Authors: H. Deutsch, G. Gilbert, D. Carey, C. Hughes, L. Lochridge (Author & Editor)

DRAFT NIST SP 1500-10X Voting Methods

Section 3.1.3 Process Models and UML Domain Models,

Sub-section 3.1.3.1 Process Models Overview

3.1.3.1 Process Models Overview

Overview

This section describes process flow models that were developed to represent the legislative and local rules plain language algorithmic definitions or specifications of tally, tabulation and voting methods. The process models introduced in this section include a generally applicable tabulation engine and voting method-specific process models that are described in detail in section 3.1.4. Each process model is presented as one or more process flow diagrams along with a corresponding process step description and explanation of variant use cases. Features that are common to all voting methods are presented in this section.

The process flow models are illustrative of the knowledge acquisition and analysis performed by surveying legislative and local rules texts. They encapsulate the functional knowledge acquired as a common process flow that represents a tally, tabulation or voting method variant (TTVV) based on research and analysis of legislative and local jurisdiction rules as specification in plain language of how elections are operated in practice. Therefore, the models in this specification are descriptive rather than prescriptive. The primary benefits of developing the process flows illustrated in this section and the following section are to act as an abstraction or encoding of the functional features of a TTVV and are:

- A practice that improves fidelity of the translation of the legislative plain language algorithmic definitions of TTVVs into mathematical models and mechanized mathematical constraint logic, and

- A broadly understandable illustrative documentation for potential adopters and a wide spectrum of stakeholders to understand the functionality and use cases that may apply in the voting methods universe UML model, and also as represented in the corresponding mathematical constraint logic models.

A generally applicable, generic, tabulation process flow that is independent of the voting method is described in this section. The features of the generic tabulation process flow presented in this section apply all of the voting method-specific process flow in the following section, 3.1.4.

Terminology

See NIST/TGDC Voting Glossary https://pages.nist.gov/ElectionResultsReporting/#b-glossary] or the Election Assistance Commission (EAC) Voluntary Voting Systems Guidelines (VVSG) 2.0 glossary section.

Tabulation Scenarios

The following tabulation scenarios are supported:

- Batch
- Batch with Accumulation
- Contest
- Reporting Unit
- Aggregation
- Compositions of Batch and Contest variants with Accumulation and Aggregation
- Multi-Contest Multi Voting Method per Ballot
- Distributed variants of this list of tabulation scenarios
- Precinct or Central tabulation

Nothing about the process flow diagrams presented in this specification are intended to depict details or system-level implementations. The process flows are generalizations or abstractions of process flow and steps that represent tabulation and voting methods process at a high level, or as a logical abstraction.

- Process flows are intended to work for the tabulation scenarios listed above, and for use cases including tabulation such as tabulation that is part of an audit process or recount process.

- The voting method and process flow models in this specification support variations in how artifacts, concepts and sub-processes of TTVVs are implemented in systems in practice. For example, the specific criteria for declaring a ballot to be a “blank ballot , may differ according to a jurisdiction's specific implementation rules or local legislation. Those 'business rules' or definitions that define what the criteria or rules are that define a blank ballot designation are intended to be defined as local jurisdiction configuration. The context in which the configuration applies might be voting method specific or not voting-method specific for the jurisdiction.

- Contest/Ballot designation in a process flow model is intended to be interpreted as a concise label that represents more detailed variants that include either contest focused or ballot focused tabulation scenarios and processing vote selection data in batches, sequences, sub-sequences, partitions of aggregations.

- Logging, error handling and other aspects of an implementation or of a more detailed complex process flow are not shown in this specification’s process flow diagrams because the purpose of this diagram is to show only the major elements of the flow specific to tabulation or counting and to produce an outcome or output, rather than depict a systems detialed implmentation.

- [**if we are combining the listed scenarios in bullet one, it could be that UNK or N/A would be one of the possible answers to Tie Detected, but we can note that or perhaps change the diagram to make reaching that decision point only occur if it makes sense to know for the tab. scenario. E.g., for a batch without accumulation, the tabulator engine would not be able to know if there was a tie. -LL]

- Contest Tie Detected is only applicable to the Contest Tabulation Scenario and only in a variant of tabulation scenario in which the tabulation reaches a point in the process and reaches a state of tabulation, where there is an accumulated tabulation known to the tabulator at the time, and/or sequence point, sufficient to determine a contest tabulation outcome.

- For Ranked Choice Voting or any multi-round voting method tabulation scenario in which is it required that tabulation store the state per round or to determine outcome per round it is an implicit requirement that ties are detected per round. In that case, Contest Tie Detected represents an attribute of the Contest state for the particular round that is being executed at the time or sequence point of the tabulation run, and not the final or complete contest outcome.

- For aggregation, distributed, or incremental tabulation scenarios, Contest Tie Detected represents an attribute of the Contest state for the particular [subsequence/subset/node (distributed) /batch/sequence, reporting, test deck, as the case may be -LL] that is being executed at the time or sequence point of the physical or logical tabulation run, and not the final or complete contest outcome.

- Tabulation or a voting method produces an 'outcome', but never declares a "winner". Only an Elections Official has the authority to determine who is a winner or whether a ballot measure has passed.

- Execute Tie-Breaking: If that step occurs in-line in a specific tabulation scenario implementation, then it is most often specified in applicable legislation or rules as a manual, human actor executed sub-process. However, there are jurisdictions for which the legislative rules state that in advance (of the election voting or in advance of tabulation occuring) the Election Official shall select a method of tie breaking, without restrictions on the method selected. There are jurisdictions which do not exclude an electronic or digital tie-breaking method. Lastly, there are jurisdictions for which there is no explicit exclusion of tie breaking outcome having been set in advance, irrespective of tie-breaking method in use.

  This generic tabulation process flow supports manual tie breaking as an alert of a tabulation contest tie (including for RCV, for which a tie within a round is detected and is necessary to a tiebreak to determine transfer).

  - If the jurisdiction uses digital/electronic tiebreaking then execution of that might be automated, and may or may not involve the use of a predetermined tiebreaking outcome or rule. By pre-determined we mean that in advance of a tie event occuring, a previously executed tie-breaking outcome, such as a digital representation of a coin toss might be provided.
  - This reference model does not represent the details of electronic tiebreaking, nor human executed tiebreaking.

-LL]

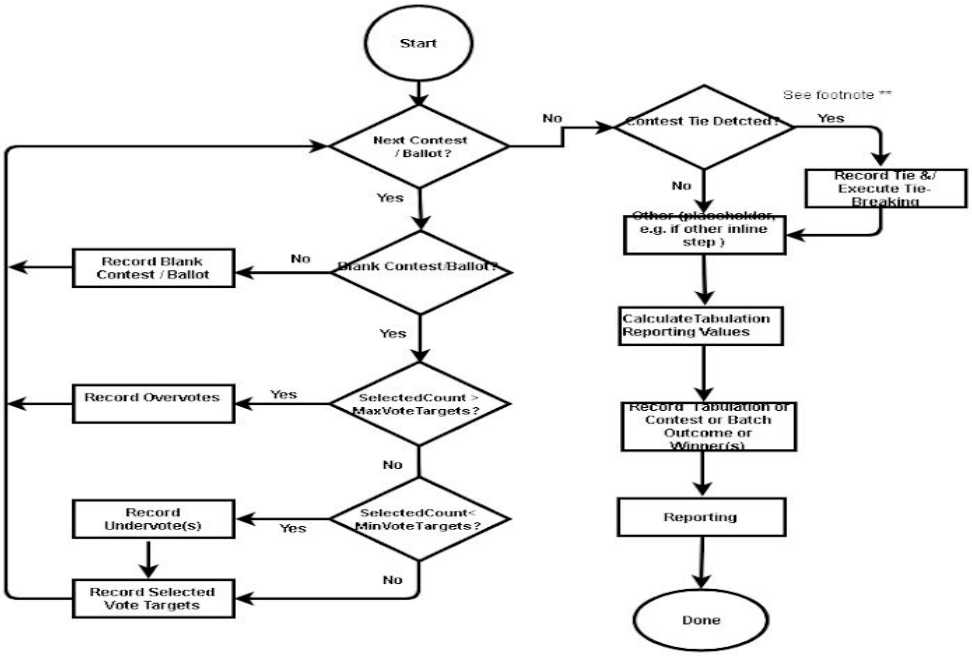
Authors: H. Deutsch, G. Gilbert, D. Carey, C. Hughes, L. Lochridge (Author & Editor)

DRAFT NIST SP 1500-10X Voting Methods

Section 3.1.3 Process Models and UML Domain Models,

Sub-section 3.1.3.1 Process Models Overview

NIST 1500-10x Voting Methods 3.1.3 Generic Ballot Tabulation Engine Process Flow



DRAFT 20171127 -LLochridge, Based on "Regular Tabulation Rules -HD 11/3/2017

Need to also doc case where outcome is winners? (cant know if batches, aggregates, etc.) H: but can be distributed even for RCV (not exist in current impl. tho)

Text

Footnotes

- Tabulation engine is intended to work for batch, batch with accumulation, aggregation, and for ballot focused or contest focused tabulation scenarios
- For cases such as blank ballot having different potential meanings (e.g. the example Ryan Macias gave per voting methods, or for a jurisdiction's specific implementation rules, those 'business rules' or definitions would be defined in the configuration.
- Contest/Ballot should be interpreted as a concise label that also may refer to batches, sequences, sub-sequences, partitions of aggregations.
- Logging, error handling and other aspects of an implementation or of a more detailed complex process flow are not shown because the purpose of this diagram is to show only the major elements of the flow specific to tabulation or counting and to produce an outcome or output.
- **if we are combining the listed scenarios in bullet one, it could be that UNK or N/A would be one of the possible answers to Tie Detected, but we can note that or perhaps change the diagram to make reaching that decision point only occur if it makes sense to know for the tab. scenario. E.g., for a batch without accumulation, the tabulator engine would not be able to know if there was a tie.
- Contest Tie Detected is only applicable to the Contest Tabulation Scenario and only in a variant of that scenario in which the tabulation is at a point in the process where the is an accumulated tabulation known to the tabulator at the time and/or sequence point.
- Execute Tie-Breaking: If that step occurs in-line to a tabulation, it is most often specified in applicable legislation or rules as a manual, human executed, sub-process. The Tabulation process flow supports that as an alert of a tabulation contest tie (including if RCV, a tie within a round is detected and is necessary to a tiebreak to determine transfer). If the jurisdiction uses digital/electronic tiebreaking then execution of that might be automated, and may or may not involve the use of a predetermined tiebreaking outcome or rule.