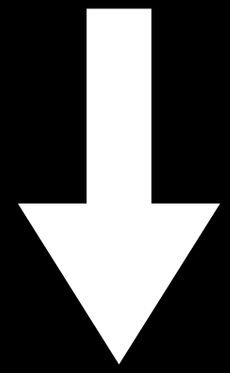
Annotation Proposal, Classifier

Gen2/3 Proposal

Neekon Saadat | 16.10.2025

Background

- Objective: investigate the sufficiency of the current annotation tags for auto-grading case performance



- Outcome: develop an auto-grader for case performance to automate the optimization of the sensing algorithm

Background

Existing Annotation Format

For info, see: `classifier / annotation_scheme_proposal.md`

- Tags mix flat and deep-nested structure without consistent pattern
  - 1.a.x/4/5/6/7/9 device state, 1.b/c.x/2/3/8 event/tip state → inconsistent role at each level
- Roman numerals require complex parsing
- Comments, device state, and tip events appear together, and tags combine state information with transition information; redundant
- **Need:** semantically consistent encoding of device/tip state
- **Outcome:** simple parsing, readability, non-breaking additions, Gen2/3 Consistency

Key			
1. Aspiration Event (Start: Stable baseline signals just before Aspiration - End: System returned to baseline signals)			
	a. Aspiration Location (Single Time Point Event / No End Time)		
		i. Distal RPA	
		ii. Distal LPA	
		iii. Proximal RPA	
		iv. Proximal LPA	
		v. Main PA	
		vi. Heart	
		vii. IVC	
		viii. Iliac	
		ix. Femoral	
		x. Popliteal	
		xi. In Sheath	
	b. Asp Type		
		i. Blood Only	
		ii. Blood to Wall Touch	
		iii. Blood to Wall Latch	
		iv. Blood to clog (if catheter was still clogged from previous aspiration)	
		v. Blood to clot to clog (Aspiration, the clot sensed, then clot clears tip and gets stuck in shaft with handle vacuum low)	
		vi. Blood to clot to blood (aspiration from a distance that clears tip and shaft)	
		vii. Blood to Clot to Wall	
		viii. Blood to Lollipop	
		ix. Clot to Blood (clot is sensed, then aspirated, then clears tip and shaft)	
		x. Clot to Clog (clot is sensed, then aspirated, clears tip, but stuck in shaft with handle vacuum held low)	
		xi. Clot to Wall	
		xii. Clot to Lollipop	
	c. Wall Contact (Start: Start of Imp Jump - End: Imp Returns to Baseline)		
2. Clot contact event (Start: Imp >1800 Ohms - End: Imp <1800 Ohms)			
3. Air			
4. Saline Prep			
5. Handle Injection			
	a. Loading Contrast (Start: Pressure Rise out of baseline signal - End: Returns to baseline signal)		
	b. Chasing with Saline (Start: Pressure Rise out of baseline signal - End: Returns to baseline signal)		
	c. Injection - Contrast At Tip (If Lingering contrast at tip)		
	d. Injection - Saline at Tip		
6. CMS Injection (If flushing through fitting at exit of syringe - Not sure if we need this or if we can sense this due to duckbill valve)			
7. Aspiration Tubing			
	a. Connected to Catheter (No End Time - Single time point event)		
	b. Disconnected from Catheter (No End Time - Single time point event)		
8. Tip In Sheath			
9. Moving in	a. Advancing		
	b. Retracting		
	c. Rotating		
	d. In Heart		
	e. In Main PA		
	f. In Prox LPA		
	g. In Prox RPA		
	h. In Distal LPA		
	i. In Distal RPA		
10. Other			

Background

Solution under Current Format Reveals Improved Format

Different annotation schemes (Gen2 vs Gen3) require separate parsing logic. Notes.md lines 214-228 discusses this problem:

```
|-----1.b.i------||---1.b.iii---||-------1.b.i-------| #Raw Data Gen3
|-----blood------||-----wall----||-------blood-------| #Range Data

vs

|-----blood------||---wall_latch_confirmed_by_contrast---||-------blood-------| #Raw Data Gen2
|-----blood------||-----------------wall-----------------||-------blood-------| #Range Data
```

There's no versioning mechanism in the annotation format itself. When we want to improve the annotation scheme, we have to write migration code that tries to guess which version a file is using based on pattern matching.

The scheme doesn't distinguish between fundamentally different types of information:

Physical reality - What material is actually at the catheter tip (blood, wall, clot, saline) **Detection goal** - What the device should ideally detect **Expected output** - What LED color should display given perfect detection **Timing** - State machine delays and minimum stay durations

```
"2. Clot Contact"                    (flat, simple)
"4 — Saline"                         (flat, simple)
"1.b.iii. Blood to Wall Latch"      (deeply nested)
"1.c"                                (moderately nested)
```

Format Specification

General Pattern: `<TISSUE>.<MODE>.<MODIFIER>[.<DETAIL>]`

- Atomic categories of information we already annotate, ranked by importance, in question format:
  - 1: What material is at the tip? **<TISSUE>**
    - Tells us: What should the device detect
  - 2: What mode is the device in? **<MODE>**
    - Tells us: What is the device doing?
  - 3: How is the tip interacting? **<MODIFIER>**
    - Tells us: What kind of TISSUE.MODE is this?
  - 4: Additional **<DETAIL>**s: information not included in existing vocabularies above, if used often then add it to the format

Solution under Current Format Reveals Improved Format

Proposal: Structured Dot Notation

A structured dot notation format will address all outstanding deficiencies in the current schema, maintaining machine readability and semantically rich structure while continuing to support ease of use by human annotators.

Format Specification

General Pattern: `<TISSUE>.<MODE>.<MODIFIER>[.<DETAIL>]` `tag` refers to the entire dot-notated sequence. `index` refers to one section of a tag, such as `<TISSUE>` or `<MODE>`. `label` refers to the option which occupies an index of a tag.

Category Definitions

The proposed format uses progressive specificity with the broadest classification on the left-hand side and the narrowest classification on the right-hand side. All-caps, hyphenated names are used for the components of a tag. Components for each index are from fixed vocabularies (extensible), and modifiers/details are optional. When an index is omitted, the dot delimiter must still be included to maintain positional clarity - for example, `BLOOD.TRACKING..` indicates tissue and mode with no modifier or detail, while `BLOOD.TRACKING..RPA` skips modifier but includes detail. `<TISSUE>` represents the vocabulary of tissue types which can appear at the device tip. A simple vocabulary may include `BLOOD`, `CLOT`, `WALL`, and this vocabulary may be extended as data begins supporting finer classifications such as `CHRONIC_CLOT`. `<MODE>` represents the device operating state, which may begin as a reduced vocabulary containing `TRACKING`, `CONTACT`, `ASPIRATING`, and may be extended to `PARTIAL-CONTACT`. `<MODIFIER>` represents a tissue-specific vocabulary which qualifies the tissue and device mode. For `BLOOD`, this may include `APPROACHING-CLOT`, `APPROACHING`, `LOW-IMPEDANCE`, `HIGH-IMPEDANCE`, and for any tissue with the mode `ASPIRATING`, `LATCH`, `ASP-VAC-INC`, `ASP-VAC-DEC`, `CLOT-SEEN`, and `CLOGGED`. Modifier will contain slowly evolving vocabularies for the different tissue/mode branches, and will allow annotators to signal patterns which aren't implicit for the tissue/mode pair or for which the modifier value may change while the tissue/mode remain contiguous. `<DETAIL>` represents a free vocabulary for annotating details not covered in the current version of the MODIFIER vocabulary, which are only useful for human readability, which are noted for discussion or review, or which are otherwise useful to annotate but not immediately relevant to the automated Classifier tool. Labels frequently used for `DETAIL` may be propagated to `MODIFIER` in the next version of the schema. For time ranges where multiple `MODIFIER` or `DETAIL` tags are applicable, multiple tags may be applied using the `_` underscore delimiter. There are edge cases which are impractical to statically check when combining tags, such as combining tags which do not make sense together, therefore new modifiers and details should be preferred when encountering recurring conditions which implicate multiple labels within a single index.

Example Annotations

The following examples demonstrate the proposed format applied to common scenarios observed in procedural data:

Basic tissue and mode combinations:

```
BLOOD.TRACKING.LOW-IMPEDANCE.                    # Catheter tracking through low-impedance blood
BLOOD.TRACKING.LOW-IMPEDANCE_APPROACHING-CLOT.   # Catheter tracking through low-impedance blood and approacl
WALL.CONTACT..                                   # Catheter in contact with wall
WALL.LATCH..                                     # Aspiration while in wall contact
CLOT.ASPIRATING..                                # Aspirating clot
AIR...PREPPING-CATHETER                          # Catheter in air (pre-insertion)
```

Using modifiers to add specificity:

```
BLOOD.TRACKING.APPROACHING-CLOT.      # Tracking while approaching clot
BLOOD.TRACKING.LOW-IMPEDANCE.         # Low impedance during blood tracking
WALL.ASPIRATING.LATCH.                # Wall latch during aspiration
CLOT.ASPIRATING.LATCH.                # Sustained clot engagement during aspiration
BLOOD.ASPIRATING.CLOGGED.             # Catheter lumen obstructed during aspiration
```

Using detail level for additional context:

```
BLOOD.TRACKING..RPA                   # Tracking in right pulmonary artery (no modifier)
WALL.CONTACT..LPA-DISTAL              # Wall contact in distal left PA (no modifier)
CLOT.ASPIRATING.LATCH.MAIN-PA         # Clot latch in main PA (with LATCH modifier)
CLOT.ASPIRATING..SMALL-CLOT           # Aspirating small clot (no modifier)
```

Temporal sequences showing state transitions:

```
BLOOD.TRACKING..                      # 100.0-150.0s
CLOT.ASPIRATING..                     # 150.0-155.0s  (transition from blood to clot)
BLOOD.ASPIRATING..                    # 155.0-160.0s  (clot cleared, back to blood)
```

Gen2/Gen3 Compatibility

Translation needed (presently)

Map Gen2 and Gen3 tags to common vocabulary:

```
|-----1.b.i------||---1.b.iii---||-------1.b.i-------| #Raw Data Gen3
|-----blood------||-----wall----||-------blood-------| #Range Data

vs

|-----blood------||---wall_latch_confirmed_by_contrast---||-------blood-------| #Raw Data Gen2
|-----blood------||-----------------wall-----------------||-------blood-------| #Range Data
```

Extract state transitions from state ranges

```
|-----blood------||---wall---||-------blood-------| #Range Data
----------------[b->w]------[w->b]------------------ #Point Data
```

Apply transition rules (defines what behavior we desire!)

```
[b->w]:
    delay: 1000ms
    light: blue
[b->c]:
    delay: 500ms
    light: orange
[c->w]:
    delay: 100ms
    light: blue
etc
```

```
---------------[b->w]------[w->b]------------------ #Point Data Condition
----------------------b---------g---------------- #Point Data Color
                   |~~~|                          #Delay for [b->w]
                               |~|                 #Delay for [w->b]
```

Generate expected state per sample

```
------------------------b---------g---------------- #Point Data Color
|-----------g----------||---b-----||-------g------| #Range Data Color
gggggggggggggggggggggggbbbbbbbbbbggggggggggggggggg #expected
```

This is convoluted!

Execute Grader, producing translation layer:

```
    1  timestamp,light_val_name,imp_mag,event,expected,actual_color
76308  219055,LIGHT_IMP_STATE_2_SALINE_BLOOD,927.0,,green,green
76309  219056,LIGHT_IMP_STATE_2_SALINE_BLOOD,927.0,,green,green
76310  219057,LIGHT_IMP_STATE_2_SALINE_BLOOD,927.0,,green,green
76311  219063,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76312  219064,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76313  219069,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76314  219070,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76315  219071,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76316  219077,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76317  219078,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76318  219084,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76319  219085,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76320  219086,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76321  219087,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76322  219094,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76323  219095,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76324  219096,LIGHT_IMP_STATE_2_SALINE_BLOOD,924.0,,green,green
76325  219102,LIGHT_IMP_STATE_2_SALINE_BLOOD,923.0,,green,green
76326  219103,LIGHT_IMP_STATE_2_SALINE_BLOOD,923.0,,green,green
76327  219110,LIGHT_IMP_STATE_2_SALINE_BLOOD,923.0,,green,green
```

And producing grades:

```
================================================================================
EPISODE: 50130-51748ms (1618ms)
Events: 2.0
Overall Grade: 0.999 (99.9%)
Grades by color:
  green     : 0.000 (0.0%)
  orange    : 1.000 (100.0%)
--------------------------------------------------------------------------------
actual:   ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
expected: ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
match:    ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||  (|=match, X=mismatch)
================================================================================
```

Gen2/Gen3 Compatibility

Direct Mapping Approach: Unified **Format,** Specified **Vocabulary,** Single **Grader**

```yaml
# Example schema file: inquis-gen3-a4f2b1c.yaml
tissue_vocabulary:
  - BLOOD
  - WALL
  - CLOT
  - SALINE
  - CONTRAST
  - AIR

mode_vocabulary:
  - TRACKING
  - ASPIRATING
  - LATCH
  - INJECTING
```

Every commit shall store a current pair of schema and config which apply to the current annotation needs and state machine. The name of the schema is included in all event files created in the annotation pipeline, which enables investigations involving old annotations corresponding to old standards - though, in principle, annotation vocabularies only grow, so old annotations are forward compatible but new annotations are not backwards compatible.

```
# Schema: inquis-gen3-a4f2b1c
# Config: inquis-gen3-d8e3a2f
Event,Start Time,End Time,Notes
BLOOD.TRACKING..,13.9,14.9,
WALL.ASPIRATING.LATCH.,659.2,660.8,
```

All Classifier results include a reference to the schema and configuration used, which enables fully reproducible results and comparison between versions of code.

```json
{
  "schema_version": "inquis_gen3-master-a4f2b1c",
  "config_version": "inquis_gen3-master-d8e3a2f",
  "classifier_version": "classifier-master-e6c35b",
  "grade": 0.87
}
```

For iterative configuration optimizations, the naming scheme of the configuration file is changed to `opt-<HASH>`, wherein the hash is computed on the entire file. This enables reversion to previous versions of the config and verifiable comparisons/roll-backs to individual iteration steps due to the 1-to-1 naming enforced by the hash.

```
configs/inquis-gen3-d8e3a2f.yaml     # Original config
configs/opt-7a3f9e2b1c4d5e6f.yaml    # Optimization iteration 1
configs/opt-2b4c8d9f3e1a5c7b.yaml    # Optimization iteration 2
```