# Simple Water Elevation Transit System

## Adding Water Level Adjustments

July 17, 2025

Michael M Lee

# Adding Water Level Adjustments

**Motivation**

- After transferring a vessel up or down a transit lane, another vessel can be transferred in the same direction in that transit lane. However, water levels need to be adjusted before doing a transfer in the opposite direction in that transit lane.

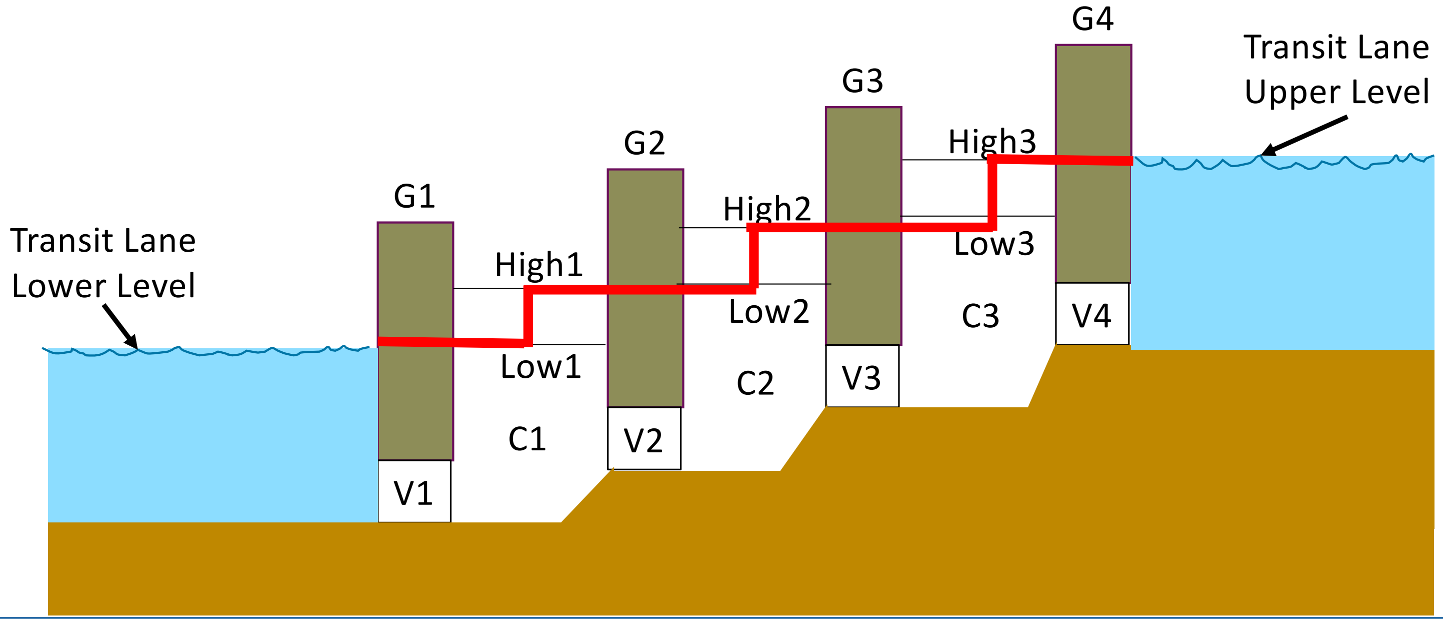
**Today's Considerations**

- Review initial requirements for water level adjustments
- Identify an approach to lower and raise a transit lane's water levels
- Look at the impact of that approach on existing models

# Reprise: Multi Lock Transfer

**Observation**

- For a Raise Transfer, the vessel enters each chamber at its Low water level and exits at its High water level
- For a Lower Transfer, the vessel enters each chamber at its High water level and exits it at its Low water level.

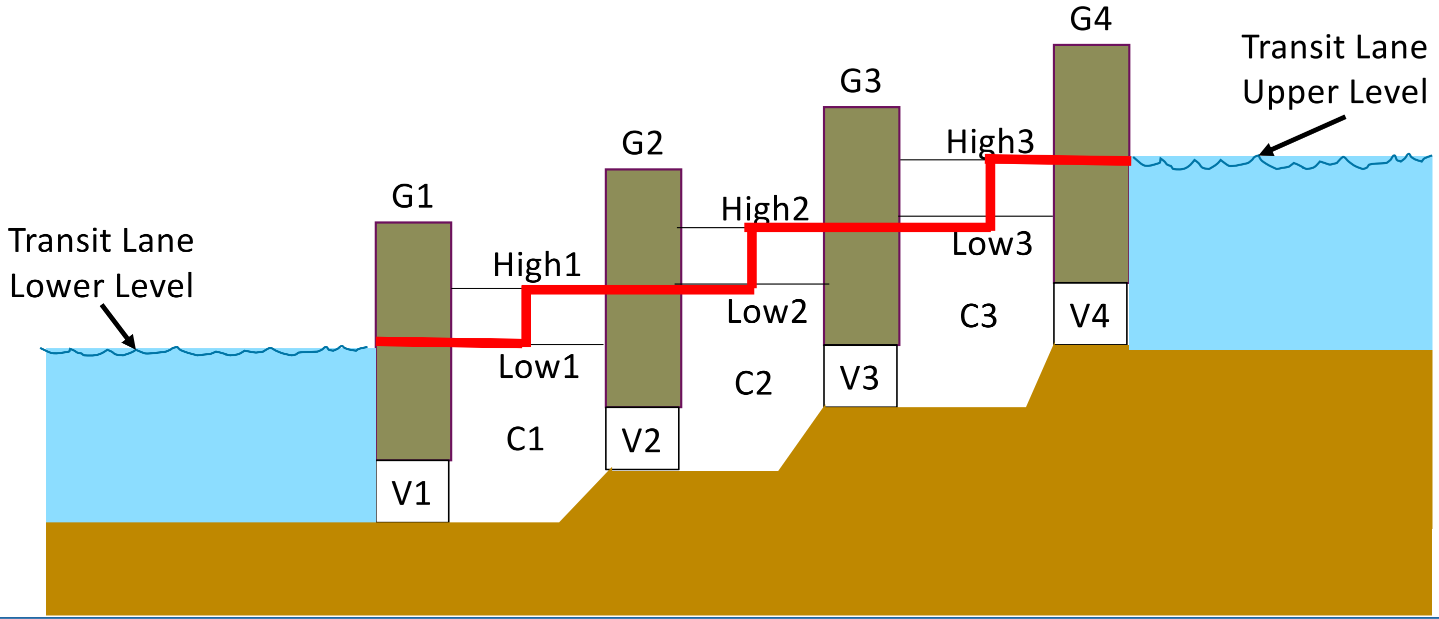


# Reprise: Assumptions for Water Management

**Assumptions**

- Volume C1=C2=C3
- The High minus Low volume is the same for all chambers
- Transit Lane Upper Level has unlimited water supply
- Transit Lane Lower Level has an unlimited water sink

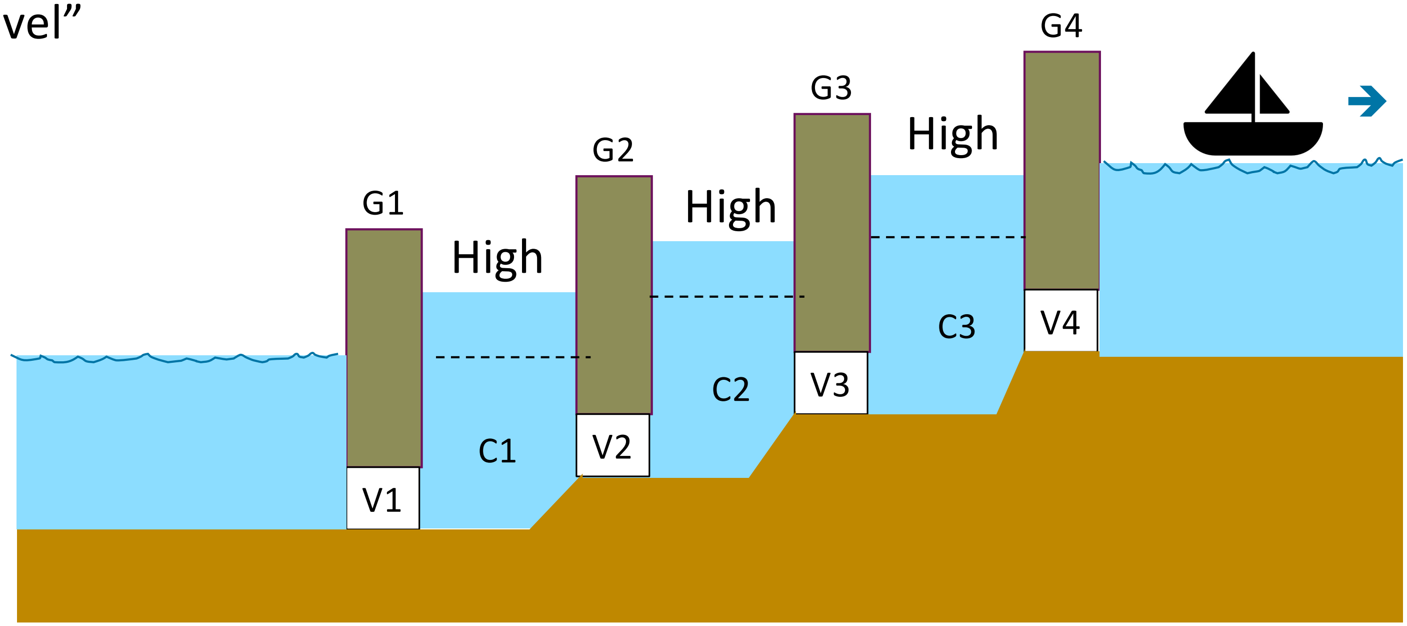
This is just one set of assumptions that meet the criteria for successfully managing water levels. Paul can elaborate other ones.



# Water Levels after a Raise Transfer

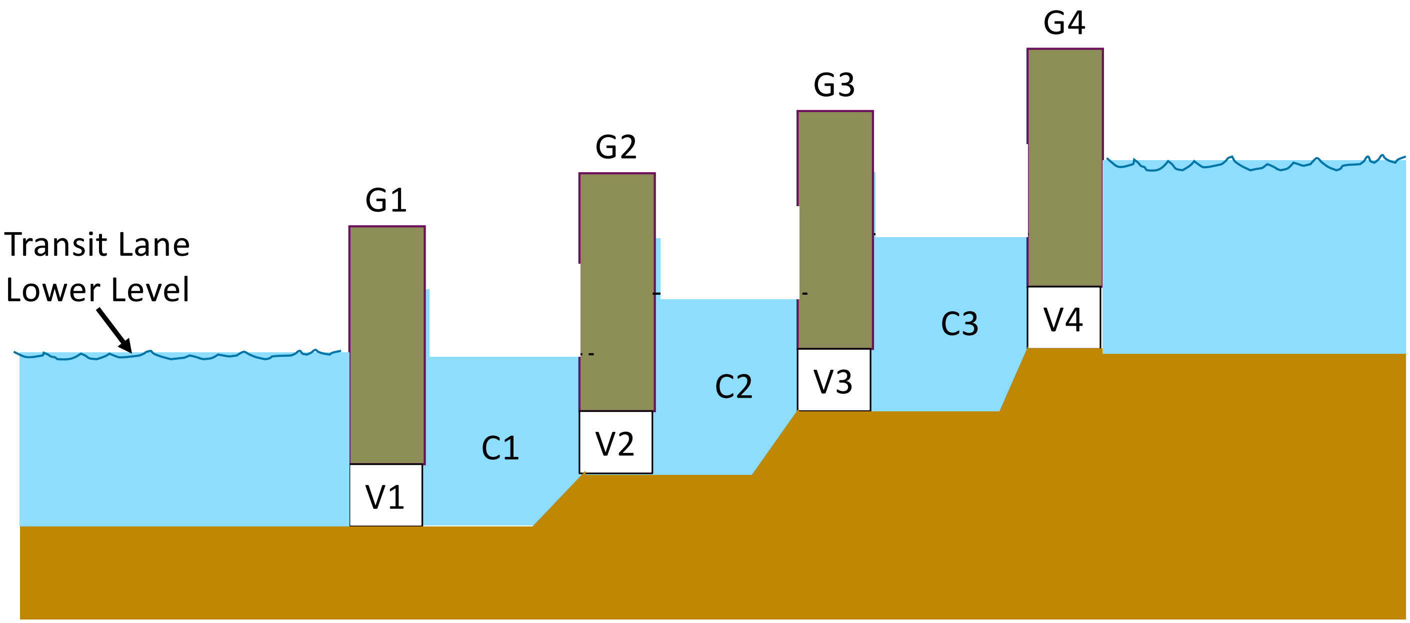
"For a Raise Transfer, the vessel enters each chamber at its Low water level and exits at its High water level"

Can raise another vessel in this transit lane, but must lower all the water levels to lower a vessel.

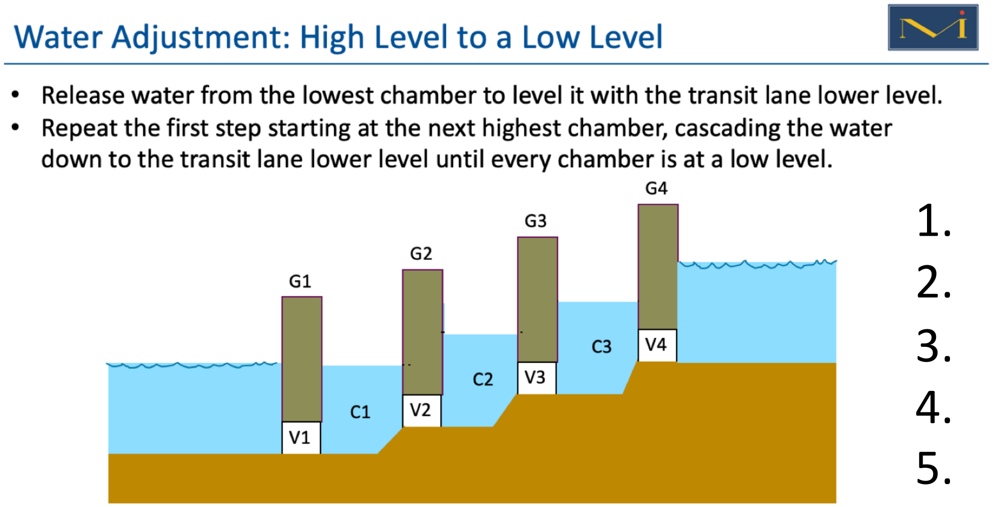


# Water Adjustment: High Level to a Low Level

- Release water from the lowest chamber to level it with the Transit Lane Lower Level.
- Repeat the first step starting at the next highest chamber, cascading the water down to the transit lane lower level until every chamber is at a low level.



# Operational Steps: High Level to a Low Level



## Water Adjustment: High Level to a Low Level

- Release water from the lowest chamber to level it with the transit lane lower level.
- Repeat the first step starting at the next highest chamber, cascading the water down to the transit lane lower level until every chamber is at a low level.

## Step → Details

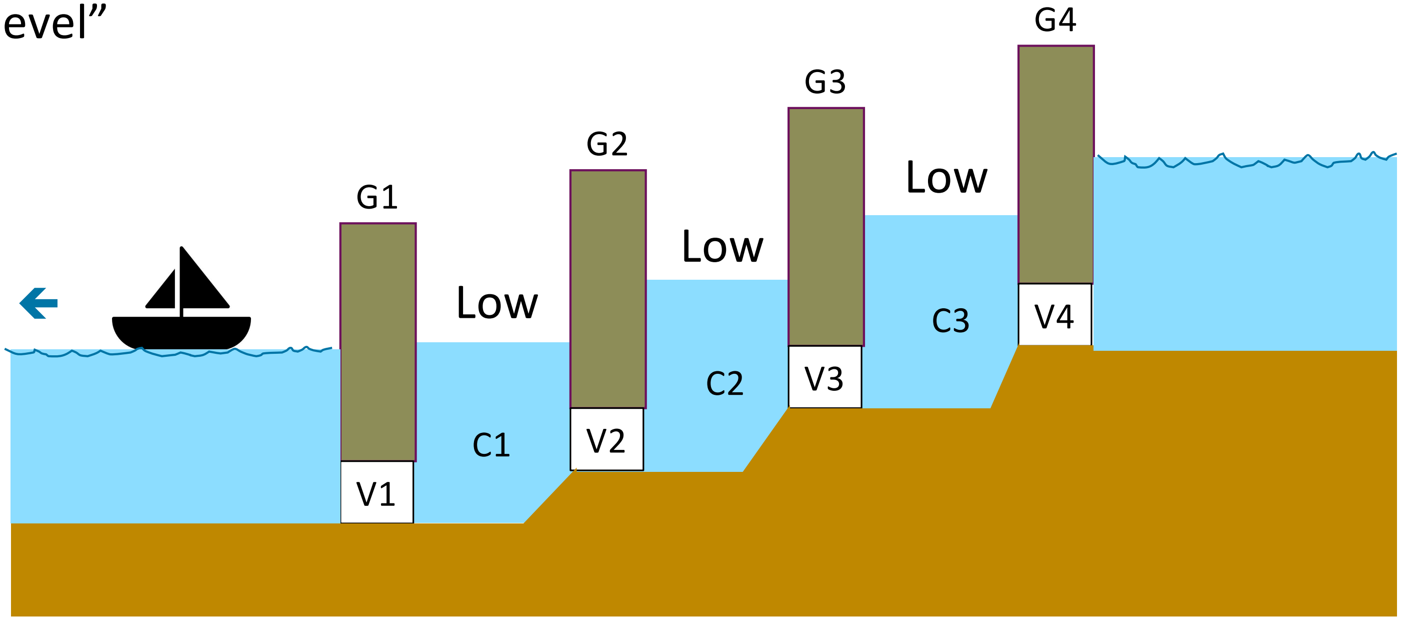
1. Lower C1 → Open V1, No Flow V1, Close V1
2. Lower C2 → Open V2, No Flow V2, Close V2
3. Lower C1 → Open V1, No Flow V1, Close V1
4. Lower C3 → Open V3, No Flow V3, Close V3
5. Lower C2 → Open V2, No Flow V2, Close V2
6. Lower C1 → Open V1, No Flow V1, Close V1

Clear Pattern

# Water Levels after a Lower Transfer

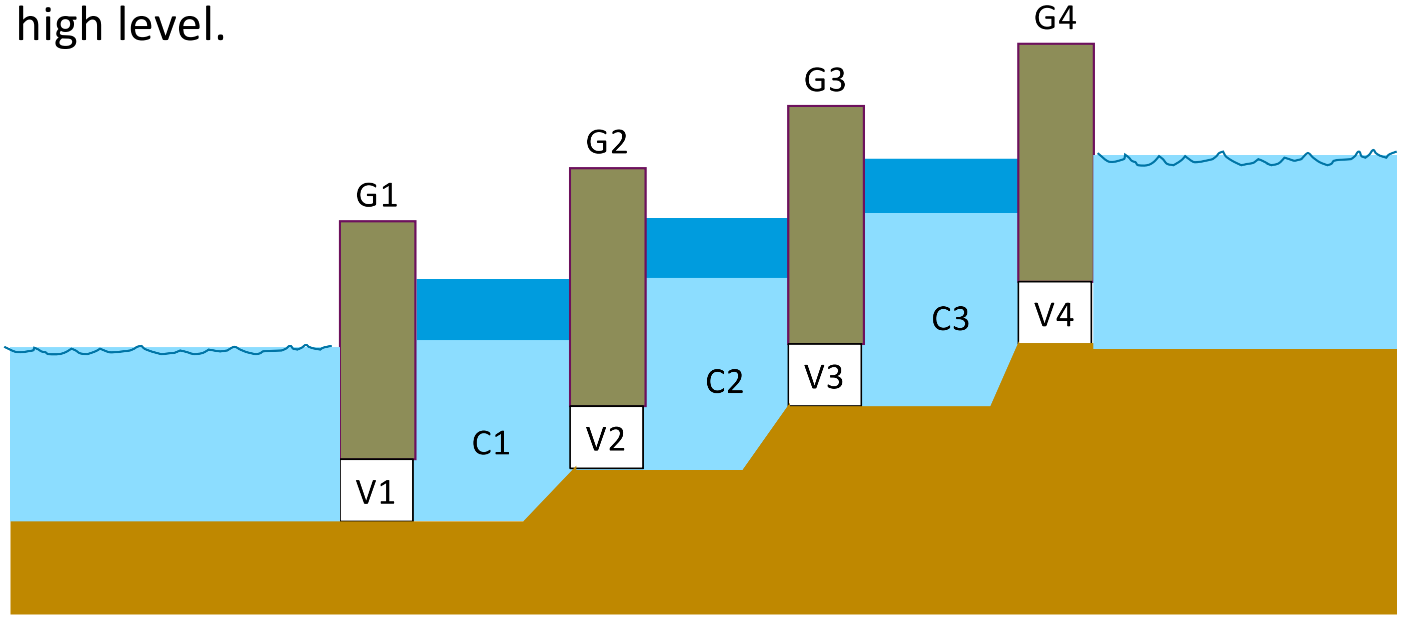
"For a Lower transfer, the vessel enters each chamber at its High water level and exits at its Low water level"

Can lower another vessel in this transit lane, but must raise all the water levels to raise a vessel.

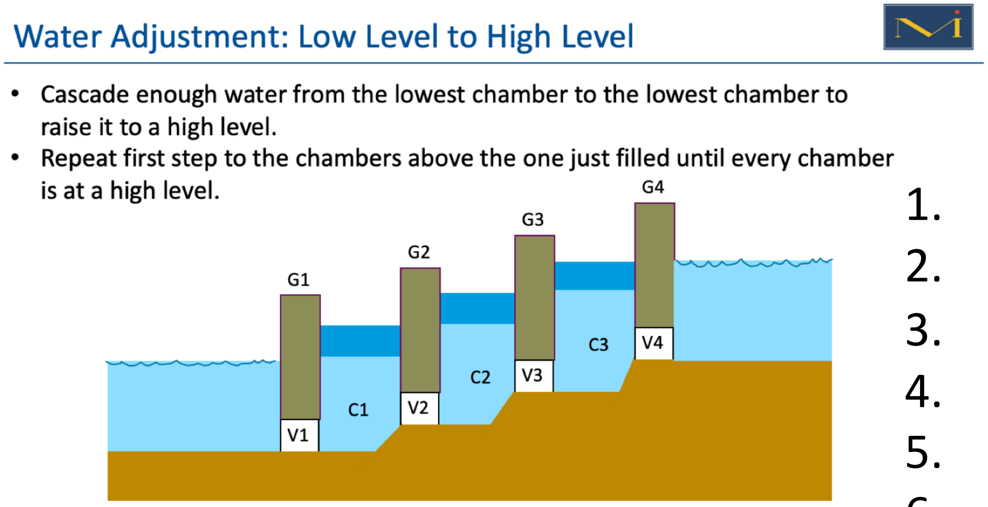


# Water Adjustment: Low Level to High Level

- Cascade enough water from the highest chamber to the lowest chamber to raise it to a high level.
- Repeat first step to the chambers above the one just filled until every chamber is at a high level.



# Operational Steps: Low Level to High Level



## Water Adjustment: Low Level to High Level

- Cascade enough water from the lowest chamber to the lowest chamber to raise it to a high level.
- Repeat first step to the chambers above the one just filled until every chamber is at a high level.

<u>Step → Details</u>

1. Raise C3 → Open V4, No Flow V4, Close V4
2. Raise C2 → Open V3, No Flow V3, Close V3
3. Raise C1 → Open V2, No Flow V2, Close V2
4. Raise C3 → Open V4, No Flow V4, Close V4
5. Raise C2 → Open V3, No Flow V3, Close V3
6. Raise C3 → Open V4, No Flow V4, Close V4

Clear Pattern

# Modeling Approaches

- **Model the pattern in data**
  - ✓
    - create a Level Adjustment Step “spec”
    - loop through “spec” with events to do each raise/lower step
  - Pro: Simple data, simple action language
  - Con: Need to stop the system to service a gate, add “spec” class

- **Model the pattern in algorithm (action language)**
    - 1st step for a raise or lower is known, do it
    - Loop through calculating the next step with events, using the raise/lower pattern and the transit lane gate configuration
  - Pro: Can service a gate without stopping the system, no new classes
  - Con: More complex action language ➔ higher probability for bugs

# Impact on Models for Modeling in Data

- **Information Model:** +1 new class, Adjustment Step
  ➔ Used same modeling pattern as Transit Lane Gate sequencing

- **Wets State Model:** No Change

- **Transit Lane State Model:** +1 state, +1 internal event, +1 external event
  ➔ Used same modeling pattern as vessel transfer

- **Transit Lane Gate State Model: +**3 states, +1 external event
  ➔ Used same modeling pattern as vessel move

- **External Entities:** No Change

No changes to previously existing (re: tested) execution threads

# Questions? Comments?? Suggestions???

# Model Pattern in Algorithm

# Patterns for Action Language Approach

- **Doing a Raise**
  - All drain sequences start at the highest gate
  - First drain sequence stops after lowest gate +1, run 1st drain sequence
  - Increment the gate to stop at, run another sequence
  - End after the next gate to stop at = highest gate

- **Doing a Lower**
  - All sequences stop at the lowest gate
  - First sequence starts at the lowest gate, run 1st drain sequence
  - Increment next gate to start at, run another sequence
  - End after the next gate to start at = highest gate

# Algorithms for Action Language Approach

**Doing a Raise**

```
  For (I =lowest gate +1 ; I > highest gate; I++) {
      For (J=highest gate; J < I; J--) {
              Drain Gate (J)
      }
  }
```

**Doing a Lower**

```
 For (I=lowest gate; I < highest gate; I++) {
      For (J=I; J < lowest gate; --) {
              Drain Gate (J)
      }
 }
```

# Impact on Models for Modeling in Algorithm

- **Information Model:** No change
- **Wets State Model:** No Change
- **Transit Lane State Model:** +1 state, +1 internal event, +1 external event
  ➔ Used same modeling pattern as vessel transfer
- **Transit Lane Gate State Model: +**3 states, +1 external event
  ➔ Used same modeling pattern as vessel move
- **External Entities:** No Change

No changes to previously existing (re: tested) execution threads