is a write-in, e.g., true or false. Assumed to be false if not present.

34

5.10 Class Coalition

For defining a coalition, i.e., a collection of parties organized for the purpose of endorsing a candidate in a contest. It inherits the attributes and elements of Party.

Coalition instances themselves are composed of multiple Party references along with a reference to an associated Contests.

If there are no associated Contests, a general default is that the coalition endorses the associated parties.
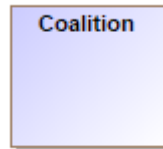
Figure 22 - Coalition

Attribute	Multiplicity	Type	Attribute Description
{Contest}	0..*	Contest	For associating contests with the coalition.
{Party}	0..*	Party	For associating parties with the coalition.

5.11 Class ContactInformation

For defining contact information about objects such as persons, boards of authorities, organizations, etc. Election, ElectionAdministration, Person, GpUnit, and Office include ContactInformation.

To include an address for the contact, use multiple occurrences of AddressLine. It is expected that the generating application will list the name of the person/organization in the first occurrence of AddressLine, with subsequent ordered occurrences for street address, city, state, zip code, etc. Directions can be used to supply any additional address-related information that may appear in multiple languages.

ContactInformation includes LatLng so as to associate latitude/longitude with the contact address.

Email, Fax, and Phone are of type AnnotatedString, which permits up to a 16-character annotation to be associated with the data. An example in XML is:

```
<Email annotation="Work">john.a.smith@26003.org</Email>
<Email annotation="Home">john.a.smith@mybusiness.com</Email>
```

ContactInformation

-AddressLine : RichText [0..*]
-Directions : InternationalizedText [0..1]
-Email : AnnotatedString [0..*]
-Fax : AnnotatedString [0..*]
«xmlAttribute»-Label : String [0..1]
-Name : RichText [0..1]
-Phone : AnnotatedString [0..*]
-Uri : AnnotatedUri [0..*]
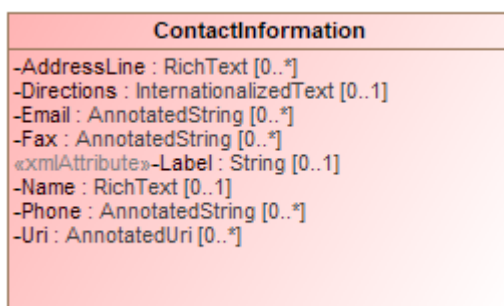
Figure 23 - ContactInformation

Attribute	Multiplicity	Type	Attribute Description
AddressLine	0..*	RichText	For associating an address with the contact.
Directions	0..1	InternationalizedText	Directional information in addition to address information.
Email	0..*	AnnotatedString	Email address associated with the contact.
Fax	0..*	AnnotatedString	Fax number associated with the contact.
Label	0..1	String	For use as needed and compatibility with the VIP schema.
{LatLng}	0..1	LatLng	For latitude and longitude information associated with the contact.
Name	0..1	RichText	Name associated with the contact.

Attribute	Multiplicity	Type	Attribute Description
Phone	0..*	AnnotatedString	Phone number associated with the contact.
{Schedule}	0..*	Schedule	For associating a schedule with the contact.
Uri	0..*	AnnotatedUri	URI associated with the contact.

5.12 Class Contest

For defining a contest and linking the contest to the associated candidates, ballot measures, parties, or retention contests. Election includes Contest.

Contest is an abstract class with four subclasses that get used according to the type of contest:

- BallotMeasureContest, used for a contest involving a ballot measure

- CandidateContest, used for a contest involving one or more candidates for an office

- PartyContest, used for a contest for a straight party selection on the ballot

- RetentionContest, used for a judicial or other type of retention contest

Examples in XML are:

- `<Contest xsi:type="BallotMeasureContest">`, used for a contest involving a ballot measure
- `<Contest xsi:type="CandidateContest">`, used for a contest involving one or more candidates for an office
- `<Contest xsi:type="PartyContest">`, used for a contest for a straight party selection on the ballot
- `<Contest xsi:type="RetentionContest">`, used for a judicial or other type of retention contest

Contest includes ContestSelection to link the selections on the ballot to the contest, e.g., to link one or more candidates to a candidate contest. Like Contest, ContestSelection is also an abstract class and has subclasses that correspond to those of Contest, as follows:

(1) BallotMeasureContest includes BallotMeasureSelection

(2) CandidateContest includes CandidateSelection

(3) PartyContest includes PartySelection

(4) RetentionContest includes BallotMeasureSelection

Contest includes a required ElectionDistrict reference to a GpUnit defined for the geographical scope of the contest. For example, in a state senate contest, ElectionDistrict would reference a GpUnit of type ReportingUnit element defined for the district associated with the contest. Office also includes an optional reference that serves the same purpose. Note that for contests that are state-wide or county-wide and so forth, the same GpUnit defined for the state or county, etc., can be re-used.

Contest includes OtherCounts for providing a summary of miscellaneous counts associated with the contest, including total number of ballots cast containing the contest, total number of overvotes, undervotes, or write-ins. The summary counts can be associated with the contest as a whole, or with precincts or other lower-level reporting units by using multiple occurrences of OtherCounts.

SequenceOrder is used for results display ordering, i.e., to display contests according to a particular ordering. For example, "100" may indicate a U.S. Senatorial contest, "200" may indicate a U.S. Congressional contest, etc. SequenceOrder is not appropriate to use as the contest order on the ballot; contest order on each ballot can be preserved, however, using the BallotStyle element, which associates ballot styles with their corresponding precincts or other geopolitical units.

When including ExternalIdentifiers, if the type is not listed in enumeration IdentifierType, use other and include the type (that is not listed in the enumeration) in OtherType.

Figure 24 - Contest

Attribute	Multiplicity	Type	Attribute Description
Abbreviation	0..1	String	Abbreviation for the contest.
BallotSubTitle	0..1	InternationalizedText	Subtitle of the contest as it appears on the ballot.
BallotTitle	0..1	InternationalizedText	Title of the contest as it appears on the ballot.
{ContestSelection}	0..*	ContestSelection	For associating a contest selection for the contest, i.e., a candidate, a ballot measure.
CountStatus	0..*	CountStatus	For providing various counting status associated with the contest.
{ElectionDistrict}	1	ReportingUnit	Link to a GpUnit instance. For associating the contest with a reporting unit that represents the geographical scope of the contest, e.g., a district, etc.
ExternalIdentifier	0..*	ExternalIdentifier	For associating an ID with the

Attribute	Multiplicity	Type	Attribute Description
			contest.
HasRotation	0..1	Boolean	Boolean to indicate whether the selections in the contest are rotated. Assumed to be "no" if not present.
Name	1	RichText	Name of the contest, not necessarily as it appears on the ballot.
{OtherCounts}	0..*	OtherCounts	For associating counts such as overvote and undervotes with the contest.
SequenceOrder	0..1	Integer	Orderering for listing the contest for purposes of results display. If not present, no order is assumed.
SubUnitsReported	0..1	Integer	Number of subunits, e.g., precincts, that have completed reporting votes for this contest.
TotalSubUnits	0..1	Integer	Total number of subunits, e.g., precincts that have this contest on the ballot.
VoteVariation	0..1	VoteVariation	Vote variation associated with the contest, e.g., n-of-m.
OtherVoteVariation	0..1	String	For use when VoteVariation is other.

5.13 Class ContestSelection

Used for the contest selections in a contest (e.g., for candidates, for ballot measures) and to generally link them to vote counts. Contest includes ContestSelection.

ContestSelection is an abstract class with three subclasses that get used according to the type of contest:

- BallotMeasureSelection, used if the contest type is for a ballot measure, including for retentions

- CandidateSelection, used if the contest type is for one or more candidates, to link the contest selection to the candidate instances and endorsement parties; and

- PartySelection, used if the contest type is for a party, e.g., for a straight party contest. ContestSelection includes VoteCounts for associating vote counts with the contest selection.

SequenceOrder is included to specify an ordering for the contest selections for purposes of display only. The original ballot ordering can be preserved, however, by using the BallotStyle class.

Its syntax in XML is:

- *<ContestSelection xsi:type="BallotMeasureSelection">,* used if the contest type is for a ballot measure, including for retentions
- *<ContestSelection xsi:type="CandidateSelection">,* used if the contest type is for one or more candidates, to link the contest selection to the candidate elements and endorsement parties, and
- *<ContestSelection xsi:type="PartySelection">,* used if the contest type is for a party, e.g., for a straight party contest.
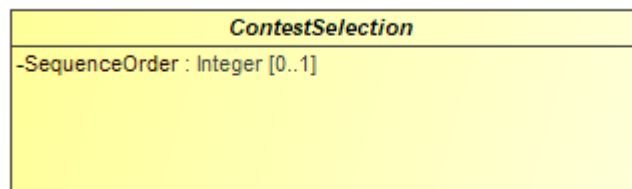
Figure 25 - ContestSelection

Attribute	Multiplicity	Type	Attribute Description
SequenceOrder	0..1	Integer	Order in which the candidate is listed on the ballot for purposes of results display. If not present, no order is assumed.
{VoteCounts}	0..*	VoteCounts	For associating the contest selection's vote counts.

5.14 Class Counts

Used for reporting on contest vote counts. Contains attributes to categorize the counts according to voting classification (e.g., election day, early voting, etc.) and type of device on which the votes were cast (e.g., DRE, accessible device, etc.).

Counts is an abstract class with two subclasses, BallotCounts, used for describing various ballot counts, and VoteCounts, used for reporting on vote counts for contest selections.
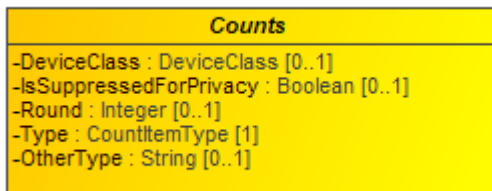
Figure 26 - Counts

Attribute	Multiplicity	Type	Attribute Description
DeviceClass	0..1	DeviceClass	For filtering counts by device type.
{GpUnit}	1	GpUnit	For filtering counts by political geography or device or device type.
IsSuppressedForPrivacy	0..1	Boolean	Boolean to indicate if votes are suppressed for voter privacy, e.g., true or false. Assumed to be false if not present.
Round	0..1	Integer	An identification of the RCV round being reported.
Type	1	CountItemType	The type of count being used as a filter on the vote counts, e.g., election day, early voting, etc.
OtherType	0..1	String	Used when Type is other.

5.15 Class CountStatus

For reporting on the counting status for various items such as ballot types or write-ins, e.g., whether for a certain type of ballot, the counts are in progress, not yet started, complete, etc. Contest, Election, and GpUnit include CountStatus.

If the type of count item is not listed in enumeration CountItemType, use "other" and include the type (that is not listed in the enumeration) in OtherType.

CountStatus
-Status : CountItemStatus [1]
-Type : CountItemType [1]
-OtherType : String [0..1]
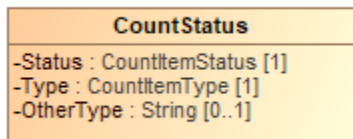
Figure 27 - CountStatus

Attribute	Multiplicity	Type	Attribute Description
Status	1	CountItemStatus	The status of the count, from the CountItemStatus enumeration.
Type	1	CountItemType	The type of item, from the CountItemType enumeration.
OtherType	0..1	String	Used when Type is other.

5.16 Class DateTimeWithZone

Restricts dateTime to require inclusion of time zone information and excludes fractional seconds.

«prim»
DateTimeWithZone

-pattern : String = [0-9]{4}-(0?[1-9]|1[012])-(0?[1-9]|[12][0-9]|3[01])T(([01][0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9]|(24:00:00))(Z|[+-]((0[0-9]|1[0-3]):[0-5][0-9]|14:00))

Figure 28 - DateTimeWithZone

Attribute	Multiplicity	Type	Attribute Description
pattern		String	Pattern used for indicating the date, time and the accompanying time zone.

5.17 Class DeviceClass

For filtering vote counts by device-related information. GpUnit, Counts, and OtherCounts include DeviceClass.
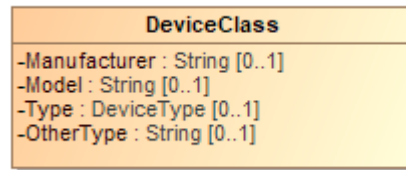
Figure 29 - DeviceClass

Attribute	Multiplicity	Type	Attribute Description
Manufacturer	0..1	String	Manufacturer of the device.
Model	0..1	String	Manufacturer's device model, used to filter on, e.g., a specific model of DRE or other device type.
Type	0..1	DeviceType	Enumerated type of device, e.g., "dre", "opscan-precinct", etc.
OtherType	0..1	String	Used when Type is other.

5.18 Class Election

For defining the status of the election and associated information such as candidates, contests, and vote counts.

Election includes links to the major instances that are specific to an election: BallotStyle, Candidate, and Contest.

Election includes a required association end ElectionScope, which links to a GpUnit instance for the purpose of identifying the geographical scope of the election. For example, for an election within a county, ElectionScope would reference a GpUnit defined for the county. If it is desired to include election authority information, the GpUnit can include ElectionAdministration.
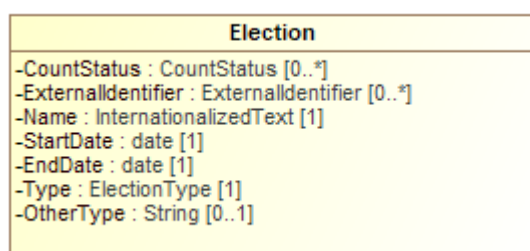
Figure 30 - Election

Attribute	Multiplicity	Type	Attribute Description
{BallotCounts}	0..*	BallotCounts	Used for identifying various ballot counts.
{BallotStyle}	0..*	BallotStyle	For defining ballot styles associated with the election.
{Candidate}	0..*	Candidate	For defining candidates associated with the election.
{ContactInformation}	0..1	ContactInformation	For associating various contact information with the election.
{Contest}	0..*	Contest	For defining contests associated with the election.
CountStatus	0..*	CountStatus	For providing various counting status on types of ballots or other items.
{ElectionScope}	1	ReportingUnit	Unique identifier for a GpUnit element. For associating the election with a reporting unit that represents the geographical scope of the election, e.g., a state, a county, etc.

46

Attribute	Multiplicity	Type	Attribute Description
ExternalIdentifier	0..*	ExternalIdentifier	For associating an ID with the election.
Name	1	InternationalizedText	For including a name for the election; the name could be the same name as appears on the ballot.
StartDate	1	date	Calendar start date of the election, e.g., "2018-11-04".
EndDate	1	date	Calendar end date of the election; for a typical one-day election, the end date is the same as the start date.
Type	1	ElectionType	Enumerated type of election, e.g., partisan-primary, open-primary, etc.
OtherType	0..1	String	Used when Type is other.

5.19 Class ElectionAdministration

Used to provide various information about an election authority. ReportingUnit includes
ElectionAdministration.

ElectionAdministration includes ContactInformation for the election authority and, using
ElectionOfficialPerson references one or more Person instances defined for individuals/organizations
associated with the election authority.
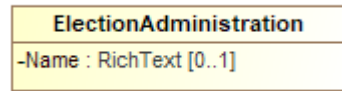
Figure 31 - ElectionAdministration

Attribute	Multiplicity	Type	Attribute Description
{ContactInformation}	0..1	ContactInformation	For including various contact information.
{ElectionOfficialPerson}	0..*	Person	Unique identifier for one or more Person elements defined for the election authority.
Name	0..1	RichText	Name of the election authority.

5.20 Class ElectionReport

For defining items pertaining to the status and format of the report and when it was generated.

ElectionReport references the major elements that are not necessarily specific to an election and that therefore can exist in a pre-election report: GpUnit, Office and OfficeGroup, Party, Person, and Election.

«Root»
ElectionReport
-ExternalIdentifier : ExternalIdentifier [0..*]
-GeneratedDate : DateTimeWithZone [1]
-Format : ReportDetailLevel [1]
-Issuer : RichText [1]
-IssuerAbbreviation : RichText [1]
-IsTest : Boolean [0..1]
-Notes : RichText [0..1]
-SequenceStart : Integer [1]
-SequenceEnd : Integer [1]
-Status : ResultsStatus [1]
-TestType : String [0..1]
-VendorApplicationId : String [1]
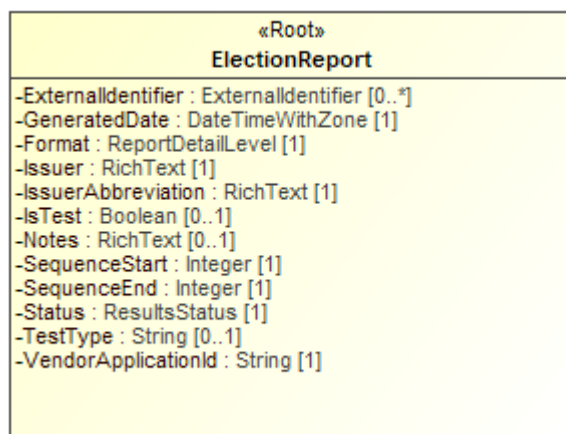
Figure 32 - ElectionReport

Attribute	Multiplicity	Type	Attribute Description
{Election}	0..*	Election	For associating elections with the report.
ExternalIdentifier	0..*	ExternalIdentifier	For associating an ID with the report.
Format	1	ReportDetailLevel	Detail level of the report, e.g., contest summary, precinct level results, etc.
GeneratedDate	1	DateTimeWithZone	Identifies the date and time that the election report was generated.
{GpUnit}	0..*	GpUnit	For associating geopolitical units with the report.
{Header}	0..*	Header	For associating headers with parts of a ballot style.
Issuer	1	RichText	Identification of the report issuer.
IssuerAbbreviation	1	RichText	An abbreviation of the report issuer such as the 2-character U.S. Census Bureau abbreviation of the state whose results are being reported, e.g., AL, TX, MN, etc.

49

Attribute	Multiplicity	Type	Attribute Description
IsTest	0..1	Boolean	Used to indicate whether the report is a test report. Assumed to be "false" if not present.
Notes	0..1	RichText	For including an arbitrary message with the report.
{Office}	0..*	Office	For associating offices with the report.
{OfficeGroup}	0..*	OfficeGroup	For associating a name for a grouping of offices with the report.
{Party}	0..*	Party	For associating parties with the report.
{Person}	0..*	Person	For associating persons with the report.
SequenceStart	1	Integer	The report's number as part of a sequence of reports, used with <SequenceEnd> so as to be read as, e.g., 1 of 1, 1 of 2, 2 of 2, etc. Starts with "1".
SequenceEnd	1	Integer	The upper bound of the sequence; e.g., "1" if there is only 1 report, "2" if there are two reports in the sequence, etc.
Status	1	ResultsStatus	Status of the election report, e.g., test mode, unofficial, etc.
TestType	0..1	String	A description of the type of test, e.g., pre-election, logic and accuracy, etc.
VendorApplicationId	1	String	An identifier of the vendor application generating the election report, e.g., X-EMS version 3.1.a.

5.21 Class ExternalIdentifier

For associating a jurisdiction's codes, i.e., identifiers, with objects such as candidates, offices, or geopolitical units such as counties, towns, precincts, etc. Multiple occurrences of the ExternalIdentifier sub-element can be used to associate multiple codes, e.g., if there is a desire to associate multiple codes with a particular object such as FIPS (Federal Information Processing Standard) codes as well as OCD-IDs (Open Civic Data Identifiers) [10]. An example in XML is:

```
<ExternalIdentifiers>
    <ExternalIdentifier>
        <Type>fips</Type>
        <Value>54</Value>
    </ExternalIdentifier>
    <ExternalIdentifier>
        <Type>ocd-id</Type>
        <Value>ocd-division/country:us/state:wv</Value>
    </ExternalIdentifier>
</ExternalIdentifiers>
```

For elements that link to ExternalIdentifier instances, if the type is not listed in enumeration IdentifierType, use "other" and include the type (that is not listed in the enumeration) in OtherType.
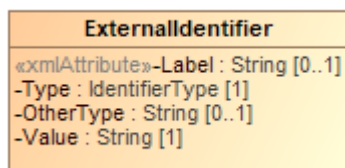
Figure 33 - ExternalIdentifier

Attribute	Multiplicity	Type	Attribute Description
Label	0..1	String	For use as needed and compatibility with the VIP schema.
Type	1	IdentifierType	An identifier type, e.g., FIPS.
OtherType	0..1	String	Used when IdentifierType value is other.
Value	1	String	The identifier used by the jurisdiction.

51

5.22 Class GpUnit

Class/element for describing a geo-politically bounded area of geography such as a city, district, or jurisdiction, or a precinct or split-precinct, or specific vote-capture device, for the purpose of associating contest vote counts and ballot counts (and other information) with the reporting unit. Reporting units can link to each other to form a hierarchiclly-oriented model of a state's (or a county's, etc.) jurisdictions, districts, and precincts.

GpUnit is an abstract class with two subclasses that get used according to the type of unit:

- ReportingDevice, used for associating vote counts with a specific vote-capture device
- ReportingUnit, for associating vote counts with geopolitical units such as cities, districts, counties, precincts, etc.

Its syntax in XML is:

- `<GpUnit xsi:type="ReportingDevice">`, for associating vote counts with a specific vote-capture device
- `<GpUnit xsi:type="ReportingUnit">`, for associating vote counts with geopolitical units such as cities, districts, counties, precincts, etc.

Election and Contest contain a required reference to GpUnit representing the jurisdiction of the election or contest respectively; Office contains a similar reference that is optional. OtherCounts and VoteCounts reference GpUnit to link vote or summary counts to GpUnit instances defined for precincts or other types of geopolitical units. BallotStyle references GpUnit to link a ballot style to its corresponding geopolitical unit.
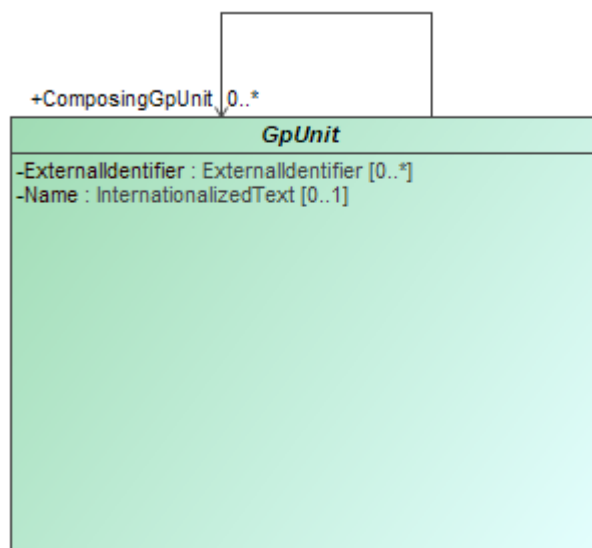
Figure 34 - GpUnit

Attribute	Multiplicity	Type	Attribute Description
{ComposingGpUnit}	0..*	GpUnit	Unique identifier for one or more GpUnit instances. For creating a reference to another GpUnit that is contained with the parent GpUnit.
ExternalIdentifier	0..*	ExternalIdentifier	For associating an ID with the GpUnit, e.g., a district's or county's code.
Name	0..1	InternationalizedText	Name of the geopolitical unit.

5.23 Class Header

For defining a reusable set of headers.
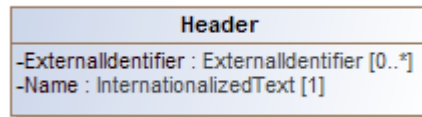
Figure 35 - Header

Attribute	Multiplicity	Type	Attribute Description
ExternalIdentifier	0..*	ExternalIdentifier	For associating an ID with the header.
Name	1	InternationalizedText	Name of the header, as it is to appear on a ballot style.

5.24 Class Hours

Hours is used to specify a specific day and hours on that day, including the time zone. Multiple occurrences of Hours can be used if the schedule includes a range of days and hours. An example in XML for specific hours on a Wednesday and Thursday is:

```
<Schedule StartDate="2018-05-15">
    <Hours Day="Friday" StartTime="09:00:00-05:00"
        EndTime="17:00:00-05:00"/>
    <Hours Day="Saturday" StartTime="09:00:00-05:00"
        EndTime="21:00:00-05:00"/>
</Schedule>
```
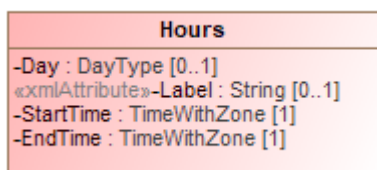

Figure 36 - Hours

Attribute	Multiplicity	Type	Attribute Description
Day	0..1	DayType	Day of the week.
Label	0..1	String	For use as needed and compatibility with the VIP schema.
StartTime	1	TimeWithZone	Start time of the schedule.
EndTime	1	TimeWithZone	End time of the schedule.

5.25 Class HtmlColorString

For a string containing a 6-digit Red-Green-Blue (RGB) code that can be displayed using HTML. Used in Party to associate a web-displayable color with the party. The RGB code is specified in hexadecimal, such that the RGB code for the color green is "00FF00" ("#00" + "#FF" + "#00").

«prim»
HtmlColorString

-pattern : String = [0-9a-f]{6}

Figure 37 - HtmlColorString

Attribute	Multiplicity	Type	Attribute Description
pattern		String	Pattern used for indicating the RGB color to use.

5.26 Class InternationalizedText

For strings that can contain multi-national text, for use with text as shown on a ballot containing multi-national text. The Identifier attribute can be used to assign an identifier to the text as desired.

Text uses the *xsd:language* type such that its language attribute must be set to a value that identifies the language. An example in XML is:

```
<InternationalizedText>
    <Text Language="es">Este es el contenido en Español.</LanguageString>
</InternationalizedText>
```

Values for language are from ISO 639 [11] and include:

(1) en – English

(2) en-US – U.S. English

(3) en-GB – U.K. English

(4) fr – French

(5) es – Spanish

(6) zh – Chinese

(7) ja – Japanese

(8) ko – Korean
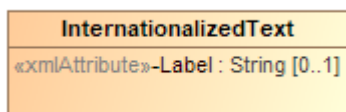
Figure 38 - InternationalizedText

Attribute	Multiplicity	Type	Attribute Description
Label	0..1	String	For use as needed and compatibility with the VIP schema.
{Text}	1..*	LanguageString	Used to hold a string of text with an associated table indicating the language used.

57

5.27 Class LanguageString

Used to hold a string of text with an associated table indicating the language used.
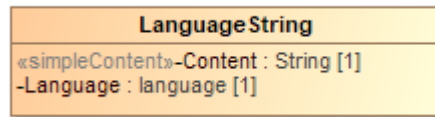
Figure 39 - LanguageString

Attribute	Multiplicity	Type	Attribute Description
Content	1	String	The string in the specified language.
Language	1	language	Identification of the language, such as 'es'.

5.28 Class LatLng

Associates latitude/longitude with a contact address.
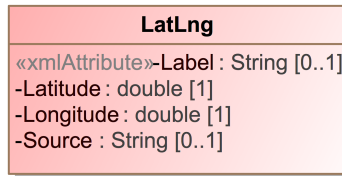
Figure 40 - LatLng

Attribute	Multiplicity	Type	Attribute Description
Label	0..1	String	For use as needed and compatibility with the VIP schema.
Latitude	1	double	Latitude of the contact location.
Longitude	1	double	Longitude of the contact location.
Source	0..1	String	System used to perform the lookup from location name to lat/lng, e.g., the name of a geocoding service.

5.29 Class Office

For defining an office and information associated with a contest and/or a district. ElectionReport includes Office. CandidateContest and RetentionContest reference Office.

Office includes Term for defining details about the term of an office such as start/end dates and the type of term. OfficeGroup is included from ElectionReport to assign a name to a grouping of office definitions.

Office includes an optional ElectionDistrict reference to a GpUnit for the purpose of identifying the geographical scope of the office. For example, for an office for a state senate seat, ElectionDistrict would include a reference to the GpUnit defined for the district associated with that office.
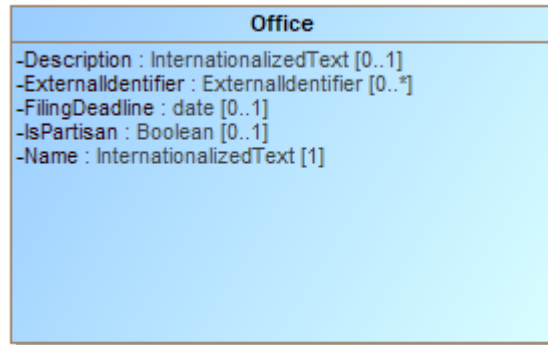
Figure 41 - Office

Attribute	Multiplicity	Type	Attribute Description
{ContactInformation}	0..1	ContactInformation	For associating various contact information with the office.
Description	0..1	InternationalizedText	A description of the office, possibly as shown on the ballot to the voter.
{ElectionDistrict}	0..1	ReportingUnit	Link to a GpUnit instance. For associating the office with a reporting unit that represents the geographical scope of the contest, e.g., a district, etc.
ExternalIdentifier	0..*	ExternalIdentifier	For associating an ID with the office.
FilingDeadline	0..1	date	Date and time when a candidate must have filed for the contest for the office.

Attribute	Multiplicity	Type	Attribute Description
IsPartisan	0..1	Boolean	Boolean to indicate whether the office is partisan, e.g., true or false. If not present, assumption is true.
Name	1	InternationalizedText	Name of the office; can appear on the ballot.
{OfficeHolderPerson}	0..*	Person	Links to one or more Person instances defined for the office holder.
{Term}	0..1	Term	For including office term-related information.

5.30 Class OfficeGroup

Used to assign a name to a grouping of office definitions. It includes references to Office instances and a name to identify the grouping of references, e.g., "Judicial" or "Statewide", etc. SubOfficeGroup can be used to create a nested hierarchy of groupings. ElectionReport includes OfficeGroup.

As an example in XML, *<OfficeGroup>* can be used to group state offices, etc., as follows:

```
<OfficeGroup>
    <Name>State House Offices</Name>
    <SubOfficeGroup>
        <Name>Upper House Offices</Name>
        <OfficeIds>121iqw 122iqw 123iqw</OfficeIds>
    </SubOfficeGroup>
    ...
</OfficeGroup>
```
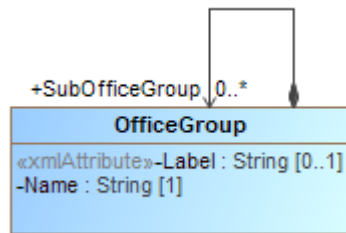

Figure 42 - OfficeGroup

Attribute	Multiplicity	Type	Attribute Description
Label	0..1	String	For use as needed and compatibility with the VIP schema.
Name	1	String	Name of the office grouping.
{Office}	0..*	Office	Link to one or more Office instances.
{SubOfficeGroup}	0..*	OfficeGroup	For defining a nested hierarchy of Office instance groupings.

5.31 Class OrderedContent

An abstract base class for content that can appear under a particular ballot style.

OrderedContent is an abstract class with two subclasses that get used according to the type of content:

- OrderedContest, used for the appearance of a contest.

- OrderedHeader, used for the appearance of a header, optionally with the inclusion of contests.
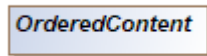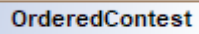
Figure 43 - OrderedContent

5.32 Class OrderedContest

For the appearance of a contest on a particular ballot style.

Figure 44 - OrderedContest

Attribute	Multiplicity	Type	Attribute Description
{Contest}	1	Contest	The contest associated represented by OrderedContest.
{OrderedContestSelection}	0..*	ContestSelection	The contest selections for the ballot.

5.33 Class OrderedHeader

For the appearance of a header on a particular ballot style.

Figure 45 - OrderedHeader

Attribute	Multiplicity	Type	Attribute Description
{Header}	1	<u>Header</u>	Association to the header to be used.
{OrderedContent}	0..*	<u>OrderedContent</u>	For associating a header with ballot content, such as contests or nested headers.

5.34 Class OtherCounts

Identifies other counts associated with a contest.
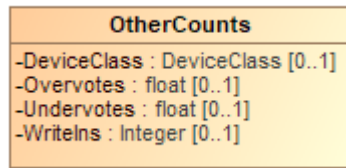
Figure 46 - OtherCounts

Attribute	Multiplicity	Type	Attribute Description
DeviceClass	0..1	DeviceClass	For filtering counts by device type.
{GpUnit}	1	GpUnit	For filter counts by political geography or device or device type.
Overvotes	0..1	float	Number of overvotes.
Undervotes	0..1	float	Number of undervotes.
WriteIns	0..1	Integer	Number of write-ins.

5.35 Class Party

Used to describe a political party that can then be referenced in other elements. ElectionReport includes Party. Candidate, PartyContest, PartyRegistration, and Person reference Party.

Party is an abstract type with one subtype Coalition, used to define coalitions. Its syntax in XML is:

> *<Party xsi:type="Coalition">, used to define coalitions*

The Color attribute specifies a 6-digit RGB code displayable using HTML. As an example in XML, the color green is encoded as:

> *<Party Color="00FF00" ... />*
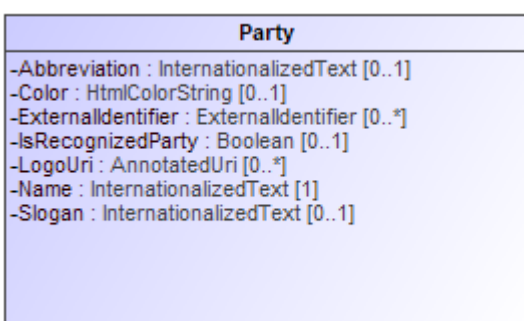
Figure 47 - Party

Attribute	Multiplicity	Type	Attribute Description
Abbreviation	0..1	InternationalizedText	Short name for the party, e.g., "DEM".
Color	0..1	HtmlColorString	For associating an HTML RGB color coding with the party.
{ContactInformation}	0..1	ContactInformation	For associating contact information regarding the party, e.g., party offices.
ExternalIdentifier	0..*	ExternalIdentifier	For associating an ID with the party.
IsRecognizedParty	0..1	Boolean	For indicating whether the party is recognized by the election authority; "false" is assumed if not present.
{LeaderPerson}	0..*	Person	Identification of a Party's leader.
LogoUri	0..*	AnnotatedUri	A URI to the party's graphical

Attribute	Multiplicity	Type	Attribute Description
			logo.
Name	1	InternationalizedText	Official full name of the party, e.g., "Republican"; can appear on the ballot.
{PartyScopeGpUnit}	0..*	GpUnit	The GpUnit(s) the party operates in or the top-most GpUnit.
Slogan	0..1	InternationalizedText	The slogan or motto used by a political party.

5.36 Class PartyContest

For a contest that involves choosing a party, typically for a straight party selection on the ballot. Its syntax in XML is:

```
<Contest xsi:type="PartyContest" ... />
```
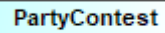

Figure 48 - PartyContest

5.37 Class PartyRegistration

For tracking the number of registered voters per party per geopolitical unit, i.e., for reporting on the number of registered voters of a particular party in a district or other type of reporting unit. Referenced by GpUnit.

PartyRegistration
-Count : Integer [1]
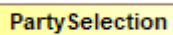
Figure 49 - PartyRegistration

Attribute	Multiplicity	Type	Attribute Description
Count	1	Integer	A count for tracking the number of registered voters.
{Party}	1	Party	Link to a Party instance. For associating a political party.

5.38 Class PartySelection

For a contest selection involving a party such as for a straight party selection on the ballot. It inherits the attributes of ContestSelection. Its syntax in XML is:

```
<ContestSelection xsi:type="PartySelection" ... />
```


Figure 50 - PartySelection

Attribute	Multiplicity	Type	Attribute Description
{Party}	1..*	Party	Link to one or more Party instances. For associating one or more parties with the party selection.

5.39 Class Person

For defining information about a person; the person may be a candidate, election official, authority for a reporting unit, etc. ElectionReport includes Person. Candidate and ElectionAdministration and

GpUnit references Person. Person optionally references ContactInformation for associating contact information.

Multiple occurrences of the MiddleName attribute can be used as needed, e.g., for names such as "John Andrew Winston Smith". An example in XML is:

```
<Person ObjectId="P12321">
    <FirstName>John</FirstName>
    <MiddleName>Andrew</MiddleName>
    <MiddleName>Winston</MiddleName>
    <LastName>Smith</LastName>
</Person>
```


Person
-DateOfBirth : date [0..1]
-ExternalIdentifier : ExternalIdentifier [0..*]
-FirstName : RichText [0..1]
-FullName : InternationalizedText [0..1]
-Gender : RichText [0..1]
-LastName : RichText [0..1]
-MiddleName : RichText [0..*]
-Nickname : RichText [0..1]
-Prefix : RichText [0..1]
-Profession : InternationalizedText [0..1]
-Suffix : RichText [0..1]
-Title : InternationalizedText [0..1]
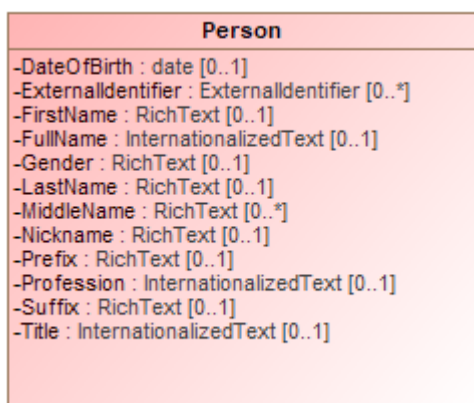
Figure 51 - Person

Attribute	Multiplicity	Type	Attribute Description
{ContactInformation}	0..*	ContactInformation	For associating contact information with the person.
DateOfBirth	0..1	date	Person's date of birth.
ExternalIdentifier	0..*	ExternalIdentifier	For associating codes with the person.
FirstName	0..1	RichText	Person's first (given) name.
FullName	0..1	InternationalizedText	Person's full name.
Gender	0..1	RichText	Person's gender.

72

Attribute	Multiplicity	Type	Attribute Description
LastName	0..1	RichText	Person's last (family) name.
MiddleName	0..*	RichText	Person's middle name.
Nickname	0..1	RichText	Nickname associated with the person.
{Party}	0..1	Party	Links to a Party instance. For associating a political party with the person.
Prefix	0..1	RichText	A prefix associated with the person, e.g., Mr.
Profession	0..1	InternationalizedText	Person's profession.
Suffix	0..1	RichText	A suffix associated with the person, e.g., Jr.
Title	0..1	InternationalizedText	A title associated with the person.

5.40 Class ReportingDevice

Class/element describing a specific vote-capture device. Its syntax in XML is:

```
<GpUnit xsi:type="ReportingDevice" ... />
```

```
ReportingDevice
-DeviceClass : DeviceClass [0..1]
-SerialNumber : RichText [0..1]
```

Figure 52 - ReportingDevice

Attribute	Multiplicity	Type	Attribute Description
DeviceClass	0..1	DeviceClass	Used for reporting on details about the type of voting device used for the results in question.
SerialNumber	0..1	RichText	Device's serial number or other unique identifier.

5.41 Class ReportingUnit

For defining a geopolitical unit such as state, county, township, precinct, etc., using the ReportingUnitType enumeration. It inherits the attributes of GpUnit. Its syntax in XML is:

```
<GpUnit xsi:type="ReportingUnit" ... />
```

This class optionally references Person to associate one or more individuals, e.g., authorities, for the reporting unit. It also includes ContactInformation to provide contact addresses for the reporting unit, such as an address of a vote center.

Election references this class so as to identify the geographical scope of the election. In this case, the GpUnit element defined for the scope of the election may include ElectionAdministration to include election authority-related information.

The Type attribute uses the ReportingUnitType enumeration to specify the type of geopolitical geography being defined. ReportingUnitType contains the most common types of geographies, e.g., state, county, precinct, and so forth. If the reporting unit type is not listed in enumeration ReportingUnitType, use "other" and include the reporting unit type (that is not listed in the enumeration) in OtherType.

The IsDistricted boolean can be used in a number of ways. It is not strictly necessary, as it is possible to identify districts by their Type attribute or by examining the Contest instance's ElectionDistrict reference, which links to the election district associated with the contest. However, if a district is defined but is not linked from a contest, or if the type of district is not listed in the ReportingUnitType enumeration and therefore OtherType is used, then IsDistricted is necessary to identify the GpUnit as a district. The IsDistricted boolean can also be used to signify that a GpUnit defined as a jurisdiction, e.g., a county, is also used as a district for, e.g., county-wide contests.

```
                    ReportingUnit
-CountStatus : CountStatus [0..*]
-IsDistricted : Boolean [0..1]
-IsMailOnly : Boolean [0..1]
-Number : String [0..1]
-SubUnitsReported : Integer [0..1]
-TotalSubUnits : Integer [0..1]
-Type : ReportingUnitType [1]
-OtherType : String [0..1]
-VotersParticipated : Integer [0..1]
-VotersRegistered : Integer [0..1]
```
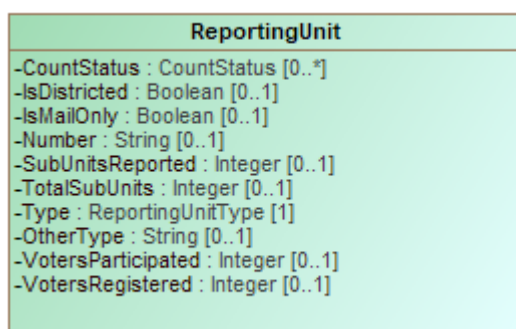
Figure 53 - ReportingUnit

Attribute	Multiplicity	Type	Attribute Description
{Authority}	0..*	Person	A link to one or more Person instances describing

Attribute	Multiplicity	Type	Attribute Description
			an authority responsible for the reporting unit.
{ContactInformation}	0..1	ContactInformation	For associating contact information with the reporting unit.
CountStatus	0..*	CountStatus	For providing various counting status on types of ballots or other items.
{ElectionAdministration}	0..1	ElectionAdministration	For use when the reporting unit serves as the authority in the election.
IsDistricted	0..1	Boolean	Boolean to indicate whether the reporting unit is a district; assumed to be "false" if not present.
IsMailOnly	0..1	Boolean	Boolean to indicate whether the reporting unit handles only mail-in or absentee ballot elections, assumed to be "false" if not present.
Number	0..1	String	A number associated with the reporting unit; for compatibility with VIP.
{PartyRegistration}	0..*	PartyRegistration	For associating a count of registered voters per party with the geopolitical unit.
{SpatialDimension}	0..1	SpatialDimension	For describing the reporting unit's spatial extent (a polygon that shows the related area).
SubUnitsReported	0..1	Integer	Number of associated subunits such as precincts that have completed reporting.
TotalSubUnits	0..1	Integer	Total number of associated subunits such as precincts.
Type	1	ReportingUnitType	Enumerated type of reporting unit, e.g., state, county, district, precinct, etc.

Attribute	Multiplicity	Type	Attribute Description
OtherType	0..1	String	For use when ReportingUnitType value is other.
VotersParticipated	0..1	Integer	Number of voters who have participated in the election, i.e., shown up at the polls, including those who did not cast ballots.
VotersRegistered	0..1	Integer	Number of registered voters residing within the boundaries of the geopolitical unit.

5.42 Class RetentionContest

For judicial retention or other types of retention contests. Retention contests can be treated essentially as ballot measure contests, however this element differs from BallotMeasureContest in that it can include a reference to a candidate or the associated office.

This element uses BallotMeasureContest as a superclass. Therefore, it inherits the attributes of Contest as well as BallotMeasureContest. Its syntax in XML is:

```
<Contest xsi:type="RetentionContest" ... />
```
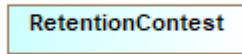

Figure 54 - RetentionContest

Attribute	Multiplicity	Type	Attribute Description
{Candidate}	1	Candidate	Link to a Candidate instance. For associating a candidate with the retention contest.
{Office}	0..1	Office	Link to an Office instance. For associating an office description with the retention contest.

5.43 Class Schedule

For defining a schedule associated with a particular election office or location. ContactInformation includes Schedule.
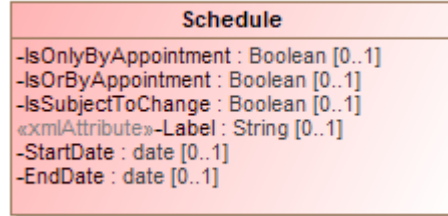
Figure 55 - Schedule

Attribute	Multiplicity	Type	Attribute Description
{Hours}	0..*	Hours	For specifying a range of hours for a schedule.
IsOnlyByAppointment	0..1	Boolean	If an appointment is only by appointment; assumed to be "no" if not present.
IsOrByAppointment	0..1	Boolean	If an appointment can by appointment presumably as desired; assumed to be "no" if not present.
IsSubjectToChange	0..1	Boolean	If an appointment may be subject to change; assumed to be "no" if not present.
Label	0..1	String	For use as needed and compatibility with the VIP schema.
StartDate	0..1	date	For the starting date of the schedule.
EndDate	0..1	date	For the ending date of the schedule.

5.44 Class ShortString

For defining a 32-character annotation, used with character strings in AnnotatedString.
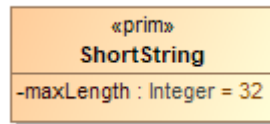
Figure 56 - ShortString

Attribute	Multiplicity	Type	Attribute Description
maxLength		Integer	The maximum allowed length of a ShortString.

5.45 Class SpatialDimension

For defining the spatial layout of a GpUnit, e.g., a map or a spatial extent (a polygon that shows the related area) for various purposes, including to visualize election results, to understand the composition of districts, or to determine whether GpUnit instances are properly related. GpUnit includes SpatialDimension.

Figure 57 - SpatialDimension

Attribute	Multiplicity	Type	Attribute Description
MapUri	0..*	AnnotatedUri	Typically a URI to a map of the GpUnit.
{SpatialExtent}	0..1	SpatialExtent	For associating a GpUnit element's spatial extent information.

81

5.46 Class SpatialExtent

SpatialDimension includes SpatialExtent for defining a GpUnit instance's spatial extent data and the format used for the spatial extent.

Figure 58 - SpatialExtent

Attribute	Multiplicity	Type	Attribute Description
Coordinates	1	RichText	The data coordinates constituting the spatial extent.
Format	1	GeoSpatialFormat	Enumerated type for the format used, e.g., gml, kml, wkt, etc.

5.47 Class Term

For describing information about an office term. Term is included by Office.
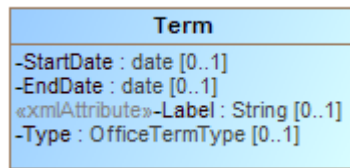
Figure 59 - Term

Attribute	Multiplicity	Type	Attribute Description
StartDate	0..1	date	Start date for the current term of the office.
EndDate	0..1	date	End date for the current term of the office.
Label	0..1	String	For use as needed and compatibility with the VIP schema.
Type	0..1	OfficeTermType	Enumerated type of term, e.g., full-term, unexpired-term, etc.

5.48 Class TimeWithZone

Restricts time to require inclusion of time zone information and excludes fractional seconds.

Figure 60 - TimeWithZone

Attribute	Multiplicity	Type	Attribute Description
pattern		String	Pattern used for indicating the time with the accompanying time zone.

5.49 Class VoteCounts

For reporting on vote counts for contest selections in a contest. VoteCounts includes Counts as an extension base and therefore inherits the elements from Counts, but it is included directly by ContestSelection.
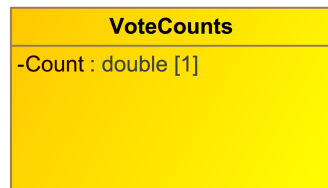
Figure 61 - VoteCounts

Attribute	Multiplicity	Type	Attribute Description
Count	1	double	Count of contest votes cast; can include a factional component in special cases.

5.50 Enumeration BallotMeasureType

Enumeration for types of ballot measures in the BallotMeasureContest element.
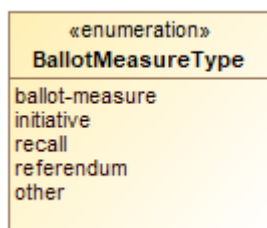
Figure 62 - BallotMeasureType

Value	Value Description
`ballot-measure`	For reports that contain only aggregated counts.
`initiative`	For an initiative.
`recall`	For a recall.
`referendum`	For a referendum.
`other`	Used when the type of ballot measure is not included in this enumeration.

5.51 Enumeration CandidatePostElectionStatus

Enumeration for various post-election statuses applicable to a candidate in the Candidate element.
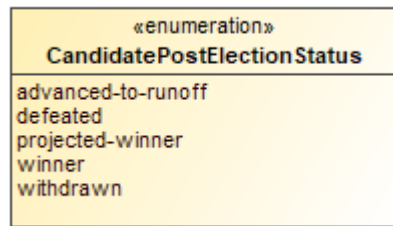
Figure 63 - CandidatePostElectionStatus

Value	Value Description
advanced-to-runoff	For candidates who have advanced to a runoff.
defeated	Used for candidates who were defeated in the election.
projected-winner	For a projected contest winner.
winner	For the official contest winner or one of "n" contest winners for n-of-m voting.
withdrawn	For candidates who have withdrawn from the contest.

5.52 Enumeration CandidatePreElectionStatus

Enumeration for various pre-election statuses applicable to a candidate in the Candidate class.
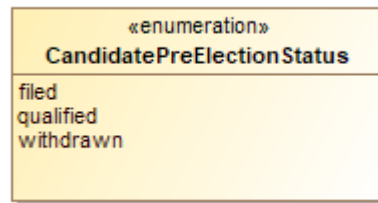
Figure 64 - CandidatePreElectionStatus

Value	Value Description
filed	For candidates who have filed with the election authority but not necessarily qualified.
qualified	For candidates who are qualified by the election authority to be on the ballot for a contest.
withdrawn	For candidates who have withdrawn from the contest.

5.53 Enumeration CountItemStatus

Enumeration for various counting-related statuses for types of ballots or write-ins, used in the CountStatus class.
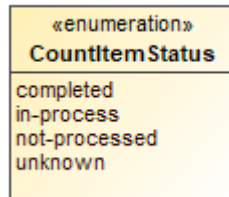
Figure 65 - CountItemStatus

Value	Value Description
completed	For counts that are complete.
in-process	For counts that are in process.
not-processed	When the counting has not started or is not underway.
unknown	When the status of the counting is unknown.

5.54 Enumeration CountItemType

Enumeration for the items that are counted during the course of an election and for which the status of the counts is of interest. Used in the Counts and CountStatus classes.
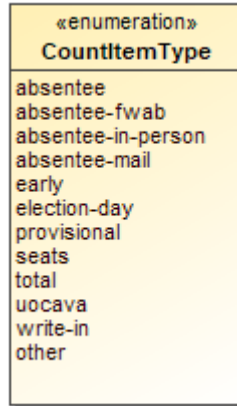
Figure 66 - CountItemType

Value	Value Description
absentee	For any/all classes of absentee, generally when absentee is not broken out into specific classes.
absentee-fwab	A type of absentee; for Federal Write-in Absentee Ballots.
absentee-in-person	A class of absentee; for absentee ballots cast in-person, e.g., at a county office.
absentee-mail	A class of absentee; for postal mail absentee ballots separately.
early	For ballots cast during early voting periods.
election-day	For ballots cast on election day.
provisional	For challenged ballots.
seats	For legislative balance-of-power results information.
total	Total of all ballots cast regardless of voting class.
uocava	A class of absentee; for absentee ballots from Uniformed and Overseas Citizens Absentee Voting Act (UOCAVA) voters.
write-in	For write-ins on ballots.
other	Used when the type of counting item is not included in this enumeration.

5.55 Enumeration DayType

Enumeration for the day(s) in a schedule in the Schedule element.

Figure 67 - DayType

Value	Value Description
all	Used for all days of the week.
sunday	Used if day of week is Sunday.
monday	Used if day of week is Monday.
tuesday	Used if day of week is Tuesday.
wednesday	Used if day of week is Wednesday.
thursday	Used if day of week is Thursday.
friday	Used if day of week is Friday.
saturday	Used if day of week is Saturday.
weekday	Used for any day of the week.
weekend	Used for both Saturday and Sunday.

5.56 Enumeration DeviceType

Enumeration for the type of device in the DeviceClass class.
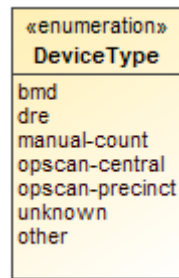
Figure 68 - DeviceType

Value	Value Description
bmd	For ballots prepared on ballot marking devices and then scanned.
dre	For DRE (Direct Record Electronic) and other all-electronic devices.
manual-count	For hand-counted paper ballots.
opscan-central	For an optical scanner used at a central office with no opportunity for voter correction of mistakes.
opscan-precinct	For an optical scanner used at a precinct or other location where voter correction of mistakes such as overvotes is possible.
unknown	Used when the type of device is unknown.
other	Used when the device type is not listed in this enumeration.

5.57 Enumeration ElectionType

Enumeration for the type of election in the ElectionReport class.
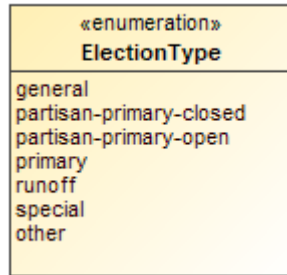
Figure 69 - ElectionType

Value	Value Description
general	Election in which all eligible voters, regardless of party affiliation, are permitted to select candidates to fill public office and/or vote on ballot measures.
partisan-primary-closed	Primary election in which the voter receives a ballot containing only those party-specific contests pertaining to the political party with which the voter is affiliated, along with non-party-specific contests presented at the same election. Unaffiliated voters may be permitted to vote only on non-party-specific contests.
partisan-primary-open	Primary election in which the voter may choose a political party at the time of voting and vote in party-specific contests associated with that party, along with non-party-specific contests presented at the same election. Some states require voters to publicly declare their choice of party at the polling place, after which the election worker provides or activates the appropriate ballot. Other states allow the voters to make their choice of party within the privacy of the voting booth.
primary	Election held to determine which candidates qualify to appear as contest options in subsequent elections.
runoff	Election to select a winner following a primary or a general election, in which no candidate in the contest received the required minimum percentage of the votes cast. The two candidates receiving the most votes for the contest in question proceed to a runoff election.
special	Primary or general election that is not regularly scheduled. A special election may be combined with a scheduled election.
other	Used when the election type is not listed in this enumeration.

5.58 Enumeration GeoSpatialFormat

Enumeration for geospatial vector data formats used in Geographic Information System (GIS) software, used in the SpatialExtent class.
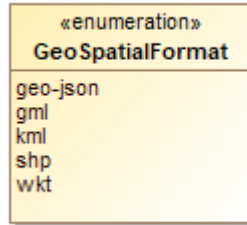
Figure 70 - GeoSpatialFormat

Value	Value Description
geo-json	For GeoJSON open standard format.
gml	For Geography Markup Language format.
kml	For Keyhole Markup Language format.
shp	For the shape file format associated with Esri.
wkt	For Well-known Text format.

5.59 Enumeration IdentifierType

Enumeration for election data-related codes in the ExternalIdentifier class .
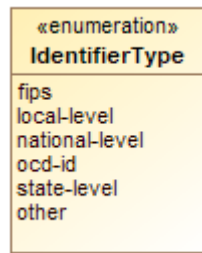
Figure 71 - IdentifierType

Value	Value Description
fips	For FIPS codes.
local-level	For a code that is specific to a county or other similar locality.
national-level	For a code that is used at the national level other than "fips" or "ocd-id".
ocd-id	For Open Civic Data identifiers.
state-level	For a code that is specific to a state.
other	Used when the type of code is not included in this enumeration.

5.60 Enumeration OfficeTermType

Enumeration for the office term type in the Office class.
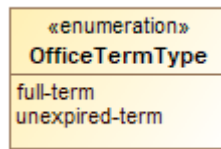
Figure 72 - OfficeTermType

Value	Value Description
full-term	When the officeholder's term began at the beginning of the full term of the office, e.g., 6 years for U.S. Senate.
unexpired-term	When the officeholder's term began at some date after the beginning of the full term of the office, generally because the previous officeholder vacated the office before the fullterm expired.

5.61 Enumeration ReportDetailLevel

Enumeration for the detail level of the election results report in the Election class.
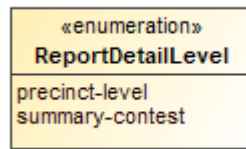
Figure 73 - ReportDetailLevel

Value	Value Description
precinct-level	For reports that contain counts from precincts in the reporting jurisdiction.
summary-contest	For reports that contain only aggregated counts.

5.62 Enumeration ReportingUnitType

Enumeration for the type of geopolitical unit in the ReportingUnit class.

Figure 74 - ReportingUnitType

Value	Value Description
`ballot-batch`	Used for reporting batches of ballots that may cross precinct boundaries.
`ballot-style-area`	Used for ballot style areas generally composed of precincts.
`borough`	Used in CT, NJ, PA, other states, and New York City for boroughs. For AK and LA, see county.
`city`	Used for a city that reports results and/or for the district that encompasses it.
`city-council`	Used for city council districts.
`combined-precinct`	Used for one or more precincts that have been combined for the purposes of reporting. Used for "Ward" if "Ward" is used interchangeably with "CombinedPrecinct".
`congressional`	Used for U.S. Congressional districts.

Value	Value Description
country	Used for a reporting unit of type country.
county	Used for a county and/or for the district that encompasses it. In AK, used for counties that are called boroughs. In LA, used for parishes.
county-council	Used for county council districts.
drop-box	Used for a dropbox for absentee ballots.
judicial	Used for judicial districts.
municipality	Used as applicable for various units such as towns, townships, villages that report votes and/or for the district that encompasses it.
polling-place	Used for a polling place.
precinct	Used also for "Ward" or "District" when these terms are used interchangeably with "Precinct".
school	Used for a school district.
special	Used for a special district.
split-precinct	Used for splits of precincts.
state	Used for a state and/or for the district that encompasses it.
state-house	Used for a state house or assembly district.
state-senate	Used for a state senate district.
town	Used in some New England states as a type of municipality that reports votes and/or for the district that encompasses it.
township	Used in some mid-western states as a type of municipality that reports votes and/or for the district that encompasses it.
utility	Used for a utility district.
village	Used as a type of municipality that reports votes and/or for the district that encompasses it.
vote-center	Used for a vote center.
ward	Used for combinations or groupings of precincts or other units.
water	Used for a water district.
other	Used for other types of reporting units not included in this enumeration.

5.63 Enumeration ResultsStatus

Enumeration for the status of the election results in the ElectionReport class.
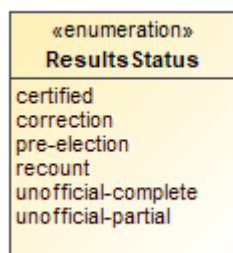
Figure 75 - ResultsStatus

Value	Value Description
certified	For results that have been certified by the election authority.
correction	For results that are a correction to an earlier report.
pre-election	For a pre-election data.
recount	For results that are a recount of an earlier election.
unofficial-complete	For results that are unofficial and complete, e.g., the complete election night results.
unofficial-partial	For results that are unofficial and partial, e.g., partial election night results.

5.64 Enumeration VoteVariation

Enumeration for contest decision algorithm or rules in the <u>Contest</u> element.
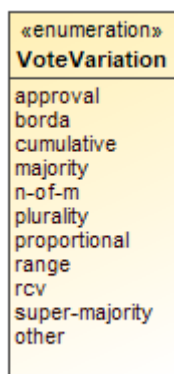
Figure 76 - VoteVariation

Value	Value Description
approval	When voter can select as many candidates as desired in a contest up to a maximum number.
borda	For the Borda count voting.
cumulative	When voter can allocate more than one vote to a given candidate.
majority	For majority voting.
n-of-m	Includes vote for 1, i.e., 1-of-m.
plurality	For plurality voting.
proportional	For proportional voting.
range	For range voting.
rcv	For ranked choice voting.
super-majority	For super majority voting.
other	Used when the vote variation type is not included in this enumeration.

6   XML Schema Aspects and Usage

This section presents an overview of various aspects and usage of the XML schema, including:

1. General layout of instance files,
2. How UML relationships are implemented in the XML schema,
3. How geopolitical geography is implemented,
4. How contests and contest selections are implemented,
5. How vote counts and ballot counts are implemented, and
6. How ballot counts are linked to geopolitical geography

No attempt is made to provide a complete overview of usage, but more to describe in general how most elements are to be used.

6.1   General Layout of XML Instance Files

Instance files are built by creating collections of elements such for office descriptions, geopolitical units, candidates, contests, etc., and then linking the elements as necessary using the *ObjectId* attributes.
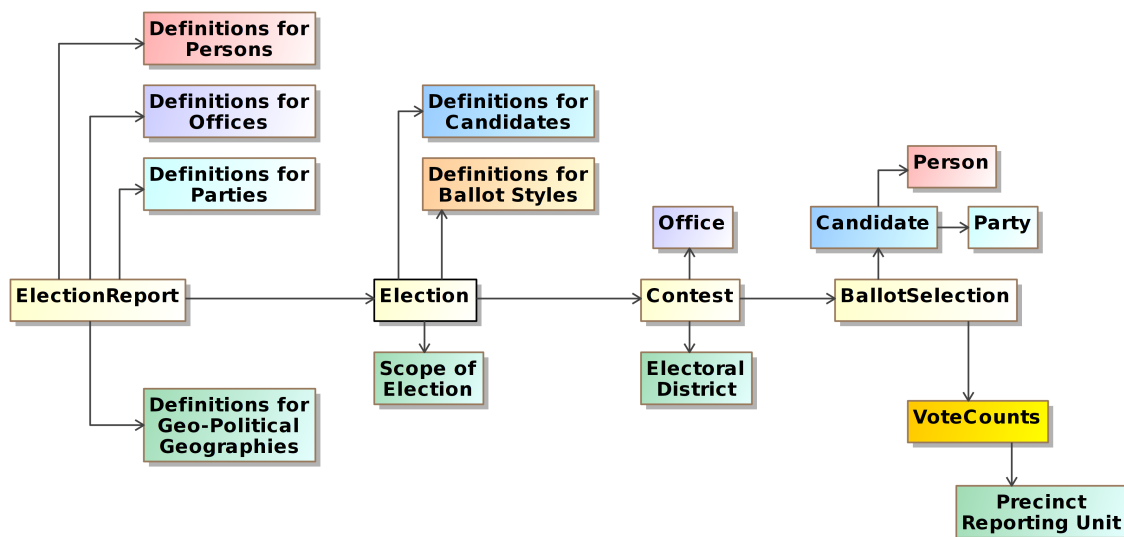
Figure 77 – General Layout of Instance Files

So as to accommodate all use cases (pre-election, election night, and post-election), many elements in the schema are optional, thus instance files will vary in content but generally follow the same structure as shown in Figure 77 and listed below:

- *<ElectionReport>* element, containing:
    o *<Election>* element(s);
    o *<GpUnit>* elements;

      o   *<Office>* and *<OfficeGroup>* elements;

      o   *<Party>* elements; and

      o   *<Person>* elements;

- *<Election>* element(s), containing:
  - o   *<BallotStyle>* elements;
  - o   *<Candidate>* elements;
  - o   *<Contest>* element for Contest 1:
    - ▪   *<ContestSelection>* elements for the contest selections in Contest 1;
      - • *<VoteCounts>* elements associated with the contest selections:
      - • *<GpUnit>* elements for associating vote counts with precincts;
  - o   *<Contest>* element for Contest 2, etc.

Many of the elements make use of enumerated values, and most enumerations include a value of *other* for use when none of the enumerated values is applicable, and the elements using these enumerations include an *<OtherType>* element that can be used to hold another value.

6.1.1   How UML Relationships Are Implemented in the XML Schema

The XML schema is generated from the UML model, thus having some understanding of how the XML is generated will assist users of the schema as well as others who may wish to extend the model or schema for other purposes. This section provides a brief overview of the schema generation.

When XML is generated from the UML model, the classes in the model get generated as major XML elements, e.g., the UML Candidate class results in the XML *<Candidate>* element. The attributes in the UML classes get generated as XML sub-elements within the XML major element, e.g., the UML attribute Name in the Candidate class gets generated as the sub-element *<Name>* in the *<Candidate>* major element.

The schema treats major elements such as candidates or contests or parties as if they are tables in a database that are linked together depending on the type of relationship between the objects. The relationships between UML classes result in XML elements that function differently depending upon the type of relationship. As a brief guide, using Figure 8, the directed composition relationship between Election and Contest results in one or more XML sub-elements named *<Contest>* being created in *<Election>*.

```
<Contest xsi:type="CandidateContest" ObjectId="OID-AG">
   <Name>Attorney General</Name>
   ...
</Contest>
```

Figure 78 – "Is a Type of" Relationship in XML

"Is a Type of" or "Instance of" relationships are slightly more complex. For Contest and CandidateContest in Figure 8, CandidateContest is a type of Contest and thus is generated with *<Contest>* as an XML extension base and all attributes and elements of *<Contest>* are global to elements that use *<Contest>* as an extension base. The XML syntax for using *<Contest>* for a candidate contest is shown in Figure 78.

"Is a Type of" relationships are used in cases such as *<Contest>* where there can be several types of contests, but the different types of contests generally require the same set of attributes plus additional attributes that are specific to a type. Thus, *<Contest>* is generated as an abstract XML element with all of its sub-elements and attributes global to the specific types of contests, e.g., *CandidateContest*. But, the sub-elements of *CandidateContest* are private to *CandidateContest*.

6.1.2   Directed Association Relationships

Directed association relationships require more explanation. Directed association relationships are used throughout the UML model and in the XML schema to reference or link one element to another, and they give XML instance files the capability to be structured and used as tables in a database, with pointers in one table referencing other tables as needed. An example of a directed association is the *<PartyId>* sub-element in the *<Candidate>* element, which serves to link a candidate with the candidate's party.

First, when the XML schema is generated, every class in the UML model that is at the endpoint of a directed association relationship is generated with an *ObjectId* attribute of type *xsd:ID*. For example, *<Party>* is referenced from a number of classes, thus the *<Party>* XML element is generated with the *ObjectId* attribute. Type *xsd:ID* requires that *ObjectId* be a unique legal XML value that does not contain a colon, and digits, the hyphen, and the period may not be used as the starting character.

Secondly, for the class at the starting point of the directed association, an element is generated that serves to reference the element at the endpoint. Furthermore, the name of the element is the name of the class at the endpoint with "Id" concatenated to it.

For Candidate and Party from Figure 8, the directed association relationship results in an XML sub-element named *<PartyId>* of type *xsd:IDREF* being created for *<Candidate>*. The *<PartyId>* sub-element links to a specific *<Party>* element that will be associated with that candidate, thus linking the candidate to the candidate's party. The usage is shown below:

```
<Party ObjectId="OID-IND">
   <Name>
      <Text Language="en-us">Independent</Text>
   </Name>
</Party>
...
...
<Candidate ObjectId="OID-C1">
   <BallotName>
      <Text Language="en-us">John Brown</Text>
   </BallotName>
   <PartyId>OID-IND</PartyId>
</Candidate>
```

Figure 79 – Directed Association in XML

In this example, "OID-IND" is the value of the *ObjectId* attribute for the *<Party>* element defined for the Independent Party. To link candidate John Brown to this *<Party>* element, the corresponding *<Candidate>* element uses *<PartyId>* and the same "OID-IND" value.

6.2 Defining Geopolitical Geography

A primary feature of the schema is the geopolitical geography element *<GpUnit>*, which can be used in a variety of ways to exactly mirror a jurisdiction's geopolitical geography. Vote counts in the real world are reported by geopolitical geographies, thus *<GpUnit>* elements are used in an election report to list the geopolitical units of geography that lay within the geographic scope of the election results such as counties, cities, precincts, districts, etc., so that ballot and vote counts can be associated with them. The UML model and XML schema permit counts to be reported at a high aggregation level only if desired, thus *<GpUnit>* elements are required in two places:

1. The *<Contest>* element contains a required link to a *<GpUnit>* defined for the election district of the contest, so that vote counts can be linked to that *<GpUnit>* element. The *<ElectionDistrictId>* sub-element is used to link to the *<GpUnit>* element defined for the election district.

2. The *<Election>* element contains a required link to a *<GpUnit>* defined for the geographical scope of the election, again using the *<ElectionScopeId>* sub-element.

If counts need to be associated with the smaller geographies, i.e., precincts, additional *<GpUnit>* elements need to be defined for these geographies. *<GpUnit>* elements, as described in section can be used for the following major types of geographies:

- Geographies that report election results or that can have vote counts associated with them, including for states, counties, cities, towns, precincts, split precincts, ballot style areas, etc.;
- Geographies that serve as election districts, i.e., that have contests associated with them;
- Geographies that serve as districts that have no contests associated (e.g., tax districts); and
- Geographies that are for specific locations or specific objects including:
  - Voting devices;
  - Polling places or vote centers;
  - Drop boxes; and
  - Other miscellaneous objects.

Additionally, *<GpUnit>* elements can be defined for other types of objects not listed in the *<ReportingUnitType>* enumeration by using *other* in the *<Type>* sub-element and then listing the other type of object in the *<OtherType>* sub-element.

6.2.1 Defining <GpUnit> Elements for Governmental and Administrative-based Geographies

Figure 80[2] shows an example of geopolitical geography in the town of Richmond, WI. Figure 81 shows the *<GpUnit>* definitions for the governmental and administrative-based geographies (in Wisconsin, a ward is the same as a precinct). The assumption here is that the election results will be reported by these geographies (county, town, and ward), thus accordingly there are *<GpUnit>*

---

[2] This figure is repeated from Figure 7.

definitions for these geographies. The county needs to be linked with the state, the town needs to be linked with the county, the combined wards need to be linked with the town, the five precincts need to be linked with their parent combined precincts, and the two split precinct elements need to be linked with their parent Ward 1 precinct. Thus, the `<GpUnit>` definitions for the parent geographies include `<ComposingGpUnitIds>` to reference the geographies that are contained within the parent elements.
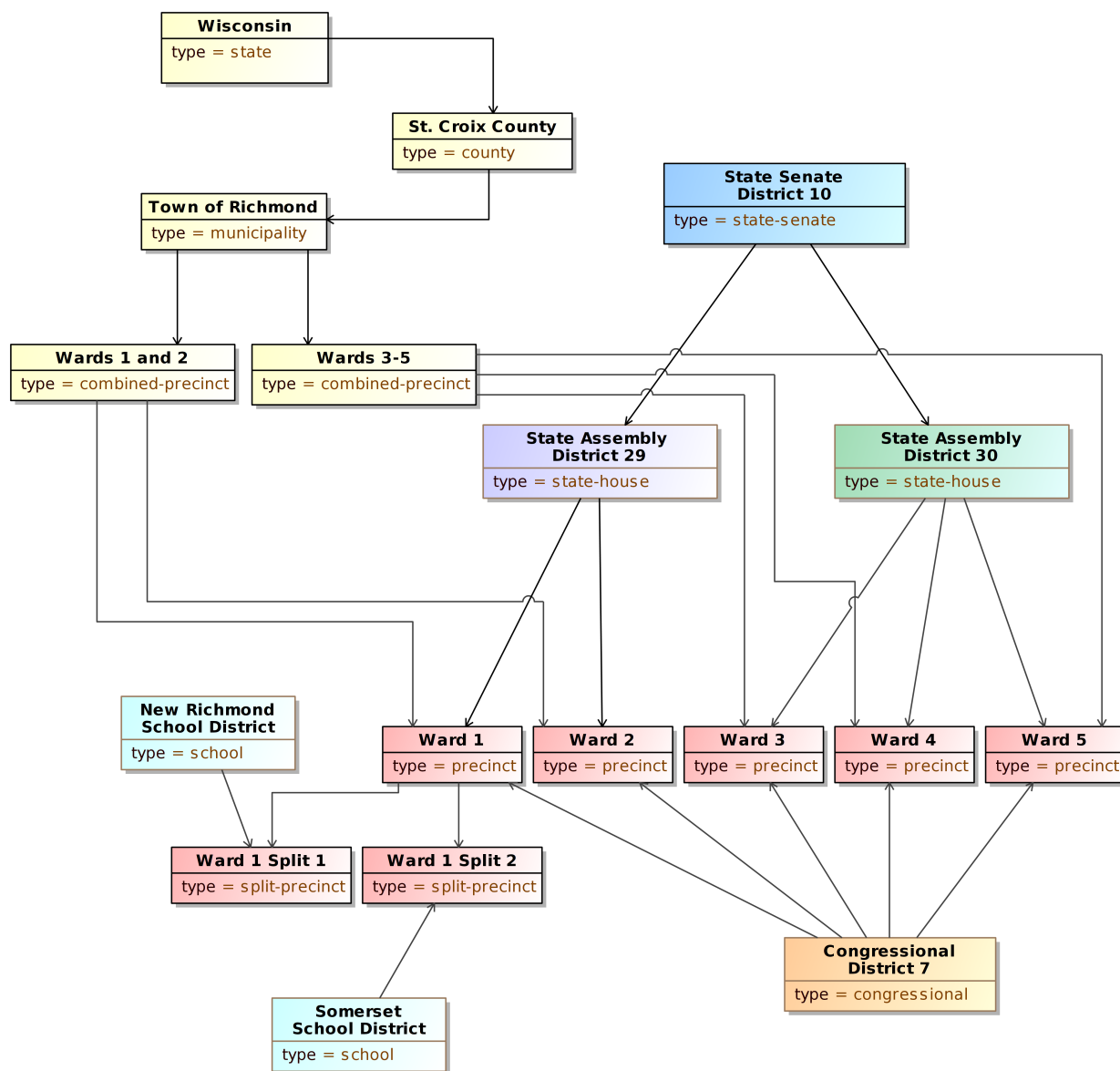
Figure 80 – Geopolitical Geography Example from Wisconsin

However, there is a rule that **must** be observed when defining these relationships:

> *A <GpUnit> element's <ComposingGpUnitIds> element **must only** reference those geopolitical units that are wholly contained within the scope of the <GpUnit> element.*

```
<!-- Definitions for the state, county, and city -->
   <GpUnit xsi:type="ReportingUnit" ObjectId="OID-State">
      <Name>
         <Text language="en">State of Wisconsin</Text>
      </Name>
      <ComposingGpUnitIds>OID-ST-CROIX</ComposingGpUnitIds>
      <Type>state</Type>
   </GpUnit>
   <GpUnit xsi:type="ReportingUnit" ObjectId="OID-ST-CROIX">
      <Name>
         <Text language="en">St. Croix County</Text>
      </Name>
      <ComposingGpUnitIds>OID-Richmond</ComposingGpUnitIds>
      <Type>county</Type>
   </GpUnit>
   <GpUnit xsi:type="ReportingUnit" ObjectId="OID-Richmond">
      <Name>
         <Text language="en">Town of Richmond</Text>
      </Name>
      <ComposingGpUnitIds>OID-WARDS1-2 OID-WARDS3-5</ComposingGpUnitIds>
      <Type>municipality</Type>
   </GpUnit>
   <!--Definitions for combined wards -->
   <GpUnit xsi:type="ReportingUnit" ObjectId="OID-WARDS1-2">
      <Name>
         <Text language="en">Combined Wards 1 and 2</Text>
      </Name>
      <ComposingGpUnitIds>OID-WARD1 OID-WARD2</ComposingGpUnitIds>
      <Type>combined-precinct</Type>
   </GpUnit>
   <GpUnit xsi:type="ReportingUnit" ObjectId="OID-WARDS3-5">
      <Name>
         <Text language="en">Combined Wards 3-5</Text>
      </Name>
      <ComposingGpUnitIds>OID-WARD3 OID-WARD4 OID-WARD5</ComposingGpUnitIds>
      <Type>combined-precinct</Type>
   </GpUnit>
   <!--Definitions for precincts and split precincts -->
   <GpUnit xsi:type="ReportingUnit" ObjectId="OID-WARD1">
      <Name>
         <Text language="en">Ward 1</Text>
      </Name>
      <ComposingGpUnitIds>OID-WARD1-1 OID-WARD1-2</ComposingGpUnitIds>
      <Type>precinct</Type>
   </GpUnit>
   <GpUnit xsi:type="ReportingUnit" ObjectId="OID-WARD1-1">
      <Name>
         <Text language="en">Ward 1-1</Text>
      </Name>
      <Type>split-precinct</Type>
   </GpUnit>
   <GpUnit xsi:type="ReportingUnit" ObjectId="OID-WARD1-2">
      <Name>
         <Text language="en">Ward 1-2</Text>
      </Name>
      <Type>split-precinct</Type>
   </GpUnit>
   <GpUnit xsi:type="ReportingUnit" ObjectId="OID-WARD2">
      <Name>
         <Text language="en">Ward 2</Text>
      </Name>
      <Type>precinct</Type>
   </GpUnit>
   <GpUnit xsi:type="ReportingUnit" ObjectId="OID-WARD3">
```

```
    <Name>
        <Text language="en">Ward 3</Text>
    </Name>
    <Type>precinct</Type>
</GpUnit>
<GpUnit xsi:type="ReportingUnit" ObjectId="OID-WARD4">
    <Name>
        <Text language="en">Ward 4</Text>
    </Name>
    <Type>precinct</Type>
</GpUnit>
<GpUnit xsi:type="ReportingUnit" ObjectId="OID-WARD5">
    <Name>
        <Text language="en">Ward 5</Text>
    </Name>
    <Type>precinct</Type>
</GpUnit>
```

Figure 81 - <GpUnit> Definitions for Governmental and Administrative Geographies

Accordingly, split precincts are wholly contained within precincts, thus the proper way to structure the associated *<GpUnit>* elements is for the precinct element to contain a *<ComposingGpUnitIds>* element that references the split precinct elements – and not vice versa. If this rule is violated, an application would not be able to find information correctly within the instance file. Figure 82 shows a hierarchy of containment relationships, with precincts and splits being the foundational bricks for the geographies.
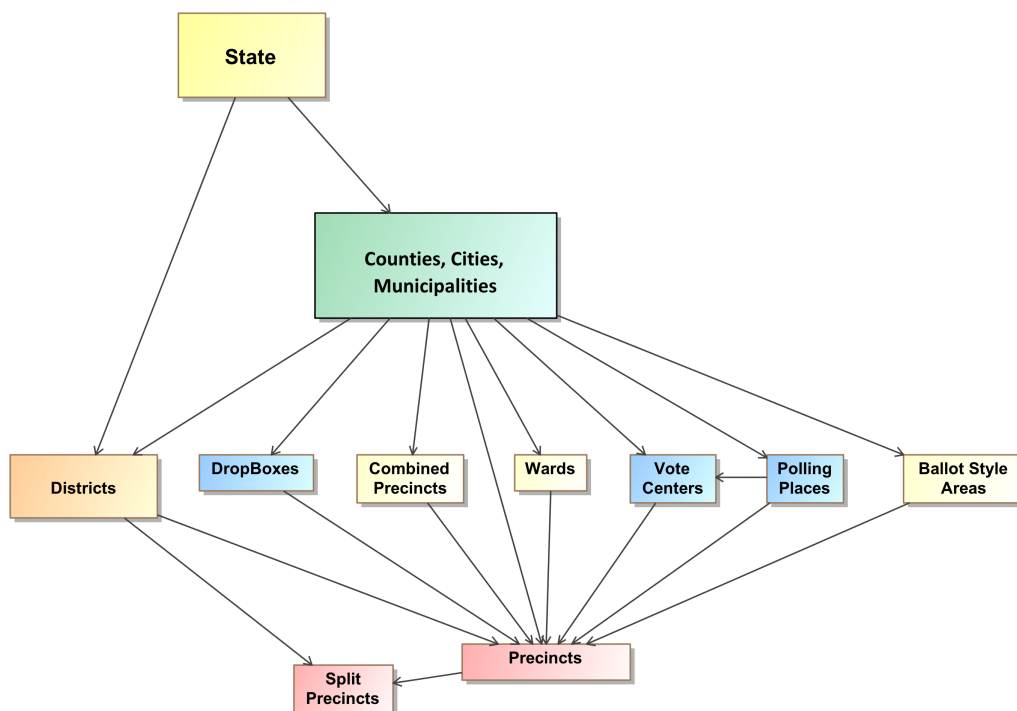
Figure 82 – Containment Relationships

6.2.2 Associating an Election Authority with the <GpUnit> Element Defined for the Scope of the Election

In Figure 80, the State of Wisconsin was assumed as the scope of the election, and in Figure 81, a *<GpUnit>* element was defined for the state. Figure 83 shows the definition for the state, but this time with the election authority information added.

```
   <GpUnit xsi:type="ReportingUnit" ObjectId="OID-State">
      <ElectionAdministration>
         <ContactInformation>
            <Name>Wisconsin Election Authority</Name>
            <Uri>https://wisconsin-demo-election-url.wisconsin.gov</Uri>
         </ContactInformation>
         <ElectionOfficialPersonIds>OID</ElectionOfficialPersonIds>
         <Name>State of Wisconsin Demo Election</Name>
      </ElectionAdministration>
      <ComposingGpUnitIds>OID-ST-CROIX</ComposingGpUnitIds>
      <Name>
         <Text language="en">State of Wisconsin</Text>
      </Name>
      <Type>state</Type>
   </GpUnit>
```

Figure 83 – Associating Election Authority Information

The *<Election>* element references a *<GpUnit>* element defined for the scope of the election, and that *<GpUnit>* element can then include the *<ElectionAdministration>* element for associating election authority information. *<ElectionAdministration>* includes *<ContactInformation>* for the election authority and, using *<ElectionOfficialPersonIds>*, references one or more *<Person>* elements defined for individuals/organizations associated with the election authority.

6.2.3 Defining <GpUnit> Elements for Political-based Geographies

Figure 84 shows the districts and the wards that compose them from Figure 80. Figure 85 shows the *<GpUnit>* definitions for the districts as well as the use of *<ComposingGpUnitIds>* elements to link the districts to the wards that compose them.

The *<IsDistricted>* boolean is used in this example. It is not strictly necessary, as it is possible to identify districts by their type or by examining the *<Contest>* element's *<ElectionDistrictId>* sub-element, which links to the election district associated with the contest. However, if a district is defined but is not linked from a contest, or if the type of district is not listed in the *<ReportingUnitType>* enumeration and therefore *<OtherType>* is used, then *<IsDistricted>* is necessary to identify the *<GpUnit>* as a district. The *<IsDistricted>* boolean can also be used to signify that a *<GpUnit>* defined as a jurisdiction, e.g., a county, is also used as a district for, e.g., county-wide contests.
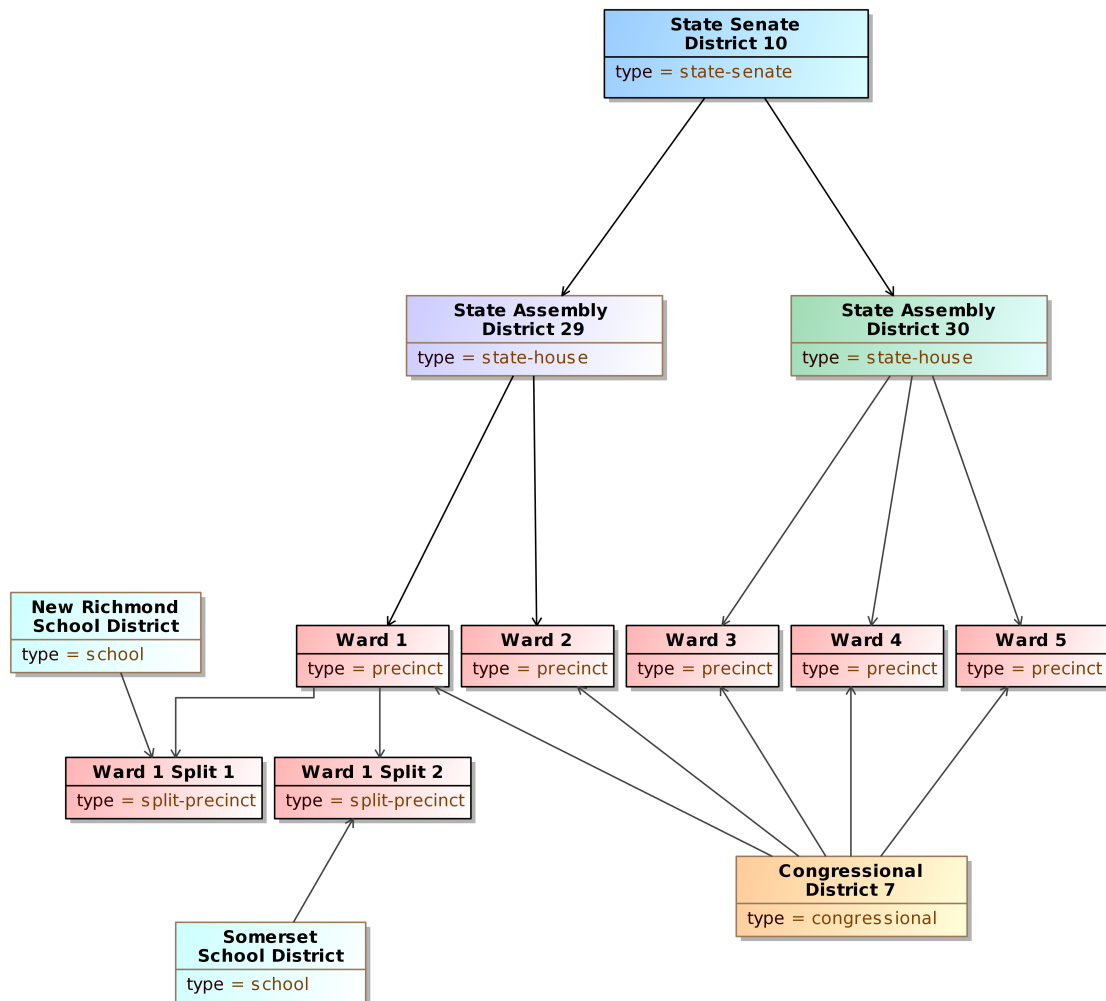
Figure 84 – District <GpUnit> Elements

```
<!--Definition for State Senate District 10 -->
<GpUnit xsi:type="ReportingUnit" ObjectId="OID-SD10">
   <ComposingGpUnitIds>OID-HD29 OID-HD30</ComposingGpUnitIds>
   <IsDistricted>true</IsDistricted>
   <Name>
      <Text language="en">State Senate District 10</Text>
   </Name>
   <Type>state-senate</Type>
</GpUnit>
<!--Definition for House Districts 29 and 30 -->
<GpUnit xsi:type="ReportingUnit" ObjectId="OID-HD29">
   <ComposingGpUnitIds>OID-WARD1 OID-WARD2</ComposingGpUnitIds>
   <IsDistricted>true</IsDistricted>
   <Name>
      <Text language="en">House District 29</Text>
   </Name>
   <Type>state-house</Type>
</GpUnit>
<GpUnit xsi:type="ReportingUnit" ObjectId="OID-HD30">
   <ComposingGpUnitIds>OID-WARD3 OID-WARD4 OID-WARD5</ComposingGpUnitIds>
   <IsDistricted>true</IsDistricted>
   <Name>
      <Text language="en">House District 30</Text>
   </Name>
   <Type>state-house</Type>
</GpUnit>
<!--Definition for Congressional District 7 -->
<GpUnit xsi:type="ReportingUnit" ObjectId="OID-HD30">
   <ComposingGpUnitIds>OID-WARD1 OID-WARD2 OID-WARD3 OID-WARD4 OID-WARD5</ComposingGpUnitIds>
   <IsDistricted>true</IsDistricted>
   <Name>
      <Text language="en">Congressional District 7</Text>
   </Name>
   <Type>congressional</Type>
</GpUnit>
<!--Definition for school districts -->
<GpUnit xsi:type="ReportingUnit" ObjectId="OID-NewRichmondSchool">
   <ComposingGpUnitIds>OID-WARD1-1</ComposingGpUnitIds>
   <IsDistricted>true</IsDistricted>
   <Name>
      <Text language="en">New Richmond School District</Text>
   </Name>
   <Type>school</Type>
</GpUnit>
<GpUnit xsi:type="ReportingUnit" ObjectId="OID-SomersetSchool">
   <ComposingGpUnitIds>OID-WARD1-2</ComposingGpUnitIds>
   <IsDistricted>true</IsDistricted>
   <Name>
      <Text language="en">New Somerset School District</Text>
   </Name>
   <Type>school</Type>
</GpUnit>
```

Figure 85 – Defining Political Geographies and Linking to Precincts

6.2.4   Dealing with Duplicate Election Districts

There are, quite often, multiple jurisdictional-wide contests that use, for their respective election districts, the same physical geography. In a given county, for example, contests for county executive and at-large county council positions will all use the county as the election district. Because *<GpUnit>* elements defined for these geographies may have very large *<ComposingGpUnitIds>* sub-elements,

instance files could grow unnecessarily large if different but essentially duplicate *<GpUnit>* elements are defined for each of the contests that share what is essentially the same election district.
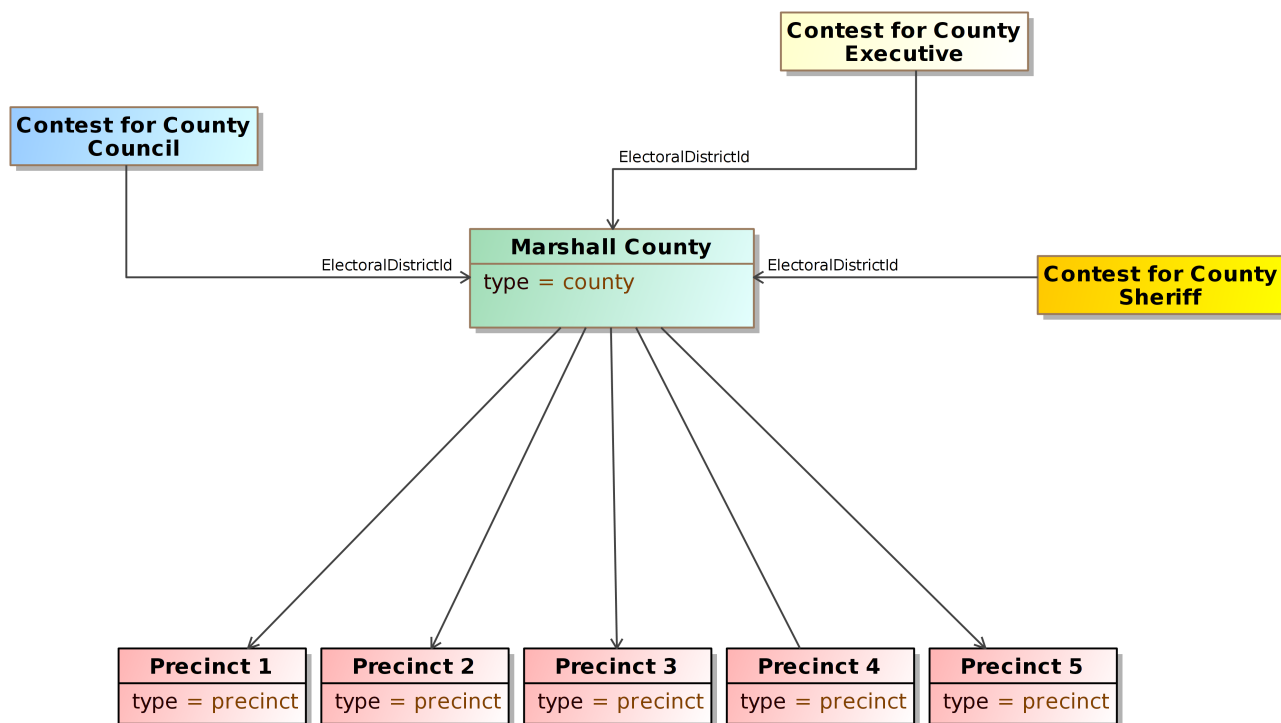
Figure 86 – Re-Using Election District Links

An option for reducing the file size is to re-use the *<GpUnit>* element defined for the county. Provided that this element is linked to its composing precincts, the contests that are county-wide can use this same *<GpUnit>* element for the election district. The generating application could, effectively, keep track of the *ObjectId* attribute of the county *<GpUnit>* element and then use the associated value for each contest's *<ElectionDistrictId>* sub-element – see Figure 86.

Thus, if all contest definitions for jurisdictional-wide contests use the same value for the *<ElectionDistrictId>* sub-element, duplication can be reduced and file sizes may be smaller, especially for larger jurisdictions.

6.3   Defining Contests

After the *<GpUnit>* elements have been defined, contests can then be defined using the *<Contest>* element. Defining contests involves the use of several elements, depending on what type of contest is involved and whether and what types of vote counts are desired. The elements needed for contest definitions are as follows:

1. The *<Contest>* abstract element, to identify the name of the contest and other attributes and its type:
    a. *CandidateContest*, for a contest involving candidates;
    b. *BallotMeasureContest*, for a contest involving a ballot measure;

     c. *PartyContest*, for a contest involving a straight party selection; and

     d. *RetentionContest,* for a contest involving a judicial or other retention;

2. The *<ContestSelection>* abstract element, to identify a selection on the ballot for that corresponding contest and its type:

     a. *CandidateSelection*, if the selection is for a candidate;

     b. *BallotMeasureSelection*, if the selection is for a ballot measure or retention; and

     c. *PartySelection*, if the selection is for a straight party selection;

3. Within *<ContestSelection>*, *<VoteCounts>* to report the number of votes for the contest selection, for the contest as a whole or broken out by, e.g., precincts and/or types of ballots;

4. A link to a GpUnit element defined for the contest's election district, using the *<ElectionDistrictId>* sub-element; and

5. The *<OtherCounts>* element, to report summary counts for the contest, e.g., overvotes, undervotes, write-ins.

It is possible to define contests without associated *<ContestSelection>* elements as well as without any of the elements for election results, for example, for a pre-election report. However, these elements are required if associating vote counts with contest selections.

```
<Candidate ObjectId="OID-C1">
  <BallotName>
    <Text Language="en">John Brown</Text>
  </BallotName>
</Candidate>
<Candidate ObjectId="OID-C2">
  <BallotName>
    <Text Language="en">Thomas Paine</Text>
  </BallotName>
</Candidate>
...
<!--Governor contest -->
<Contest xsi:type="CandidateContest" ObjectId="OID-GOV">
  <!--Vote count for John Brown -->
  <ContestSelection xsi:type="CandidateSelection" ObjectId="OID-CS1">
    <VoteCounts>
      <Count>2716</Count>
    </VoteCounts>
    <CandidateIds>OID-C1</CandidateIds>
  </ContestSelection>
  <!--Vote count for Thomas Paine -->
  <ContestSelection xsi:type="CandidateSelection" ObjectId="OID-CS2">
    <VoteCounts>
      <Count>2392</Count>
    </VoteCounts>
    <CandidateIds>OID-C2</CandidateIds>
  </ContestSelection>
  <ElectionDistrictId>OID-State</ElectionDistrictId>
  <Name>Governor</Name>
  <VotesAllowed>1</VotesAllowed>
</Contest>
```

Figure 87 – Basic Contest Definition and Linkage to Vote Counts

PartyContest is included as a contest type because a number of states keep track of the number of straight party selections made by voters.

CandidateContest is the type of contest likely to be used most often; it is for contests involving candidates and has several elements associated with it including *<PrimaryPartyIds>*, a link to the political party used when a primary election, *<NumberElected>*, for the number of seats associated with the office and *<VotesAllowed>*, for the number of selections a voter can make in the contest.

Figure 87 shows an example of a contest involving *CandidateContest* that illustrates using contest selections and counts, with the counts being aggregated, that is, associated with the election district of the contest as a whole. Contests for *BallotMeasureContest* and *PartyContest* are implemented as per this example.

6.4   Associating Vote Counts with Contests

The XML schema includes the capability to associate vote counts with contests in a variety of ways, including the following:

1. As aggregated vote counts (applying to the entire election district):
   - The vote counts can be broken out (or filtered by) ballot type, device type, and by specific device and manufacturer;
2. As subsets of vote counts, each associated with specific geopolitical units within the election district such as precincts or split precincts:
   - The vote counts can be broken out by ballot type, device type, and by specific device and manufacturer;
3. As summary vote counts that can be aggregated and associated with geopolitical units:
   - The summaries can be broken out by ballot type, device type, and by specific device and manufacturer; and
   - The summaries can include:
     - Total number of overvotes and undervotes in the contest; and
     - Total number of write-ins in the contest.

The following sections go into more detail on how to report counts accordingly.

6.4.1   Aggregated Vote Counts

The previous section showed an example of basic contest definition involving aggregated vote counts applying to the entire election district. To go into greater level of detail and report the aggregated counts by ballot type or device type, one includes additional values from the *<CountItemType>* or *<DeviceType>* enumerations. These enumerations specify different ballot types or device items that can be used as filters on the aggregated results. An example of this is shown in Figure 88, using the contest for governor and values from the *<CountItemType>* enumeration. This example answers the questions, how many votes did John Brown get for the contest as a whole, filtered by ballots cast on election day, by ballots cast during the early voting period, or by absentee ballots.

```
<!--Governor contest -->
<Contest xsi:type="CandidateContest" ObjectId="OID-GOV">

    <!--Vote counts for John Brown broken down by ballot type -->
    <ContestSelection xsi:type="CandidateSelection" ObjectId="OID-CS1">
        <VoteCounts>
            <Type>election-day</Type>
            <Count>2716</Count>

        </VoteCounts>
        <VoteCounts>
            <Type>early</Type>
            <Count>700</Count>           </VoteCounts>
        <VoteCounts>
            <Type>absentee</Type>
            <Count>16</Count>
        </VoteCounts>
        <CandidateIds>OID-C1</CandidateIds>
    </ContestSelection>

</Contest>
```

Figure 88 – Linking Aggregated Vote Counts to Contests

6.4.2   Vote Counts by Precinct, Ballot, and Device Type

For associating vote counts broken down by geopolitical units such as precincts and split precincts, one uses the *<GpUnitId>* sub-element to link the counts to *<GpUnit>* elements defined for, the precincts and split precincts.   For example, to associate the vote count for John Brown by ward from Figure 80, the XML is:

```
<!--Governor contest -->
<Contest xsi:type="CandidateContest" ObjectId="OID-GOV">

    <!--Vote count for John Brown broken down by ward -->
    <ContestSelection xsi:type="CandidateSelection" ObjectId="OID-CS1">
        <VoteCounts>
            <GpUnitId>OID-WARD1-1</GpUnitId>
            <Count>627</Count>
        </VoteCounts>
        <VoteCounts>
            <GpUnitId>OID-WARD1-2</GpUnitId>
            <Count>325</Count>
        </VoteCounts>
        <VoteCounts>
            <GpUnitId>OID-WARD2</GpUnitId>
            <Count>432</Count>
        </VoteCounts>

        <CandidateId>OID-C1</CandidateId>

    </ContestSelection>
```

Figure 89 – Linking Vote Counts to Precincts

In this example, note that the vote counts for John Brown are broken down, in Ward 1, for Ward 1's splits (as opposed to breaking it down by Ward 1 as a whole). Since all of Ward 1 is in the election district for the Governor's contest, it is not necessary to break down the vote count by the split

precincts of Ward 1, but doing so would answer questions such as, "How did the voters in the New Richmond or Somerset school districts vote in the Governor's contest?"

To break the vote counts down further by ballot type or device type, one again uses values from the *<CountItemType>* or *<DeviceType>* enumerations. Using Ward 2 from the previous example, the XML is shown in Figure 90. Using the two examples, John Brown received 300 votes in Ward 1 on election day using precinct-count optical scanner. He received 132 votes during the early voting period, again using a precinct-count optical scanner.

```
<!--Governor contest -->
<Contest xsi:type="CandidateContest" ObjectId="OID-GOV">
    <!--Vote count for John Brown broken down by ward -->
    <ContestSelection xsi:type="CandidateSelection" ObjectId="OID-CS1">
        <VoteCounts>
            <DeviceClass>
                <Type>opscan-precinct</Type>
            </DeviceClass>
            <GpUnitId>OID-WARD2</GpUnitId>
            <Type>election-day</Type>
            <Count>300</Count>
        </VoteCounts>
        <VoteCounts>
            <DeviceClass>
                <Type>opscan-precinct</Type>
            </DeviceClass>
            <GpUnitId>OID-WARD2</GpUnitId>
            <Type>early</Type>
            <Count>132</Count>
        </VoteCounts>
        <CandidateIds>OID-C1</CandidateIds>
    </ContestSelection>
</Contest>
```

Figure 90 – Filtering Vote Counts by Device and Ballot Type

6.4.3   Other Counts

It is possible to associate other counts for the contest for various purposes including for reconciliations and checks on the contest. These counts include:

- Total number of overvotes and undervotes in the contest, and
- Total number of write-ins in the contest.

As with previous examples, it is possible to further filter these counts by device type, using values from the *<DeviceType>* enumeration. It is also possible to link the counts to specific geopolitical units such as precincts and split precincts by including the *<GpUnitId>* sub-element to link the counts to *<GpUnit>* elements defined for, the precincts and split precincts.

Figure 91 shows an example of other counts for a contest using OtherCounts; the counts are further broken down by ward. Some jurisdictions report items such as overvotes and undervotes as candidates in a contest as opposed to reporting them separately as a summary of counts. This schema does not prevent one from treating overvotes, undervotes, or other types of counts as candidates in a contest, thus one would report their specific vote counts.

```
   <Contest xsi:type="CandidateContest" ObjectId="OID-GOV">
      <ContestSelection ...
      ...
      <ElectionDistrictId>OID-State</ElectionDistrictId>
      <Name>Governor</Name>
      ...
      <OtherCounts>
         <GpUnitId>OID-WARD3</GpUnitId>
         <Overvotes>6013</Overvotes>
         <Undervotes>1000</Undervotes>
         <WriteIns>37</WriteIns>
      </OtherCounts>
      <OtherCounts>
         <GpUnitId>OID-WARD3</GpUnitId>
         <Overvotes>133</Overvotes>
         <Undervotes>10</Undervotes>
         <WriteIns>3</WriteIns>
      </OtherCounts>
      ...
   </Contest>
```

Figure 91 – Associating Summary Counts with Contests


```
   <Election>

      <BallotCounts>
         <BallotsCast>8033</BallotsCast>
         <BallotsOutstanding>113</BallotsOutstanding>
         <BallotsRejected>1010</BallotsRejected>
      </BallotCounts>
      <BallotCounts>
         <DeviceClass>
            <Type>opscan-precinct</Type>
         </DeviceClass>
         <BallotsCast>2013</BallotsCast>
         <BallotsOutstanding>100</BallotsOutstanding>
         <BallotsRejected>1000</BallotRejected>
      </BallotCounts>
      <BallotCounts>
         <DeviceClass>
            <Type>bmd</Type>
         </DeviceClass>
         <BallotsCast>6020</BallotsCast>
         <BallotsOutstanding>13</BallotsOutstanding>
         <BallotsRejected>10</BallotsRejected>
      </BallotCounts>

   </Election>
```

Figure 92 – Associating Summary Ballot Counts with <Election> Elements

6.5   Associating Ballot Counts with Geopolitical Units

Lastly, it is possible to associate the ballots cast in each geopolitical unit and filter them by ballot types and device types by including the *<BallotCounts>* element. The ballot counts include:

- Total number of ballots cast at that geopolitical unit,
- Total number of ballots outstanding at that geopolitical unit,

- Total number of ballots rejected at that geopolitical unit,
- Broken out by:
    o Type of ballot,
    o Type of device, or
    o Specific device serial number and manufacturer.

Thus, one can associate ballot counts for each precinct, etc., and, as possible, go down to the level of specific devices. To do this, the *<BallotCounts>* element needs to be added to the *<Election>* definition, as shown in Figure 92.

6.6   Defining and Using Re-usable Ballot and Contest Headers

Headings are used on ballots to delineate divisions or sections of a ballot, for example:

- For Contests: Partisan Section, Proposal Section
- For Jurisdictions: City, County, State, National
- For Offices: Judicial, Legislative

Headings can appear without an explicit relationship to other content (i.e., for a contest). This could be used for contest-independent headers (e.g. "GENERAL ELECTION", "OFFICIAL BALLOT", etc.) as well as for footers.

A contest listed on the ballot has heading information that can be included in the *<BallotTitle>* and *<BallotSubTitle>* sub-elements of *<Contest>*. By using the *<BallotStyle>* element, it is also possible to create election-independent contest headings as well as other election-independent ballot headings that can be defined once and then re-used across multiple ballot styles, the advantage being that less space is used and the generation and handling of the XML and JSON content is simplified.

Under *<Election>,* each heading is defined once, as an instance of *<Header>*, and accessible through the use of directed associations from *<OrderedHeader>*.

Contests can also be associated with headers. This is accomplished by nesting *<OrderedContest>* under *<OrderedHeader>*.

A code example follows, showing several headers defined for the ballot style and for contests within the ballot style. The headers are then re-used in subsequent ballot style definitions.

```
1   <ElectionReport xmlns="NIST_V2_election_results_reporting.xsd"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="NIST_V2_election_results_reporting.xsd">
4       <Election>
5           <BallotStyle>
6               <GpUnitIds>ru-beaverton-p1</GpUnitIds>
7               <OrderedContent xsi:type="OrderedHeader">
8                   <HeaderId>he-official</HeaderId>
9               </OrderedContent>
10              <OrderedContent xsi:type="OrderedHeader">
11                  <HeaderId>he-nonpartisan</HeaderId>
12                  <OrderedContent xsi:type="OrderedContest">
13                      <ContestId>cc1</ContestId>
14                  </OrderedContent>
15                  <OrderedContent xsi:type="OrderedContest">
16                      <ContestId>cc2</ContestId>
17                  </OrderedContent>
18              </OrderedContent>
19              <OrderedContent xsi:type="OrderedContest">
20                  <ContestId>cc3</ContestId>
21              </OrderedContent>
22          </BallotStyle>
23          <BallotStyle>
24              <GpUnitIds>ru-bentley-p1</GpUnitIds>
25              <OrderedContent xsi:type="OrderedHeader">
26                  <HeaderId>he-official</HeaderId>
27              </OrderedContent>
28              <OrderedContent xsi:type="OrderedContest">
29                  <ContestId>cc3</ContestId>
30              </OrderedContent>
31              <OrderedContent xsi:type="OrderedContest">
32                  <ContestId>cc4</ContestId>
33              </OrderedContent>
34          </BallotStyle>
35          <Contest ObjectId="cc1" xsi:type="CandidateContest">
36              <BallotTitle>
37                  <Text Language="en">Council Member</Text>
38              </BallotTitle>
39              <ElectionDistrictId>ru-beaverton</ElectionDistrictId>
40              <Name>City Council Member</Name>
41              <VotesAllowed>3</VotesAllowed>
42          </Contest>
43          <Contest ObjectId="cc2" xsi:type="CandidateContest">
44              <BallotTitle>
45                  <Text Language="en">Mayor</Text>
46              </BallotTitle>
47              <ElectionDistrictId>ru-beaverton</ElectionDistrictId>
48              <Name>Mayor</Name>
49              <VotesAllowed>1</VotesAllowed>
50          </Contest>
51          <Contest ObjectId="cc3" xsi:type="BallotMeasureContest">
52              <BallotTitle>
53                  <Text Language="en">GALDWIN COUNTY ANIMAL SHELTER</Text>
54              </BallotTitle>
55              <ElectionDistrictId>ru-beaverton</ElectionDistrictId>
56              <Name>Animal Shelter</Name>
57              <SummaryText>
```

```
1              <Text Language="en">Shall the tax limiation...</Text>
2          </SummaryText>
3      </Contest>
4      <Contest ObjectId="cc4" xsi:type="BallotMeasureContest">
5          <BallotTitle>
6              <Text Language="en">FILE MILAGE PROPOSAL</Text>
7          </BallotTitle>
8          <ElectionDistrictId>ru-bentley</ElectionDistrictId>
9          <Name>File Milage Proposal</Name>
10         <SummaryText>
11             <Text Language="en">Shall the tax limiation...</Text>
12         </SummaryText>
13     </Contest>
14     <ElectionScopeId>ru-michigan</ElectionScopeId>
15     <Name>
16         <Text Language="en">General Election</Text>
17     </Name>
18     <StartDate>2018-01-29</StartDate>
19     <EndDate>2018-01-29</EndDate>
20     <Type>general</Type>
21  </Election>
22  <Format>precinct-level</Format>
23  <GeneratedDate>2018-01-29T00:00:00Z</GeneratedDate>
24  <GpUnit xsi:type="ReportingUnit" ObjectId="ru-michigan">
25     <ComposingGpUnitIds>ru-beaverton</ComposingGpUnitIds>
26     <Name>
27         <Text Language="en">Michigan</Text>
28     </Name>
29     <Type>state</Type>
30  </GpUnit>
31  <GpUnit xsi:type="ReportingUnit" ObjectId="ru-beaverton">
32     <ComposingGpUnitIds>ru-beaverton-p1</ComposingGpUnitIds>
33     <Name>
34         <Text Language="en">Beaverton</Text>
35     </Name>
36     <Type>city</Type>
37  </GpUnit>
38  <GpUnit xsi:type="ReportingUnit" ObjectId="ru-beaverton-p1">
39     <Name>
40         <Text Language="en">City of Beaverton, Precinct 1</Text>
41     </Name>
42     <Type>precinct</Type>
43  </GpUnit>
44  <GpUnit xsi:type="ReportingUnit" ObjectId="ru-bentley">
45     <ComposingGpUnitIds>ru-bentley-p1</ComposingGpUnitIds>
46     <Name>
47         <Text Language="en">Bentley</Text>
48     </Name>
49     <Type>township</Type>
50  </GpUnit>
51  <GpUnit xsi:type="ReportingUnit" ObjectId="ru-bentley-p1">
52     <Name>
53         <Text Language="en">Bentley Township, Precinct 1</Text>
54     </Name>
55     <Type>precinct</Type>
56  </GpUnit>
57  <Header ObjectId="he-official">
```

```
        <Name>
            <Text Language="en">OFFICIAL BALLOT</Text>
        </Name>
    </Header>
    <Header ObjectId="he-nonpartisan">
        <Name>
            <Text Language="en">NONPARTISAN SECTION</Text>
        </Name>
    </Header>
    <Header ObjectId="he-city">
        <Name>
            <Text Language="en">CITY</Text>
        </Name>
    </Header>
    <Header ObjectId="he-proposal">
        <Name>
            <Text Language="en">PROPOSAL SECTION</Text>
        </Name>
    </Header>
    <Header ObjectId="he-township">
        <Name>
            <Text Language="en">TOWNSHIP</Text>
        </Name>
    </Header>
    <Header ObjectId="he-county">
        <Name>
            <Text Language="en">COUNTY</Text>
        </Name>
    </Header>
    <Issuer>State of Michigan</Issuer>
    <IssuerAbbreviation>MI</IssuerAbbreviation>
    <SequenceStart>0</SequenceStart>
    <SequenceEnd>0</SequenceEnd>
    <Status>pre-election</Status>
    <VendorApplicationId>MI POC</VendorApplicationId>
</ElectionReport>
```

Appendix A—Acronyms

Selected acronyms and abbreviations used in this document are defined below.

BMD	Ballot Marking Device
CDF	Common Data Format
DRE	Direct Record Electronic
EAVS	EAC Election Administration and Voting Survey
EMS	Election Management System
FIPS	Federal Information Processing Standard
FWAB	Federal Write-in Absentee Ballot
JSON	JavaScript Object Notation
OCD-ID	Open Civic Data Identifiers
UML	Unified Modeling Language
UOCAVA	Uniform and Overseas Citizens Assistance in Voting Act
VIP	Voting Information Project
VVSG	Voluntary Voting Systems Guidelines
XML	eXtensible Markup Language

Appendix B—Glossary

Selected terms used throughout this document are defined below. In some of the definitions, there is ancillary information that is not part of the definition but helpful in understanding the definition; this ancillary information is preceded with "*Note:*". Synonyms are preceded with "*Syn:*".

Absentee voting:	Ballot resulting from voting that can occur unsupervised at a location chosen by the voter either before or on election day.
Affiliation:	Association with a political party. *See also:* endorsement.
Ballot measure:	A question that appears on a ballot with options, usually in the form of an approval or rejection.
Ballot rotation:	The process of varying the order of listed candidates within a contest. This allows each candidate to appear first on the list of candidates an approximately equal number of times across different ballot styles or jurisdictions.
Ballot style:	Ballot data that has been put into contest order for a particular precinct and considers a particular set of voter situations. Voter situations include party affiliation (for closed primaries), and age of the voter (in states that permit 17-year-olds to vote in primary elections), among others.
Borough:	Term to signify a subdivision generally smaller than cities in terms of both geographic area and population and administered through a borough code in states such as CT, NJ, PA, and other states. Used to mean a county in AK. Used in New York City to subdivide the city much as counties subdivide a state.
Closed primary:	Partisan primary election in which the voter receives a ballot containing only those party-specific contests pertaining to the political party with which the voter is affiliated, along with non-party-specific contests presented at the same election. Unaffiliated voters may be permitted to vote only on non-party-specific contests.
Combined precinct:	Two or more precincts assigned the same polling place. *Syn:* consolidated precinct.
Contest:	A single decision or set of associated decisions being put before the voters (for example, the option of candidates to fill a particular public office or the approval or disapproval of a constitutional amendment). This term encompasses other terms such as "race," "question," and "issue" that are sometimes used to refer to specific kinds of contests. It does not refer to the legal challenge of an

election outcome.

Cross-party endorsement: Endorsement of a single candidate or slate of candidates by more than one political party. The candidate or slate appears on the ballot representing each endorsing political party.

Direct record electronic
(DRE):   A voting device that allows:

- electronic presentation of a ballot,
- electronic selection of valid voter choices, and
- electronic storage of these selections as individual records. It also provides a summary of these vote selections.

Early voting: Voting that occurs prior to election day under the supervision of election workers.

Election certification: The act of confirming the final official results of a jurisdiction's election by the election authority. This event occurs after results from valid ballots are tallied from all sources (election day, absentee, early vote, provisional, etc.) and results are validated and approved by those legally responsible.

Election day: The last day on which voters may cast a ballot. Absentee votes and early votes may be cast in advance of election day.

Election night: The period of time starting when the polls close in a jurisdiction in which results may first begin to be reported and ending when all precincts have reported.

Election management
system (EMS):   Set of processing functions and databases within a voting system typically used to:

- develop and maintain election definition data,
- perform ballot layout functions,
- create ballot presentation templates for ballot printers or devices used by voters for ballot markup,
- count votes,
- consolidate and report results, and
- maintain audit trails.

Election official: Any person who is involved with administering or conducting an election, including government personnel and election day workers. This includes any county clerk and recorder, election judge, member of a canvassing board, central election official, election day worker, member of a board of county commissioners, member or secretary of a board of directors authorized to conduct public elections, representative of a governing body, or other person engaged in the

performance of election duties as required by the election code.

Election District:	Administrative area in which voters are entitled to vote in contests that are specific to that area.
Endorsement:	Approval by a political party, for example, as the candidate that the party fields in a particular contest or as the candidate that should receive straight party votes. *See also:* affiliation. In some states, more than one party may endorse a candidate or contest option.
General election:	Regularly scheduled election in which all eligible voters, regardless of party affiliation, are permitted to select candidates to fill public office and vote on ballot measures.
In-person voting:	Voting that occurs at a polling place under the supervision of poll workers on election day. *See also:* early voting. *Syn:* polling place voting.
Jurisdiction:	Term as used in election contexts to signify a geographical area to which a practical authority has been granted to administer elections for political offices. Areas of jurisdiction apply to local, state, and federal levels. *Note:* States, counties, cities, towns and townships are all examples of jurisdictions.
Municipality:	Term as used in election contexts to signify a jurisdiction such as city or town or village that has some form of local government for which elections are generally conducted.
N-of-M:	Voting variation in which the voter is entitled to allocate a fixed number of votes (N) over a list of M contest choices or write-ins, with the constraint that at most 1 vote may be allocated to a given contest choice. *Note:* This usually occurs when multiple seats are concurrently being filled in a governing body such as a city council or school board where candidates run at-large. The voter is not obliged to allocate all N votes. 1-of-M is N-of-M voting where N = 1.
Nonpartisan primary:	Election held to narrow the field of candidates in non-party-specific contests prior to the general election. A primary election may be comprised of a nonpartisan primary for some contests or measures and a partisan and/or open primary for others.
Open primary:	Partisan primary election in which the voter may choose a political party at the time of voting and vote in party-specific contests associated with that party, along with non-party-specific contests presented at the same election. Some states require voters to publicly declare their choice of party at the polling place, after which the election worker provides or activates the appropriate ballot. Other

states allow the voters to make their choice of party within the privacy of the voting booth.

Overvote:

Occurs when the number of selections made by a voter in a contest is more than the maximum number allowed. The number of allowed selections is equal to the number of votes lost. For example, if three selections are made in a vote-for-two contest, the number of votes lost is two.

Partisan primary:

Primary election held to narrow the field of candidates in party-specific contests.

Polling place:

Location at which voters may cast in-person ballots under the supervision of election workers during one or more specific time periods.

Post-election canvass:

A review of all election results and re-tabulation, resulting in the certification of the results. Generally, the canvass is conducted according to established laws and time frames.

Precinct:

Election administration division corresponding to a geographic area that is the basis for determining which contests and issues the voters legally residing in that area are eligible to vote on.

Primary election:

Election held to determine which candidates will represent a political party for particular offices in the general election or to narrow the field of candidates in non-party-specific contests prior to the general election.

Provisional ballot:

A failsafe ballot provided to a voter whose eligibility for a regular ballot cannot be immediately determined. The ballot may be counted or further processed depending on state law. See also: 52 U.S.C. Section 21082, recallable ballot.

Reporting unit:

Geographical area in which reported totals or counts are reported (for example, a jurisdiction, precinct, or election district).

Schema:

A file containing definitions of data elements and attributes with rules for usage, e.g., for XML.

Split precinct:

Precinct serving voters from two or more administrative divisions, such as election districts, that may require different ballot styles, other than for primaries. *Syn:* split.

Straight party voting:

Mechanism that allows voters to cast a single vote to select all candidates on the ballot from a single political party.

Tabulator: A device that counts votes.

Town: An urban area that has a name, defined boundaries, and local government, and that is generally larger than a village and smaller than a city. Term used in New England, New York, and Wisconsin to refer to the equivalent of the civil township in these states.

Township: A widely used unit of local government in the United States, subordinate to a county, with some form of local government for which it generally conducts elections.

UOCAVA voter: An overseas voter or an active duty member of the U.S. military, either within or outside the United States, including any accompanying spouse and family members who are eligible to vote in their last place of residence in the United States. The Uniformed and Overseas Citizens Absentee Voting Act is commonly referred to as UOCAVA.

Undervote: Occurs when the number of voter selections in a contest is less than the maximum number allowed for that contest or when no selection is made. The number of undervotes is equal to the number of votes lost, for example, if no selection is made in a vote for two contest the number of votes lost is two.

Vote-capture device: An automated device that is used directly by a voter to vote a ballot. *See also:* direct record electronic (DRE).

Vote center: A physical location where voters from multiple precincts may cast their ballots.

Voter turnout: The number of voters who have cast ballots as a percentage of the total number of voters who can cast ballots. *Note:* Various states calculate this differently, sometimes using the total number of potentially eligible voters whether registered or not.

Voting variation: A contest option or feature, e.g., approval voting, cumulative voting.

Write-in: A type of contest option that allows a voter to specify a candidate, usually not already listed under the same contest. Depending on jurisdiction rules, only certain names will be considered as valid write-ins.

Appendix C—References

[1]   W3C, *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, W3C Recommendation, November 26, 2008, http://www.w3.org/TR/xml/ [accessed

2/1/2019].

[2]   Object Management Group (OMG), *UML Specification version 1.1* (OMG document ad/97-08-11) September 22, 2011, http://omg.org/ [accessed 2/1/2019].

[3]   JavaScript Object Notation, http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf [accessed 2/1/2019].

[4]   Election Assistance Commission, *Election Administration and Voting Survey* [Web site]*,* https://www.eac.gov/research-and-data/studies-and-reports [accessed 2/28/2019].

[5]   Florida Department of State, Division of Election, *Florida Election Results Summary XML Schema Reference*, Version 3.0-S, July 2015, http://dos.myflorida.com/media/695297/ds-de-137-summary.pdf [accessed 2/1/2019].

[6]   *Voting Information Project (VIP)* [Web site], http://votinginfoproject.github.io/vip-specification/ [accessed 2/1/2019].

[7]   OASIS, Election Markup Language (EML) Specification Version 7.0, Committee Specification 01, October 27, 2011. Available at: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=election [accessed 2/1/2019].

[8]   City of Cambridge, Massachusetts, *Geographic Information Systems: Map Gallery* [Web site], https://www.cambridgema.gov/GIS/mapgallery [accessed 2/1/2019].

[9]   Hilkert, Scott, "Application of Lessons Learned in the Illinois EDC Project Toward Emerging Election Data Standards and System Guidelines," submitted to *Workshop for a Common Data Format for Electronic Voting Systems*, National Institute of Standards and Technology, Gaithersburg, Maryland, October 29-30, 2009, http://grouper.ieee.org/groups/1622/WorkingDocuments/workshop-2009-10-NIST/hilkert-Lessons-Learned-in-Illinois-EDC-1-1.pdf [accessed 2/1/2019].

[10]   Open Civic Data, *OCD Identifiers* [Web page], http://opencivicdata.readthedocs.org/en/latest/ocdids.html [accessed 2/1/2019].

[11]   Wikipedia, *ISO 639* [Web page], https://en.wikipedia.org/wiki/ISO_639 [accessed 2/1/2019].

Appendix D—File Download Locations

The files associated with this specification are available for download from a NIST repository, whose address is:

https://github.com/usnistgov/ElectionResultsReporting

These files are also available from:

http://vote.nist.gov

These files are:

- This specification,
- XML schema,
- JSON schema,
- UML model.

Appendix E—Version 2.0 Changes from Version 1.0

The 2.0 version of the Election Results Reporting XML Schema includes the following important changes from the 1.0 XML Schema:

1. Removed various 'collection' elements:
   o `BallotSelection->VoteCountsCollection`
   o `Election->BallotStyleCollection`
   o `Election->CandidateCollection`
   o `Election->ContestCollection`
   o `ElectionReport->GpUnitCollection`
   o `ElectionReport->OfficeCollection`
   o `ElectionReport->PartyCollection`
   o `ElectionReport->PersonCollection`
   o All occurrences of `ExternalIdentifiers`
2. Removed `BallotStyle->ObjectId` attribute
3. Removed `ElectionReport->Signature` element
4. Removed `SummaryCounts` from `GpUnit`
5. Removed `OrderedContest` from `BallotStyle`
6. Capitalized all lower-case attributes:
   o All `label` attributes are now "`Label`"
   o All `objectId` attributes are now "`ObjectId`"
   o `AnnotatedString->annotation` attribute is now "`Annotation`"
   o `LanguageString->language` attribute is now "`Language`"
7. Changes to reporting of other counts besides vote counts:
   o Changed name of `SummaryCounts` to `BallotCounts`
   o Added `Election->BallotCounts`
   o Added `Contest->OtherCounts` for contest-related summary counts
      ▪ `Overvotes`, `Undervotes`, `WriteIns` moved from `BallotCounts`
      ▪ Changed `Overvotes`, `Undervotes` to be `float` types
   o `Counts->Type` attribute is now required (applies to `SummaryCounts` and `VoteCounts` types)
8. Adjusted the multiplicity of various attributes and associations:
   o Adjusted multiplicity of `ContactInformation->Directions` from 0..* to 0..1
   o All attributes whose `Type` is `AnnotatedUri` now have multiplicity of 0..*
9. Added new types:
   o Added `AnnotatedUri` type
   o Added `Header` type
   o Added `OrderedContent` type
   o Added `OrderedHeader` type
   o Added `DateTimeWithZone` type to enforce time zone requirement
10. Added various new associations and attributes:
   o Added `BallotStyle-OrderedContest`
   o Added `CandidateContest->NumberRunoff`
   o Added `Counts->Round`
   o Added `Candidate->CampaignSlogan`
   o Added `Candidate->ContactInformation`

132

- o Added `ElectionReport->Header`
- o Added `Office->Description`
- o Added `Party->IsRecognizedParty`
- o Added `Party->ContactInformation`
- o Added `Party->PartyScopeGpUnit`
- o Added `Party->LeaderPersonIds`
- o Added `Party->Slogan`
- o Added `Person->ExternalIdentifier`

11. Modified enumeration literals:
- o Added `recall` enumeration literal to `BallotMeasureType`
- o Added `defeated` enumeration literal to `CandidatePostElectionStatus`
- o Removed `write-in` enumeration literal from `CandidatePreElectionStatus`
- o Added `seats` enumeration literal to `CountItemType`
- o Added `bmd`, `dre` enumeration literals to `DeviceType`
- o Removed `electronic`, `level`, `mixed-systems`, and `punchcard` from `DeviceType`
- o Added `country` enumeration literal to `ReportingUnitType`
- o Removed `1-of-m` enumeration literal from `VoteVariation`

12. Modified attribute types:
- o Changed `ContactInformation->Uri` to be of type `AnnotatedUri`
- o Changed `GpUnit->Name` to be of type `InternationalizedText`
- o Changed `OtherCounts->overvotes, undervotes` to float
- o Changed `Party->abbreviation` to be of type `InternationalizedText`
- o Changed `LatLng->Latitude` and `Longitude` from `float` to double

13. Changed `ElectionReport->GeneratedDate` to be of type `DateTimeWithZone`
14. maxLength of `ShortString` is now 32 (up from 16)
15. Converted all usages of `anyURI` to `AnnotatedUri`
16. Renamed the following:
- o Renamed `BallotSelection` to `ContestSelection`
- o Renamed `Device` to `DeviceClass`
- o Renamed `ReportingDevice->Device` to `DeviceClass`
- o Removed pluralization of `ExternalIdentifier`

Appendix F—Change Log

<u>Revision 2 Release 2 – March 31, 2020</u>

- In the UML documentation, corrected the order of UML class attributes and enumeration values in the tables so that they match the order in the UML model and in the JSON and XML schemas.
- In the UML documentation, mentions of class and attribute names are properly linked to each other. Also done for mentions of enumeration and value names.
- Adjusted UML model to use UML String instead of xs:string.
- Added Candidate::CampaignSlogan - The slogan or motto used by the candidate in their campaign.
- Added Party::Slogan - The slogan or motto used by a political party.
- Added ExternalIdentifier attribute to BallotMeasureSelection enumeration.
- Added directed association from Party to GpUnit with multiplicity 0..* with role name PartyScopeGpUnitIds - the GpUnit(s) the party operates in or the top-most GpUnit.
- Changed data type of Party::Abbreviation from String to InternationalizedText.
- Changed data type of GpUnit::Name from String to InternationalizedText.