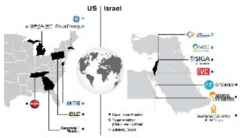
Task 2: Digital representation of physical processes and operational process modelling

Michael Faifer, Prof. Rami Puzis, Prof. Robert Moskovitch, Prof. Asaf Shabtai

Ben Gurion University of the Negev



Asaf Shabtai, PhD, CISSP

Dept. of Software and Information Systems Engineering @ BGU

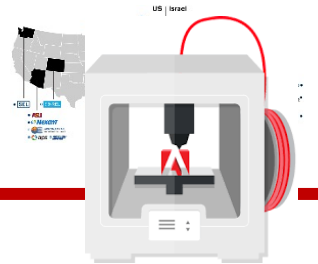
- Head of research Cyber@BGU
- Head of M.Sc. track in cyber security
- International summer camp focusing on data science and cyber security (ICSML)

Research areas

Big data security analytics

Innovative cyber-attacks

Security of medical devices



Additive manufacturing

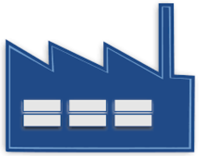
Biometric security control

Malware detection using static / dynamic analysis

Measuring the security awareness of users

Mobile device security

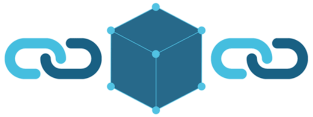
Avionic systems security (ARINC-664, 1553, ADS-B, Drones)



Pen testing and anomaly detection in OT/CPS systems (SCADA)

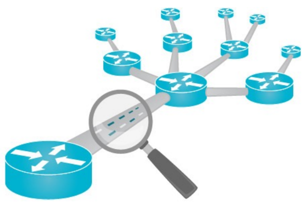
Security of replacement units

Cloud security



Using Blockchain for cyber security (IoT firmware update framework)

Data leakage and misuse detection: sensitive data representation, honeytokens, M-Score, user profiling…



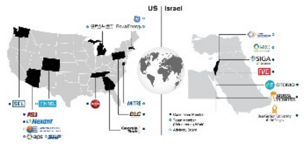
Network traffic analysis for detecting botnets, leakage, anomaly detection, device fingerprinting … (Netflow, DNS, encrypted traffic, honeypots)

Machine learning, Deep learning and Adversarial Learning

Social networks security (detecting cyber attacks, fake news, fake profiles)

IoT security (device identification, anomaly detection)

The problem: missing the operational state situational awareness

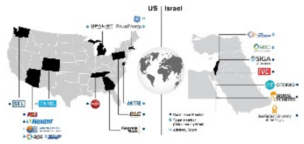


- Monitoring, detecting, and handling cybersecurity incidents in ICS
  - is based on data collected from the operational network and IT network
  - **ignores** (in most of the cases) **the operational state or the ICS system**
  - **Cannot know which control flow** was impacted by the attack

- Security personnel is **not involved** in the definition of the operational processes of the ICS; on the other hand, when designing operational processes, the **focus is on safety**; engineers are **not taking part in attack detection**

- Lack of common language for sharing OT processes

The problem: missing the operational state situational awareness



For example:

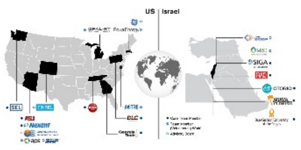
- *creating various fuels in an oil refinery*
  - *a sequence of events used to burn off excess gases:*

    "turn on flame" → "release gas" → "turn off flame"
  - *changing the order of events to*

    "turn on flame" → "turn off flame" → "release gas"

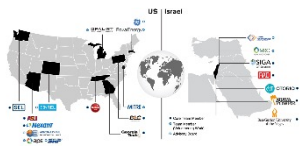
could result in the gas being continually released, potentially damaging equipment

The problem: missing the operational state situational awareness



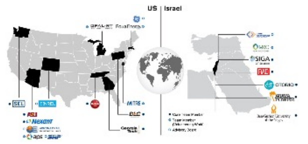
- As a result…
  - potential false alarms
  - wasted time (Investigations of incidents)
  - applying wrong countermeasures
  - miscommunications (between engineers, cyber security personal, and operators)

Research goals: providing real-time operational state situational awareness



- Creating a **relevant context** for decision making (e.g., attack detection)
- Establish **sharable modeling** language for ICS system’s operational states
- Develop a method for **modeling and defining** the states of the system
- Translating low level sensor/network data into higher level temporal patterns -- continuously
- Develop a **method for real-time, sensor-based operational state identification** using temporal patterns and temporal pattern mining
- Apply and test within ICS environments

Approach for ICS operations situational awareness



- Formulation of **common operational process enumeration** (COPE) for Industrial Control Systems (like CAPEC used for enumerating attack patterns)

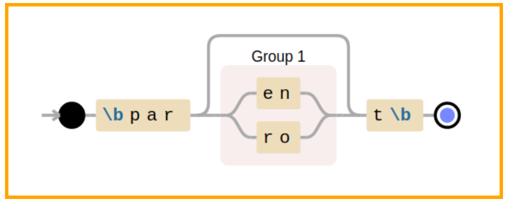
CAPEC

- COPE will be used to represent the operational processes in an ICS
  - in a **structured human readable** manner
  - while **specifying the data sources** appropriate for monitoring and identifying the process

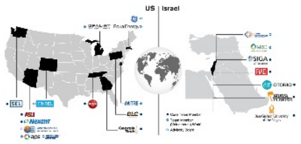
- COPE defines **shareable** information at **multiple levels of abstraction**
  - acceptable tradeoff between transparency and obscurity
  - similar processes in different ICSs share the same information

COPE

- **Name, ID**
- **Description**
- **Cope level (Tactic\Process\Low Level Process)**
- **Common Automation Level (Automatic\Manual\Both)**
- **Triggers**
- **Includes**
- **Extends**
- Process prevalence
- Impact modifiers (severity)
- Related Processes
- Execution Flow
- Prerequisites
- Skills/Resources Required
- **Required sensors/telemetry**
- **Optional Sensors**
- Related past incidents
- Example Instances
- Related Weaknesses

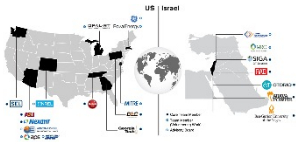


Approach for ICS operations situational awareness



- Using COPE, stakeholders can understand at any point in time the state of the ISC system
  - provide context to alerts for better understanding the risks and prioritization
  - define a process signature and detect anomalies
  - justify system behaviors and avoid false positives
  - provide COPE info when sharing threat intelligence

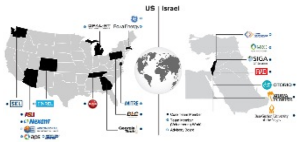
Related works



[1] Process Discovery for ICS Cyber Attack Detection (2017)

- Use **process mining** to detect ICS control flow (sequence of events, conducted by an ICS devices) anomalies
- Based on logs from PLCs
- Evaluated widely used **process discovery algorithms**: α-algorithm, the Fuzzy Miner, the ILP Miner, the Flexible Heuristics Miner (FHM), Inductive Miner; using an example setup
- Process mining-based methods operate in a form of **offline analysis**
- Some attacks may not be detected due to insufficient logging - correlate device log data and **low-level sensor data** for use in process mining based intrusion detection

Related works



[2] Anomaly detection for ICSs using process mining (2018)

- **Extending** the method presented in 2017, for **detecting anomalies**

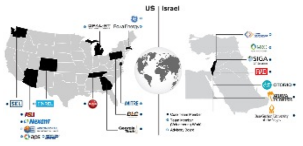
[3] Detection of Integrity Attacks to Smart Grids using Process Mining and Time-evolving Graphs (2018)

- Measurements of smart meters in smart grids
- Discover graphs from smart meter readings that represent the customer’s behaviour
- The graphs are then compared in order to detect anomalous behavior of a customer

[4] Detecting Anomalous PLC Events Using Process Mining (2022)

- Using a simulated traffic light system
- Process mining is used to create a Petri net model from the activity log
- Invalid state transition detector is created to identify anomalous

Related works



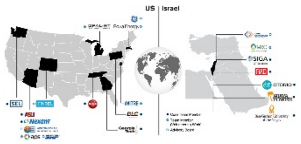
[5] Cybersecurity Analysis via Process Mining: A Systematic Literature Review (2022)

- Mentioned the **importance** of using process mining for cybersecurity
- Reviewed the usage of process mining in various domains (ICS, mobile, fraud…)

[6] 3-layer modelling method to improve the cyber resilience in ICSs (2023)

- Propose the 3-layer modelling method that reproduces ICS by the actor, asset, and process models
- Quantify the availability of ICS influenced by cyberattacks, considering the behavior of personnel involving both cybersecurity and industrial operations

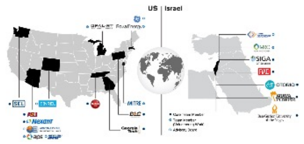
Proposed method: expert & data driven approach



- Top-Down (knowledge-based):
  - Using system description, piping and instrumentation diagram, and domain expert
  - Domain expert/process engineer defines the COPEs
  - Cannot cover all COPEs; difficult to define data-driven patterns

- Bottom-Up (data-driven):
  - Use sensory/network data of normal operation and system architecture diagrams
  - Use temporal data mining approach for finding patterns within the raw data
  - Match them meaningful identified patterns with COPEs
  - Domain expert assists in confirmation or correction

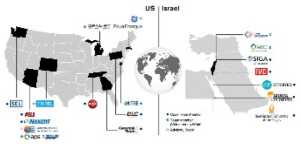
Common Attack Pattern Enumeration and Classification (CAPEC) vs Common Operational Process Enumeration (COPE)



- Attack Patterns (CAPEC)
  - Name, ID
  - Description
  - Likelihood of Attack
  - **Typical Severity**
  - Related Attack Patterns
  - Execution Flow
  - Prerequisites
  - Skills/Resources Required
  - **Indicators**
  - **Consequences**
  - Mitigations
  - Example Instances
  - Related Weaknesses

- Operational Processes (COPE)
  - **Name, ID**
  - **Description**
  - **Cope level (Tactic\Process\Low Level Process)**
  - **Common Automation Level (Automatic\Manual\Both)**
  - **Triggers**
  - **Includes**
  - **Extends**
  - Process prevalence
  - Impact modifiers (severity)
  - Related Processes
  - Execution Flow
  - Prerequisites
  - Skills/Resources Required
  - **Required sensors/telemetry**
  - **Optional Sensors**
  - Related past incidents
  - Example Instances
  - Related Weaknesses

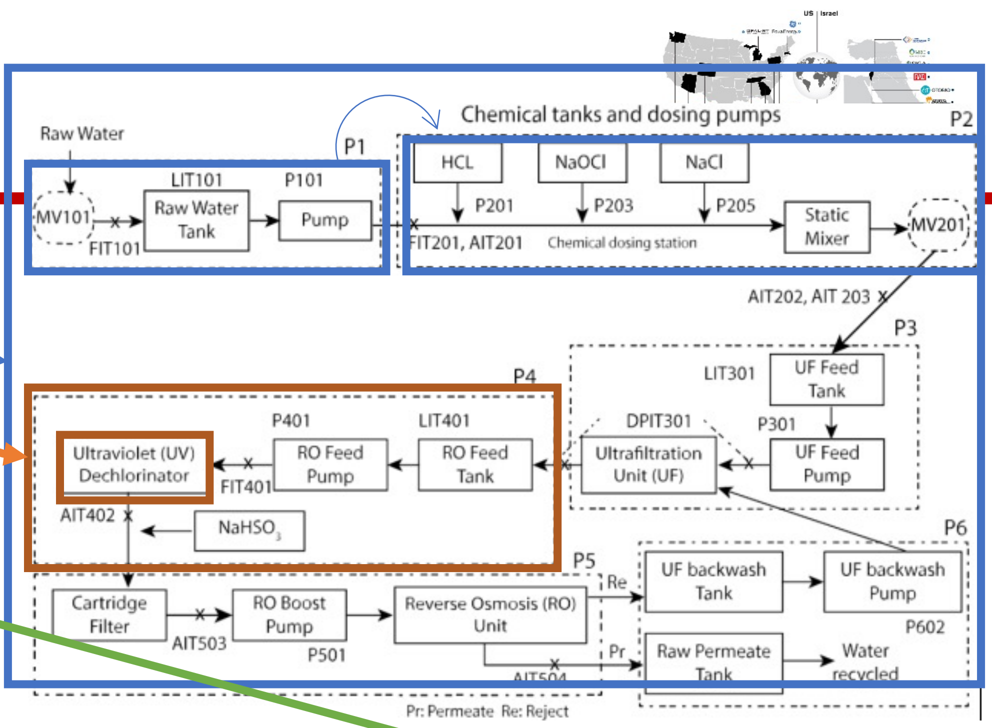
Proposed method: Main steps



- Defining COPEs
- Defining (temporal) patterns that can be used for identifying the COPEs within the raw data (sensor data, network data…)
- Looking for COPEs within raw data provided
- Identify COPEs’ instances within the data in cybersecurity tasks

Creating a COPE

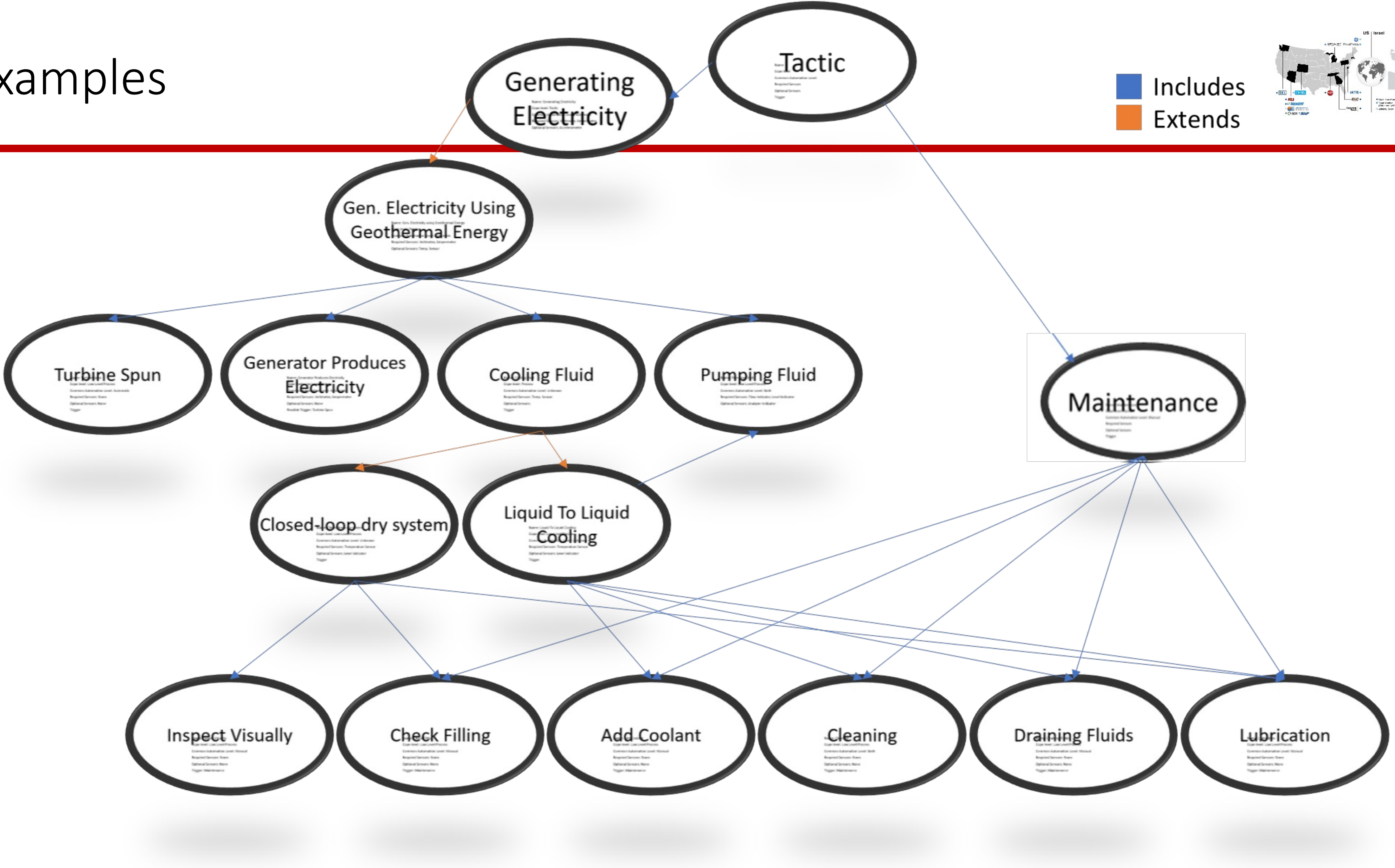
- Flow of the Process
- Flow of the Super-Process (Parent Process)
- Which sensors are involved?
- Description of the Process?
- Possible predecessor-Processes?
- ..
- ..
- consult domain experts…



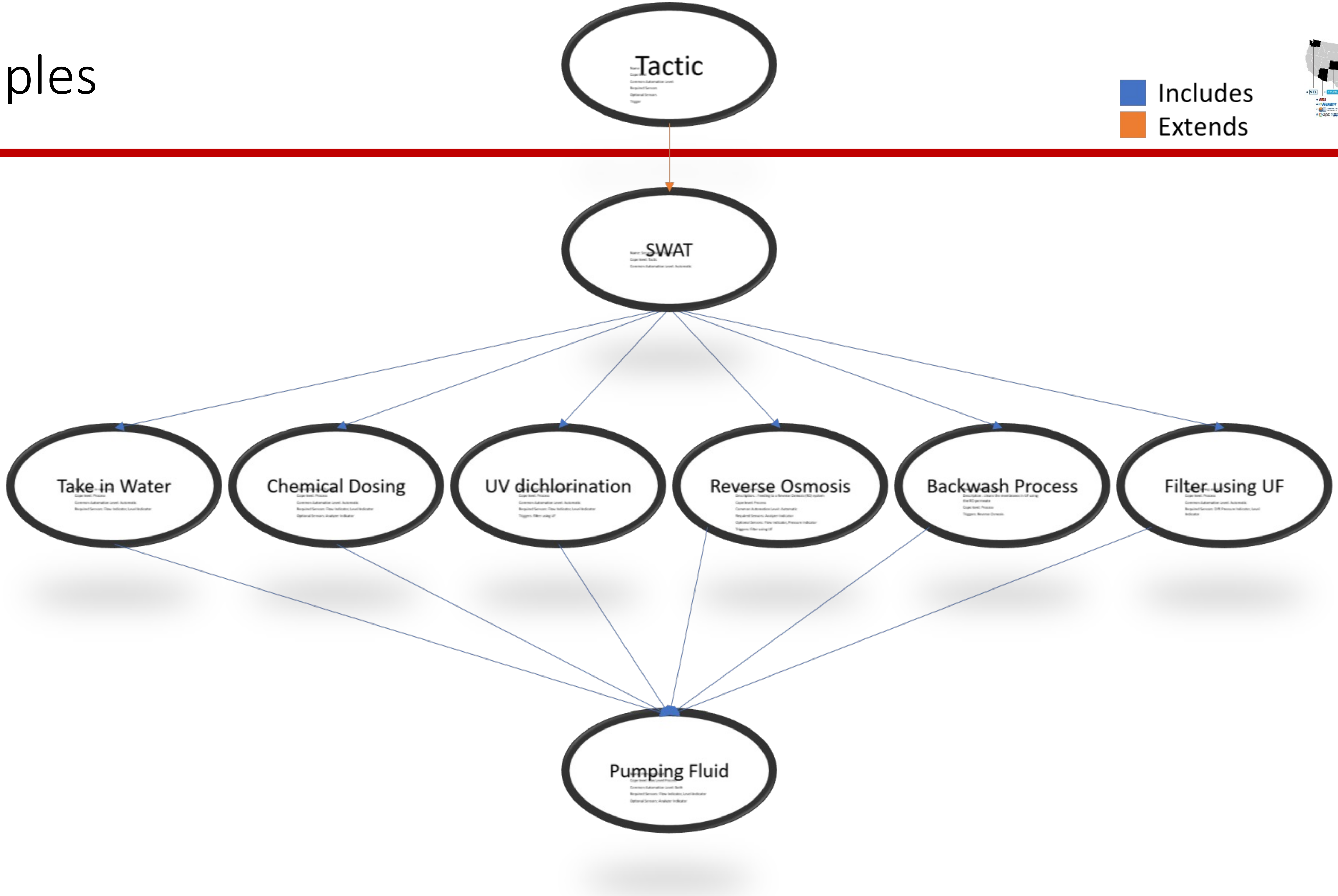
Legend

- P1: Raw water supply & storage
- P2: Chemical dosing
- P3: UF
- P4: Dechlorination
- P5: RO
- P6: RO permeate transfer, UF backwash
- AITx0y: Analyser Indicator Transmitter
- DPITTx0y: Differential Pressure Indicator Transmitter
- FITx0y: Flow Indicator Transmitter
- LITx0y: Level Indicator Transmitter
- MVx0y: Motorised Valve
- Px0y: Pump
- x = component # ; y = process module#

Examples


Generating Electricity
Tactic
Includes
Extends
Gen. Electricity Using Geothermal Energy
Turbine Spun
Generator Produces Electricity
Cooling Fluid
Pumping Fluid
Maintenance
Closed-loop dry system
Liquid To Liquid Cooling
Inspect Visually
Check Filling
Add Coolant
Cleaning
Draining Fluids
Lubrication

Examples


Tactic
SWAT
Take in Water
Chemical Dosing
UV dichlorination
Reverse Osmosis
Backwash Process
Filter using UF
Pumping Fluid
Includes
Extends

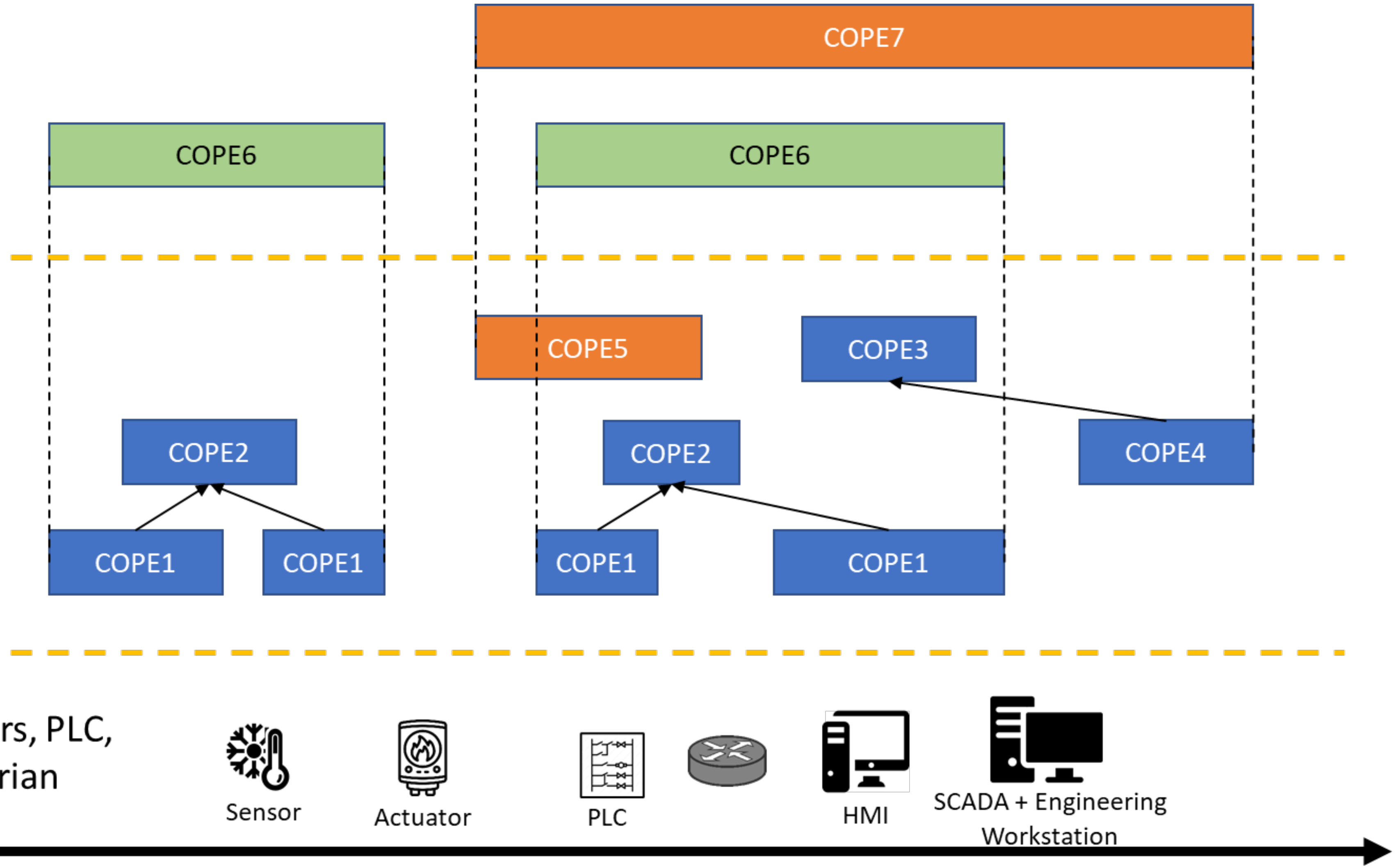
Data-driven approach

- Using **KarmaLego** – temporal pattern mining algorithm

COPE aggregation:
situational awareness

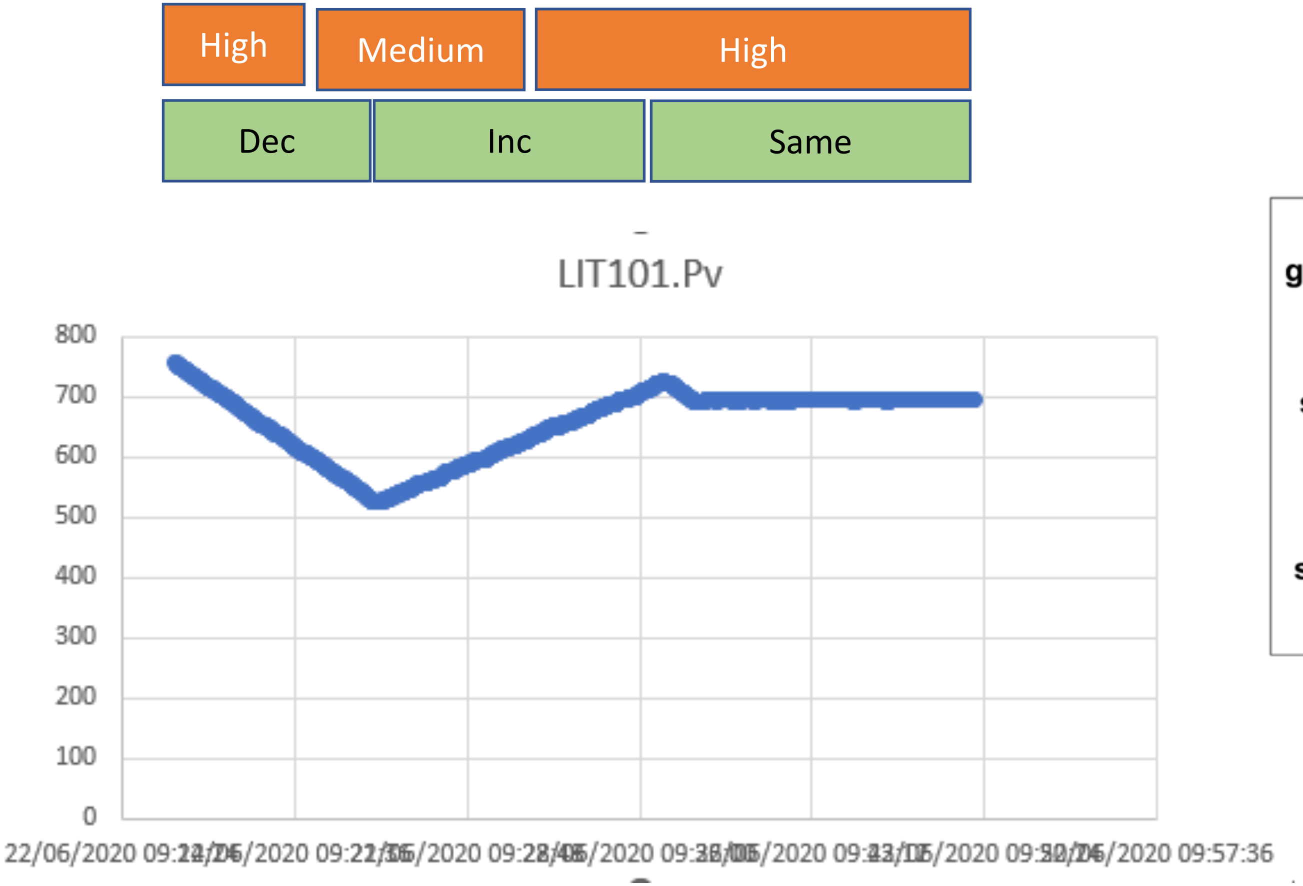
COPEs:

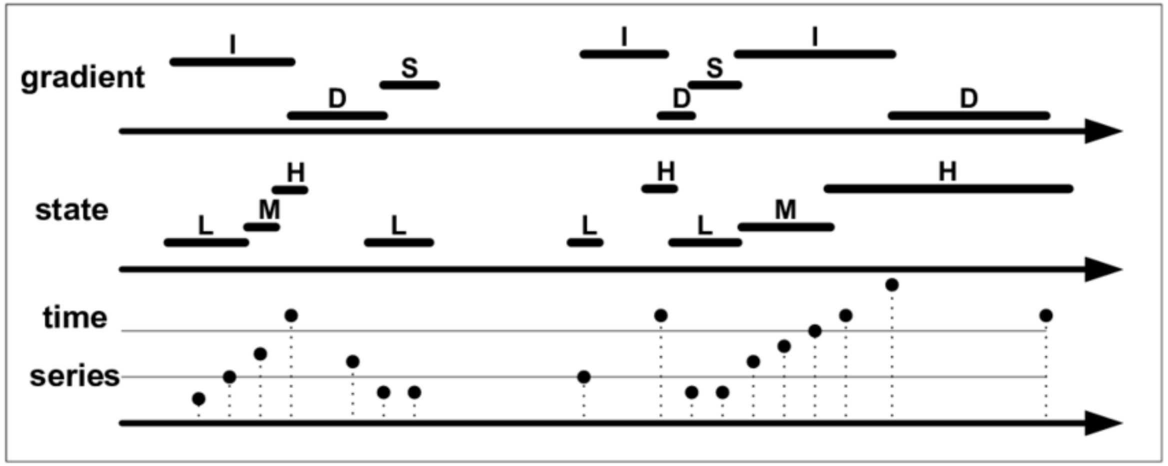
Low-level data: Sensors, actuators, PLC, network traffic (SCADA, IT) Historian



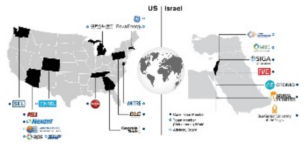
KarmaLego - illustration

- First step – defining temporal abstractions

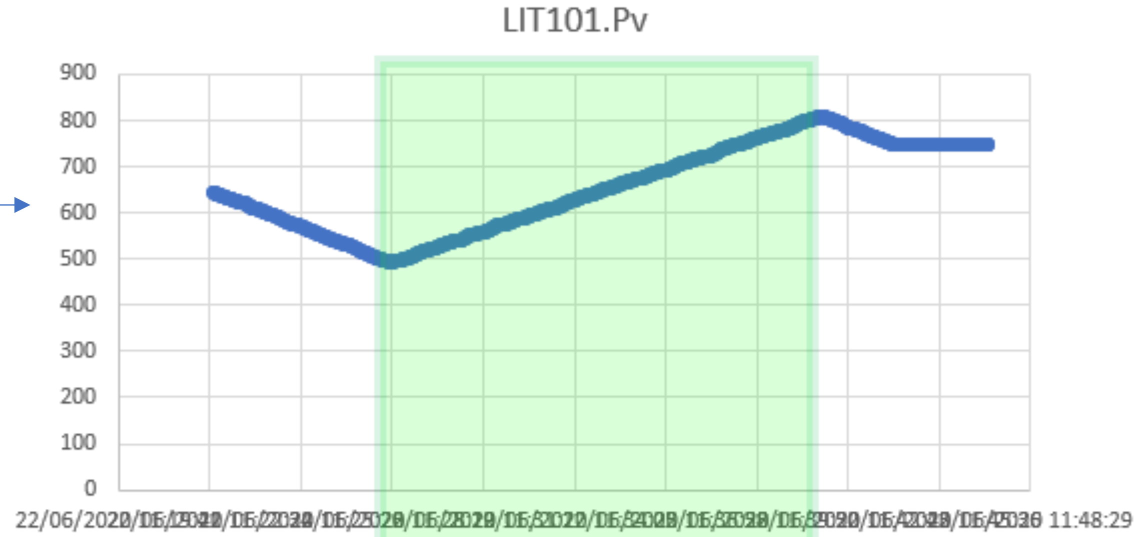




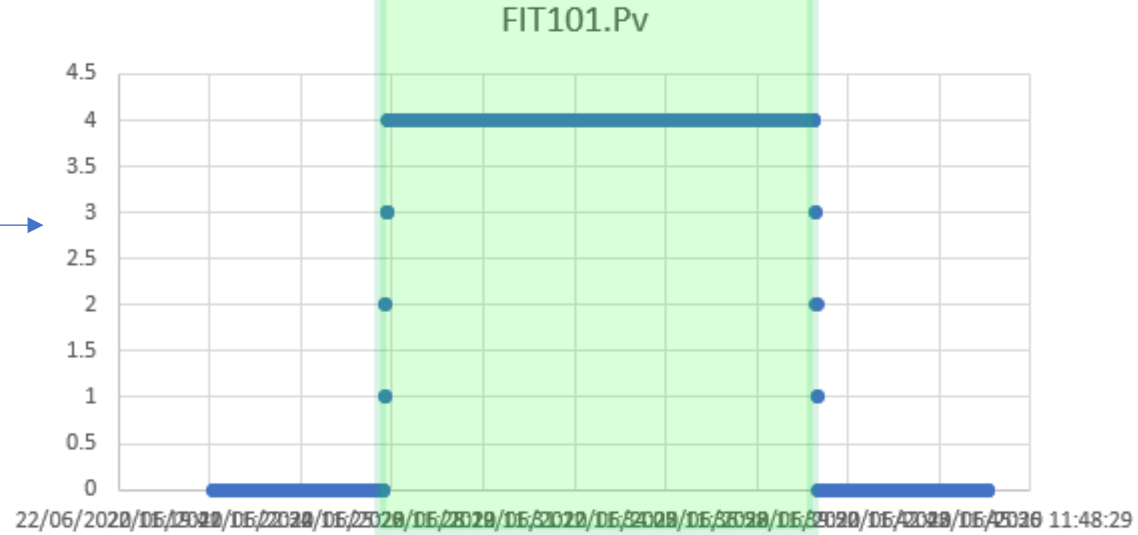
Example: Filling water COPE pattern



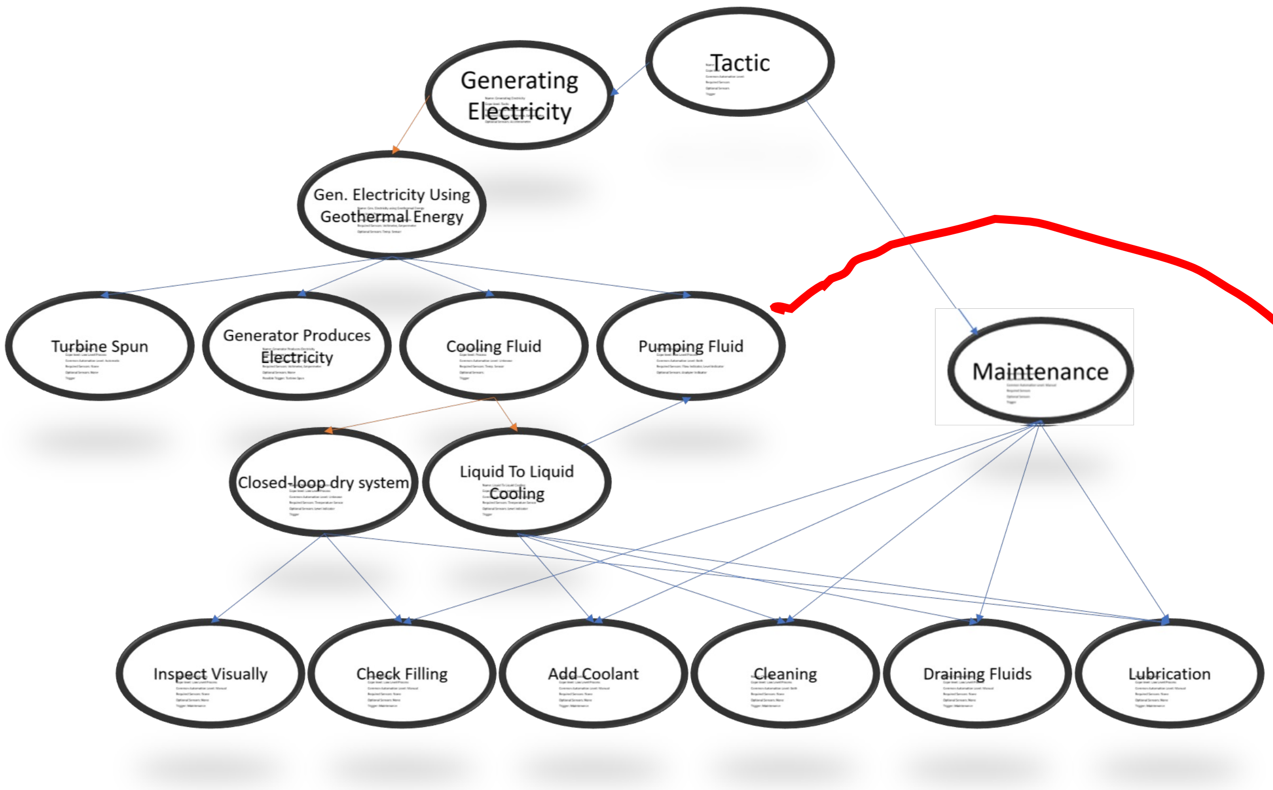
LEVEL SENSOR
INCREASING

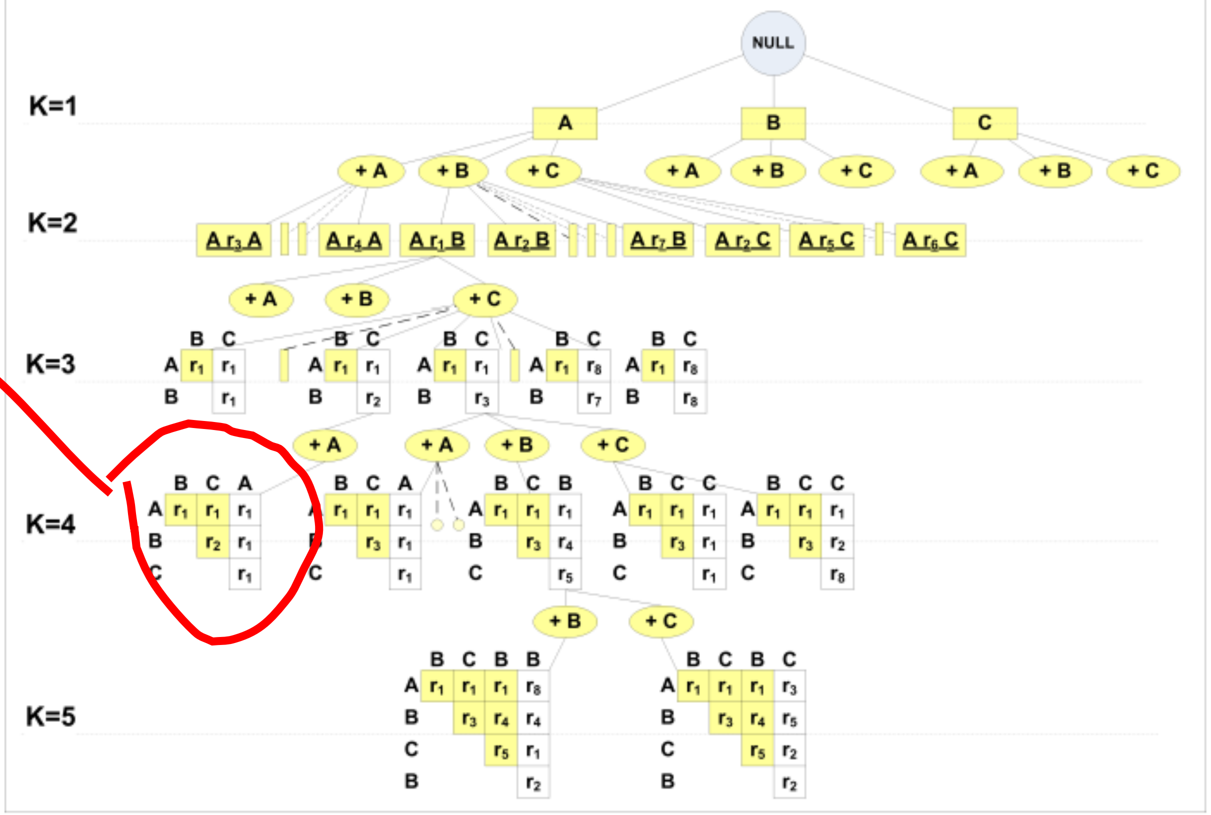


HIGH FLOW SENSOR
(CONSTANT ON 4.0)

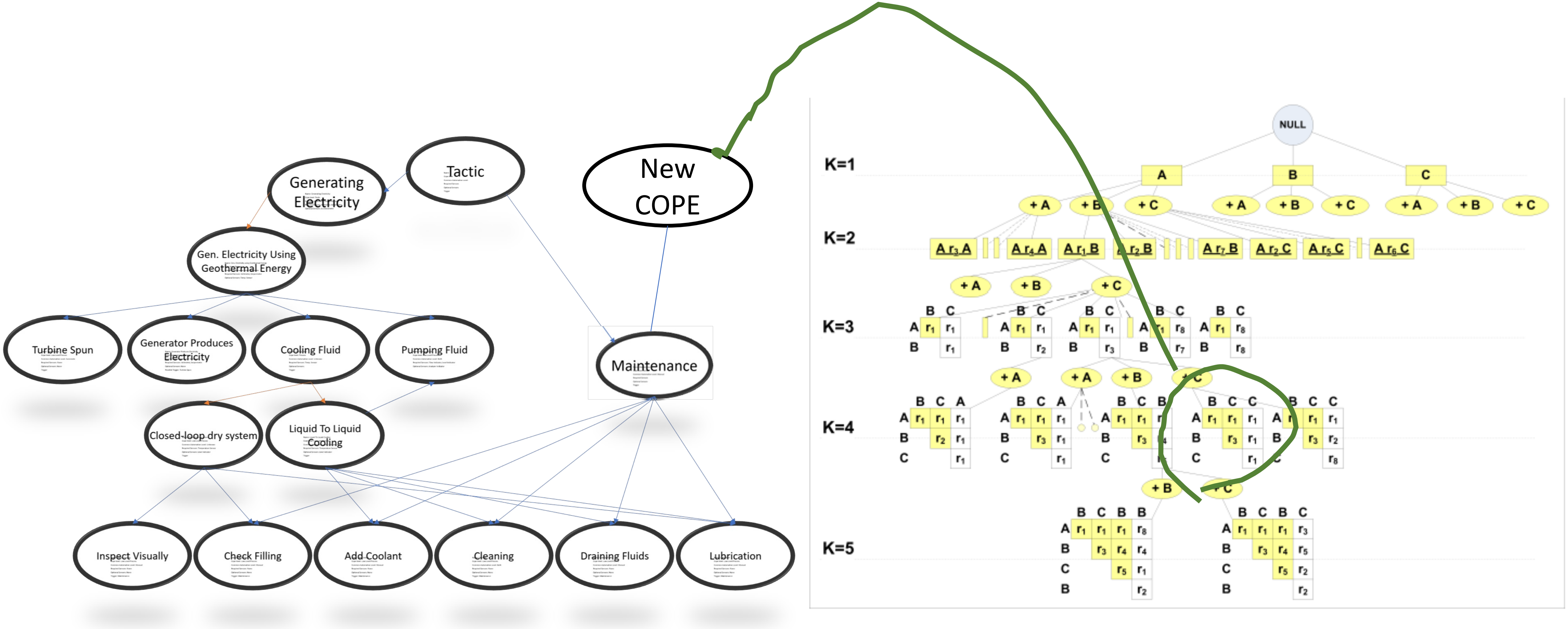


Pattern mapping to COPEs

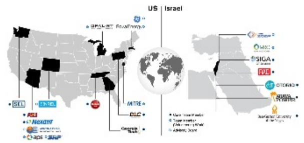


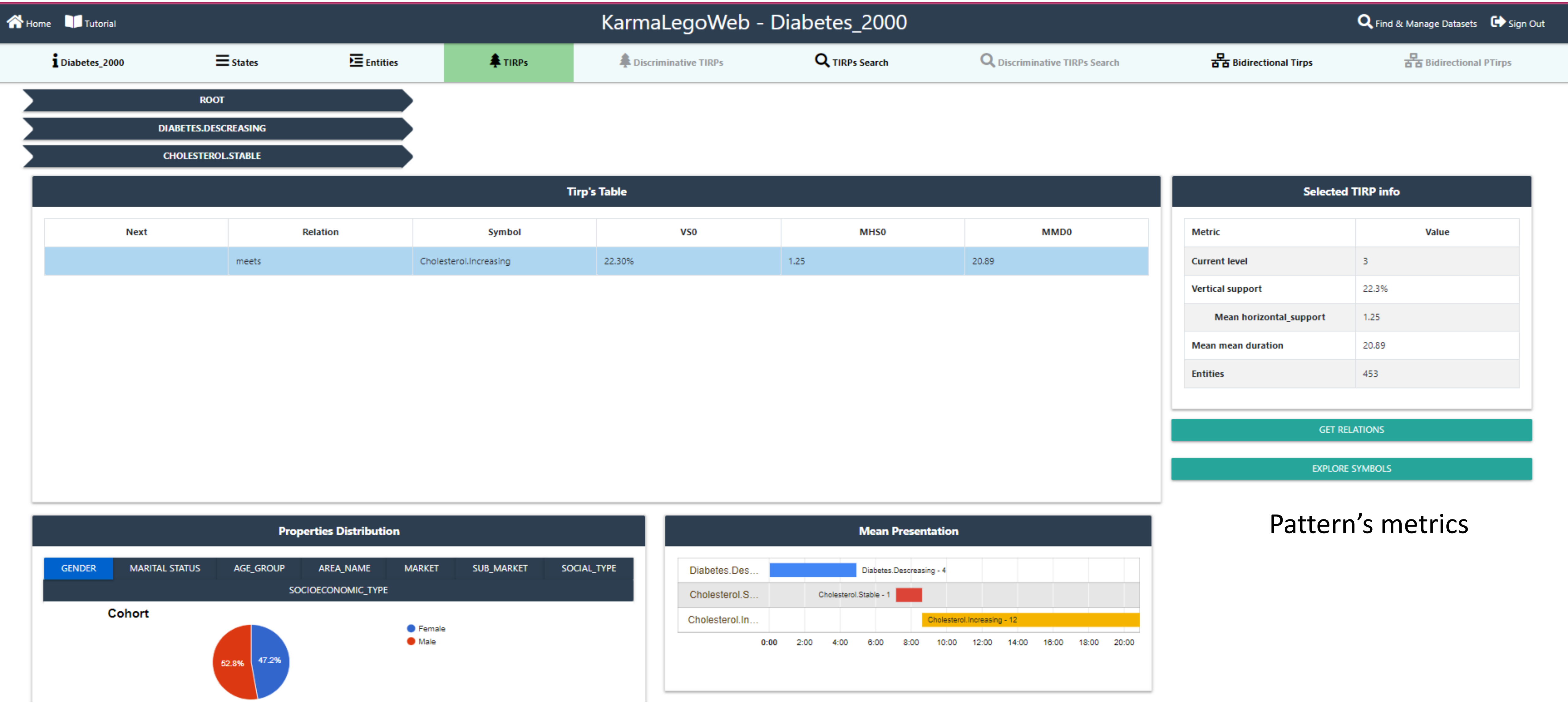


Pattern mapping to COPEs



Visualization of Frequent Patterns – Tabular View





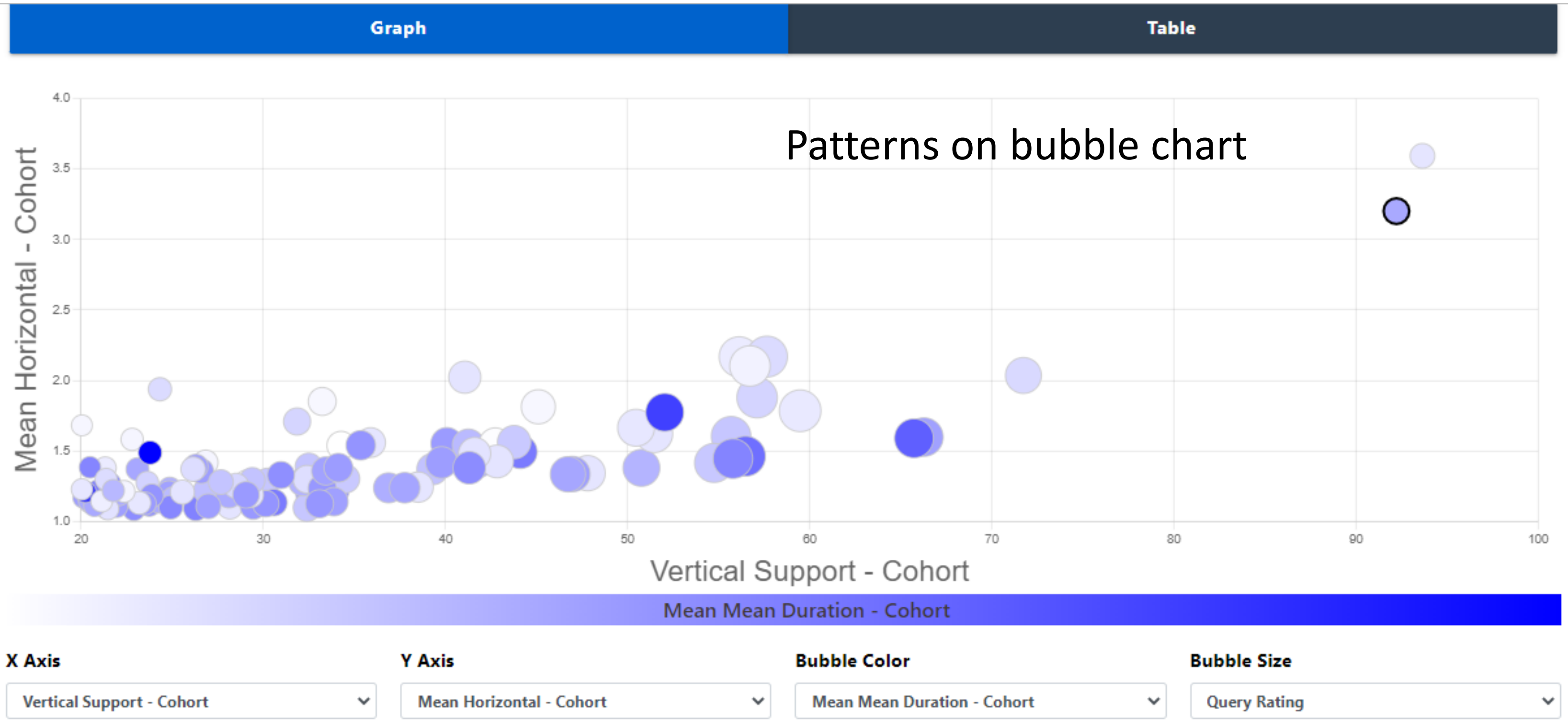
Pattern's metrics

Properties distribution

Visualization of Explored Pattern

Visualization of Frequent Patterns – Graphical View

Patterns on bubble chart

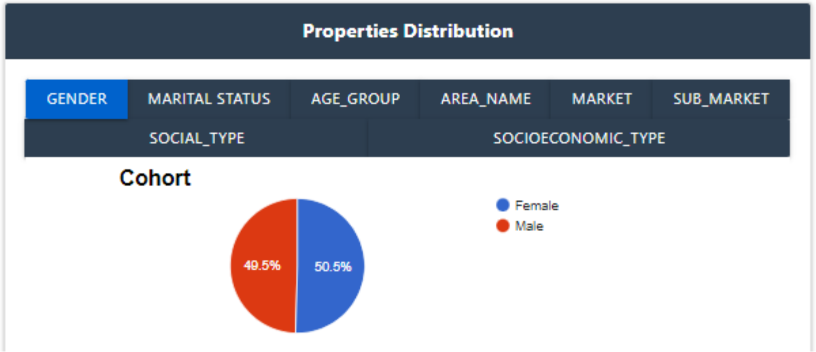


Selected TIRP info

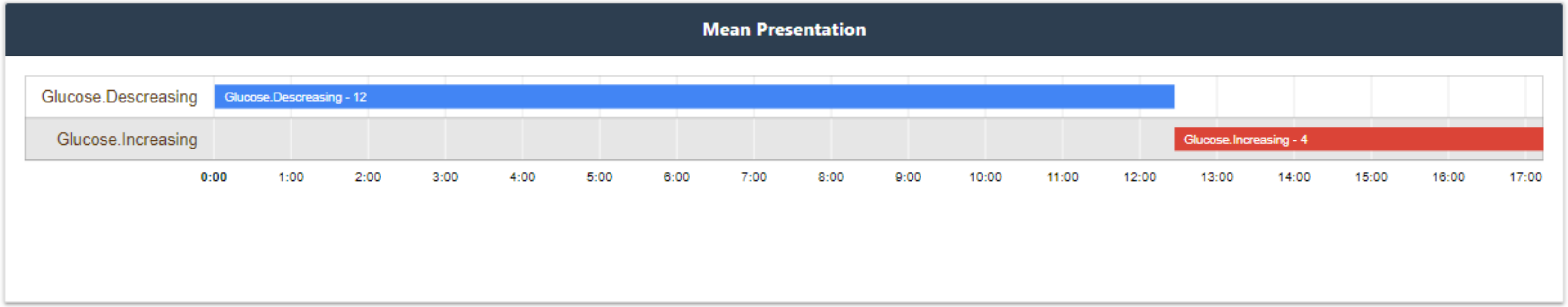
Metric	Value
Total Levels	2
V.S	92.22
M.H.S	3.20
M.M.D	12.46

EXPLORE TIRP

Pattern’s metrics

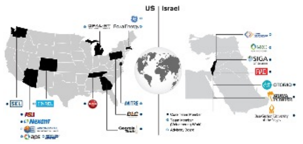


Properties distribution



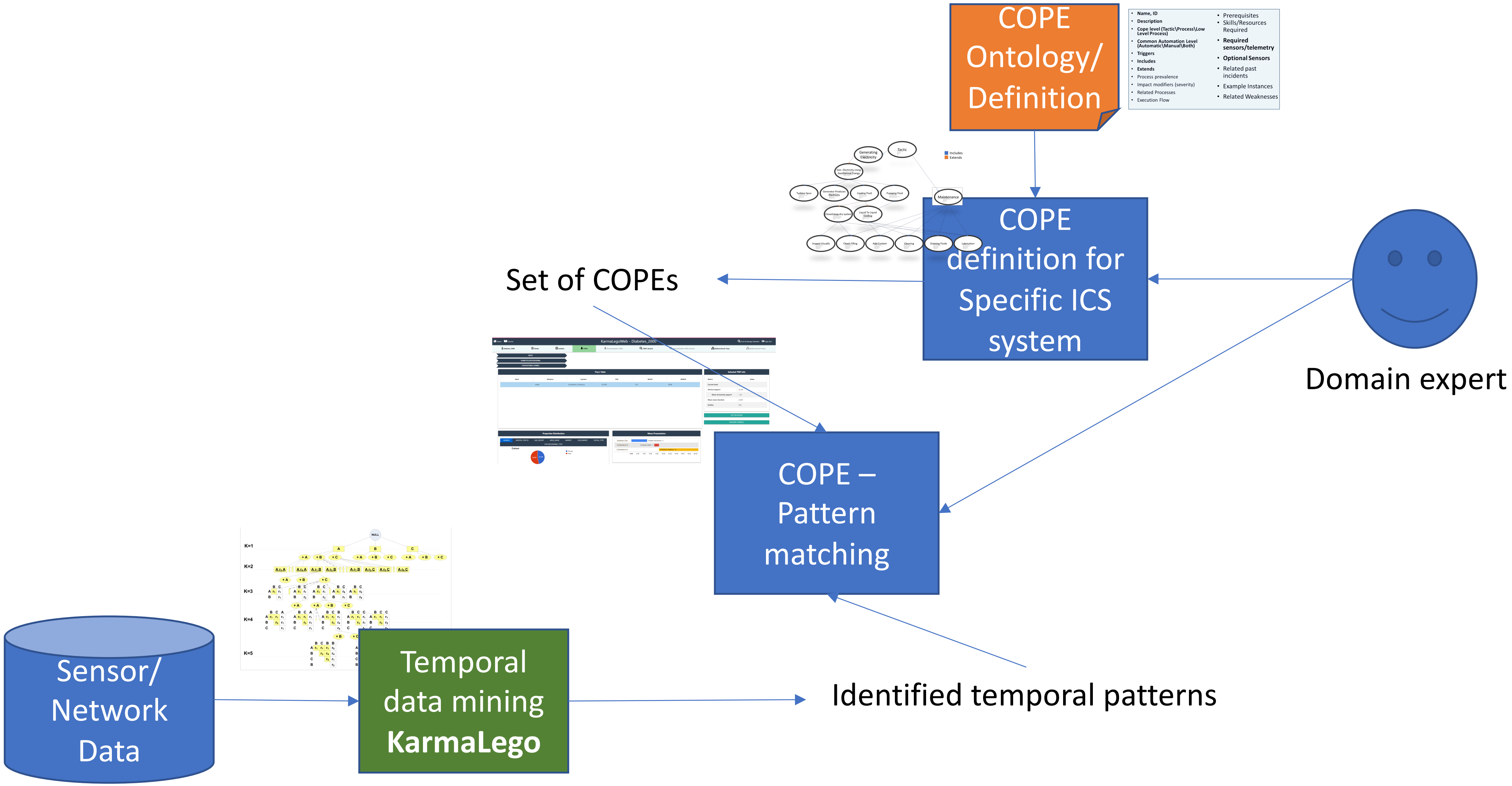
Visualization of Explored Pattern

Proposed method: Main steps

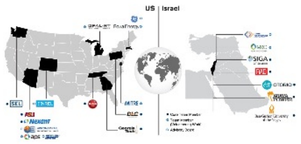


- Defining COPEs
- Defining (temporal) patterns that can be used for identifying the COPEs within the raw data (sensor data, network data…)
  - define temporal abstractions on raw data
  - apply Karmalego algorithm on the temporal abstractions and identifying temporal patterns at different levels of abstractions
- Looking for COPEs within raw data provided
  - Using an existing advanced visualization tool for investigating the patterns: (1) link between an identified pattern and predefined COPE; (2) identify interesting pattern and define it as a COPE
- Utilizing COPEs and identified instances within the data in cybersecurity tasks
  - Anomaly/attack detection

Proposed framework



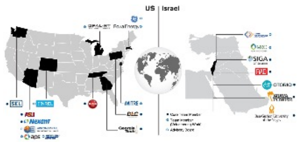
Dataset → SWaT (2015-2021)

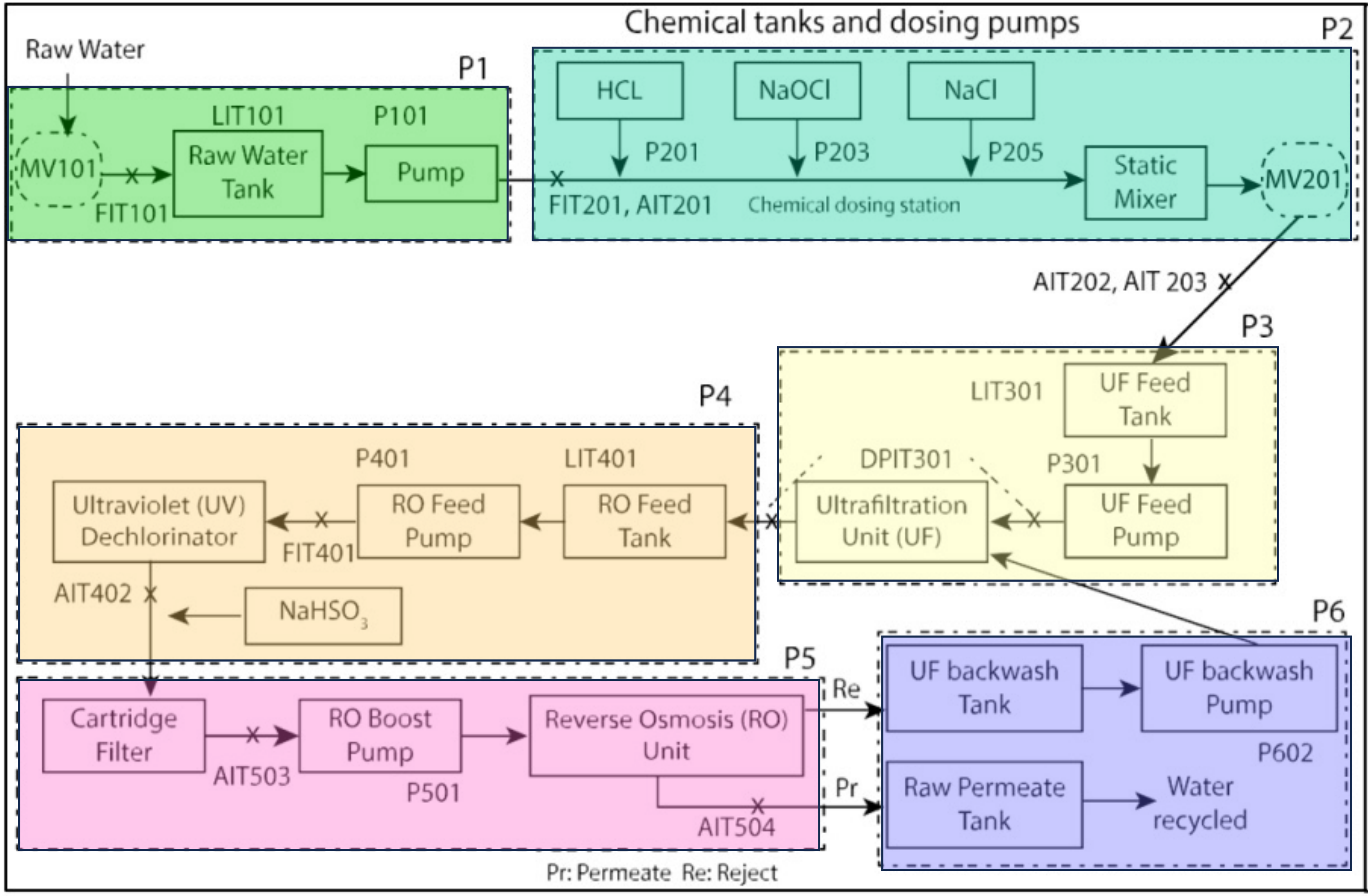


- SWaT → Secure Water Treatment Testbed
- 6 Stages (Intake, Filtering, UV, Reverse Osmosis, Backwash)
- 49 Sensors
- 11 Days of continuous operation
- Access to Raw Data

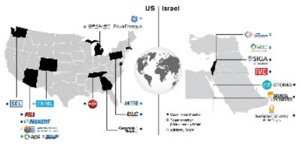
Fig. 1: Actual Photograph of SWaT testbed

Dataset → SWaT (2015-2021)

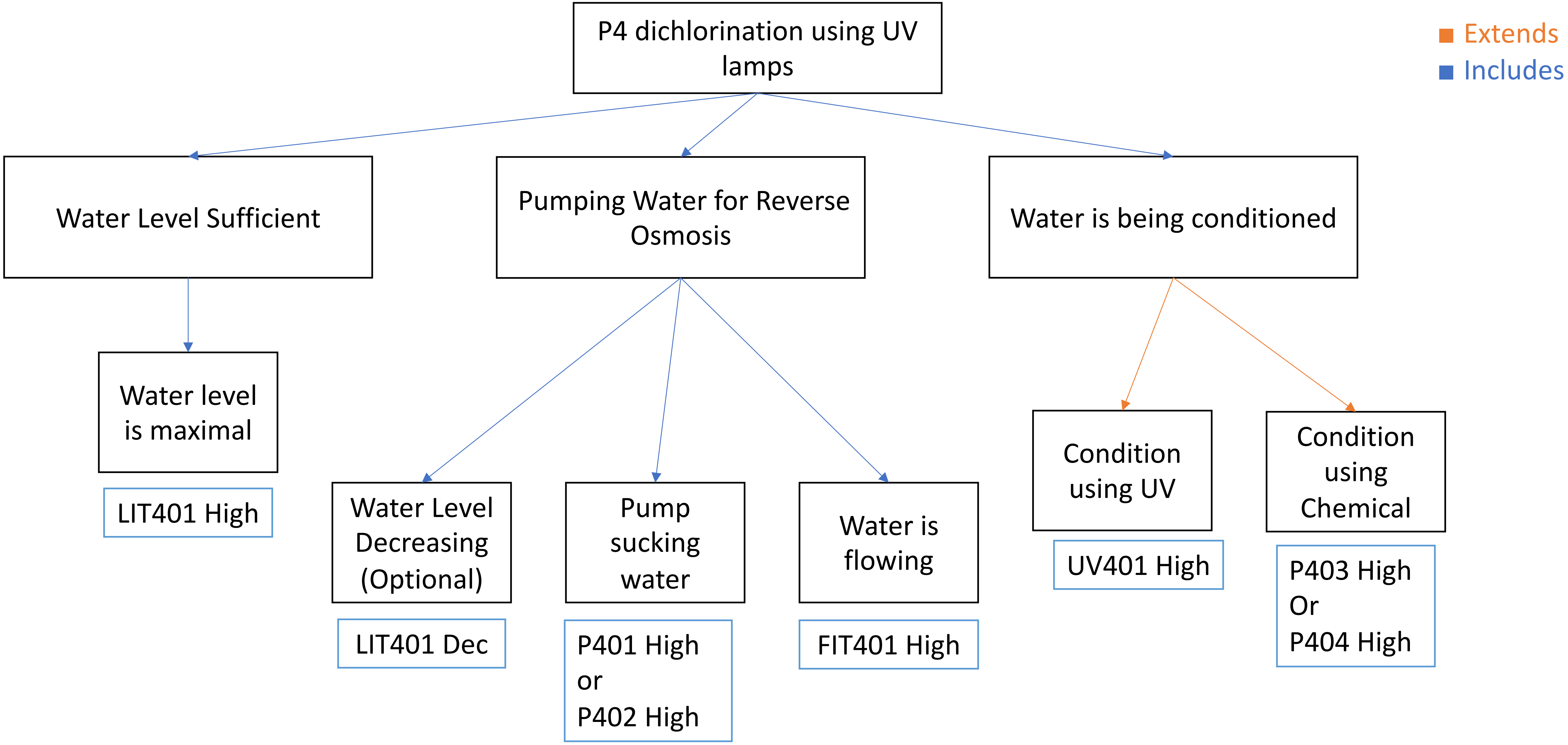




Results



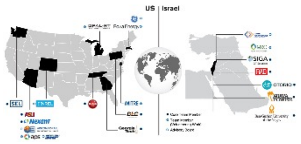
- 61 COPEs were defined by the expert (i.e., the expert-based phase)
  - Coverage of 26 sensors/actuators (Out of 49)
- KarmaLego detected ~20K patterns; only 162 of them were relevant (involving the relevant sensors)
  - Requires Pre-Processing (data abstraction) using EWD, EFD, SAX, Gradient, etc.
- Following the investigation of the generated patterns, additional 24 new COPEs were identified
- **85 COPEs in total**
- During the manual investigation we were able to match 74 temporal patterns and COPEs
- 87% success rate; 54% false patterns

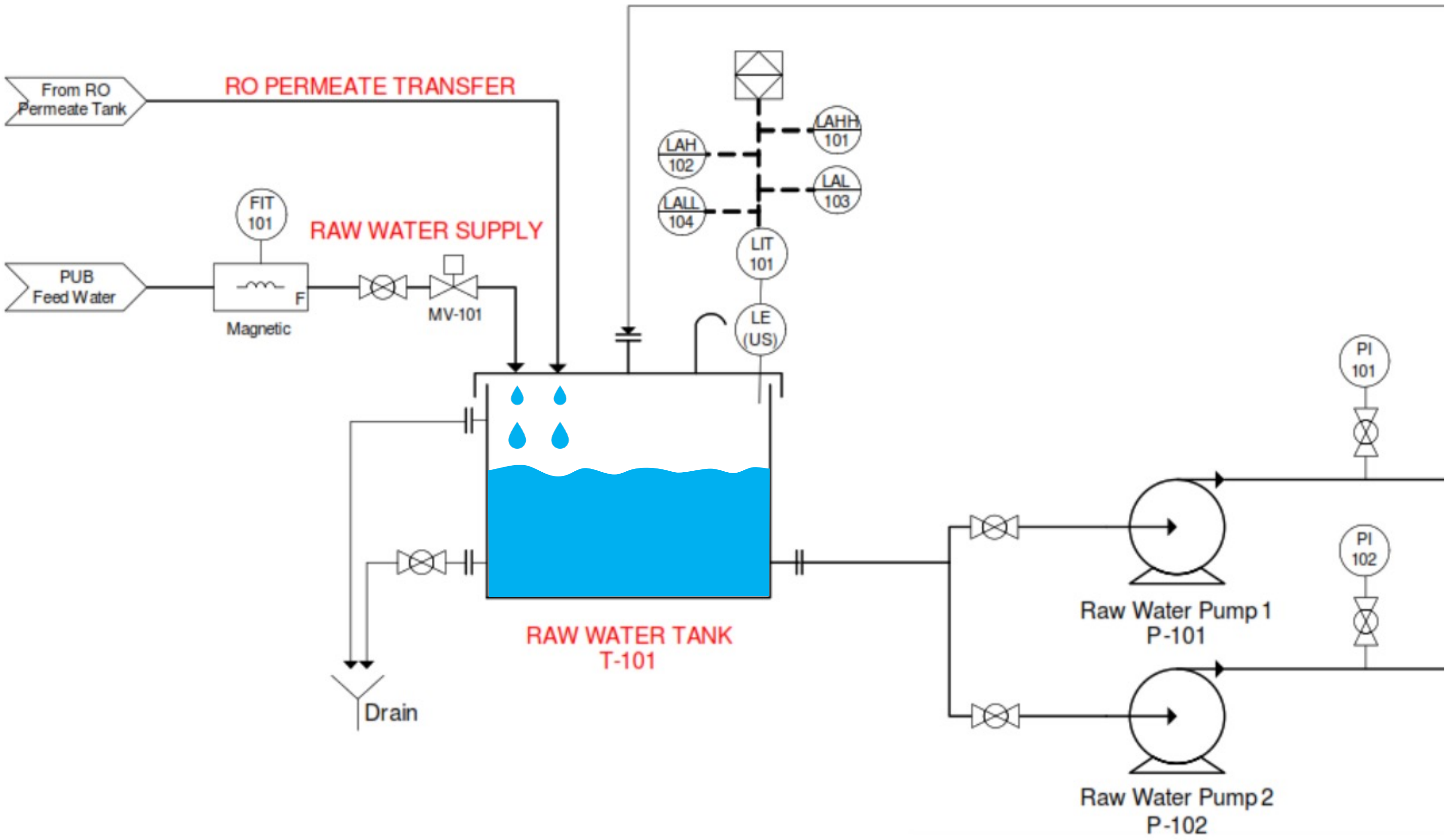

Extends
Includes
P4 dichlorination using UV lamps
Water Level Sufficient
Pumping Water for Reverse Osmosis
Water is being conditioned
Water level is maximal
LIT401 High
Water Level Decreasing (Optional)
LIT401 Dec
Pump sucking water
P401 High
or
P402 High
Water is flowing
FIT401 High
Condition using UV
UV401 High
Condition using Chemical
P403 High
Or
P404 High

Results: examples

ID	Name	Is Abstract	Based On	Description	Origin	Symbols	Image
62	Tank is draining and not refilled after being filled to max.	×	C	Tank was filled to Max, stopped to refill and started to drain only	DD	LIT101.HIGH, MV101.CLOSED, FIT101.NOFLOW, P101.ON, LIT101.MEDIUM	
63	Tank is draining and not refilled after being filled to max.	×	C	Tank was filled to Max, stopped to refill and started to drain only	DD	LIT101.HIGH, MV101.CLOSED, FIT101.NOFLOW, LIT101_GRAD.DECREASING, LIT101.MEDIUM	
64	Tank is draining and not refilled after being filled to max.	×	C	Tank was filled to Max, stopped to refill and started to drain only	DD	LIT101.HIGH, MV101.CLOSED, FIT101.NOFLOW, MV201.OPEN, LIT101.MEDIUM	
65	Maxed Tank is draining to medium and not re-filled.	×	C	Tank was filled to Max, stopped to refill and started to drain only	DD	LIT101.HIGH, P101.ON, MV201.OPEN, LIT101_GRAD.DECREASING, AIT202ABS.LOW, LIT101.MEDIUM	
66	Emptied Tank is re-filled to medium without draining	×	C	Emptied Tank started to fill rapidly with-out being sucked out.	DD	LIT101.LOW, P101.OFF, LIT101_GRAD.RAPID_INCREASING, LIT101.MEDIUM	

Example – water intake





Includes
Extends

P1 filling the water tank

Water tank being drained

water can flow **out**

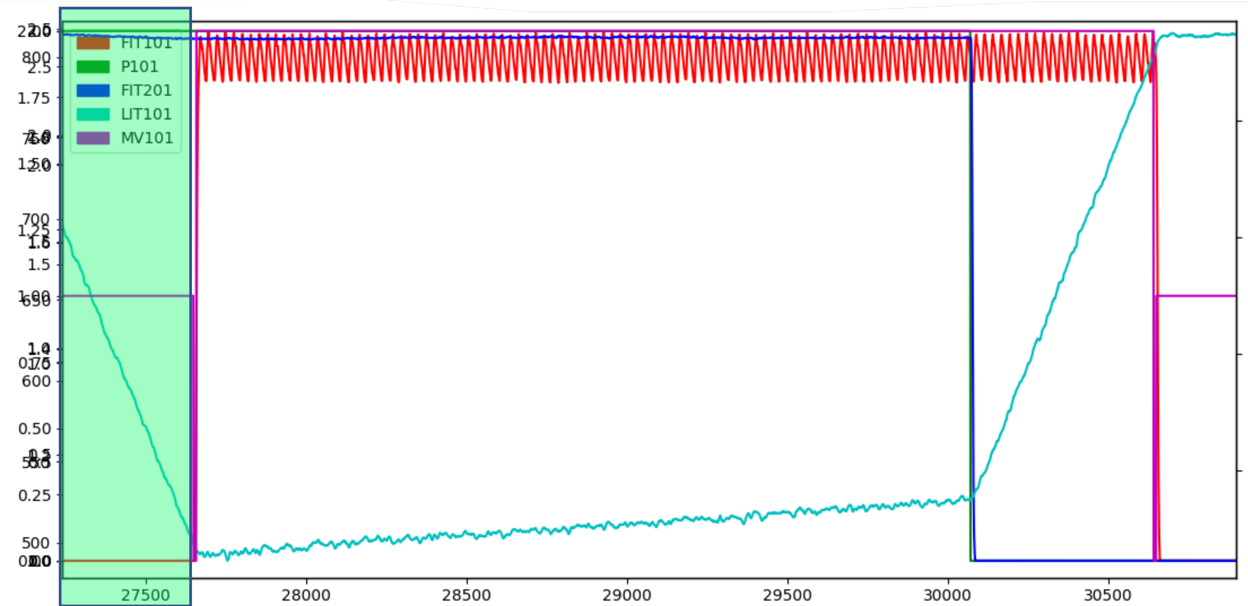
P101 OPEN &
FIT201 FLOW

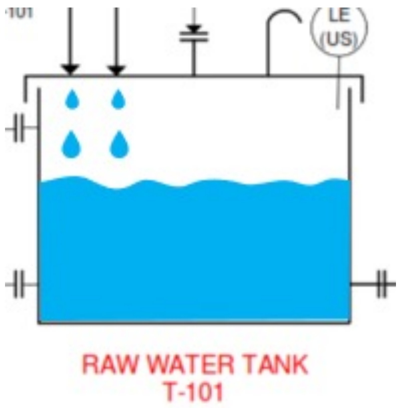
Tank draining

LIT101 DECRESING

No inflow

FIT101 LOW





Includes
Extends

P1 filling the water tank

Water tank being drained

water can flow **out**

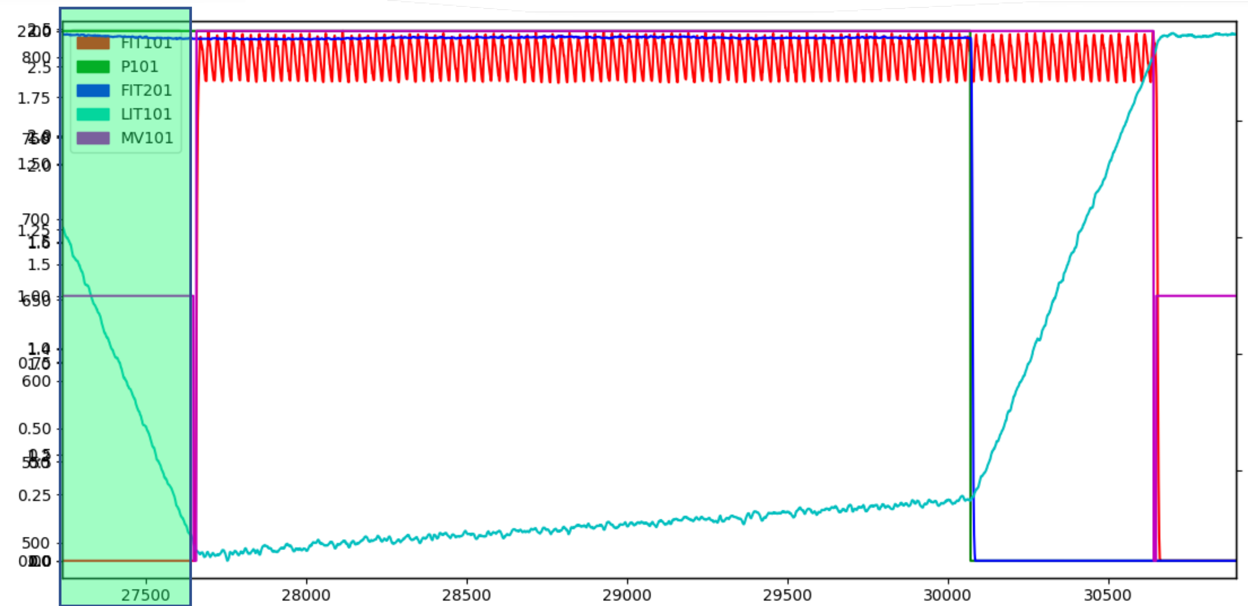
P101 OPEN &
FIT201 FLOW

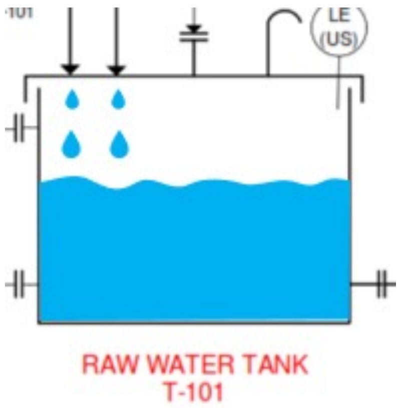
Tank draining

LIT101 DECRESING

No inflow

FIT101 LOW





Includes
Extends

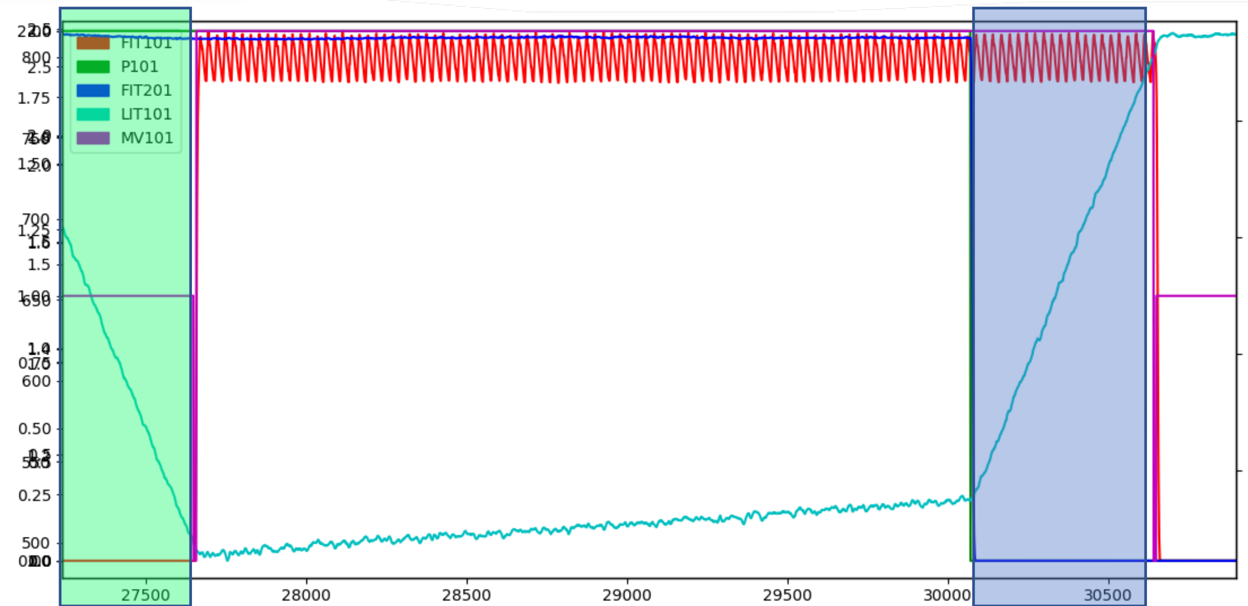
P1 filling the water tank

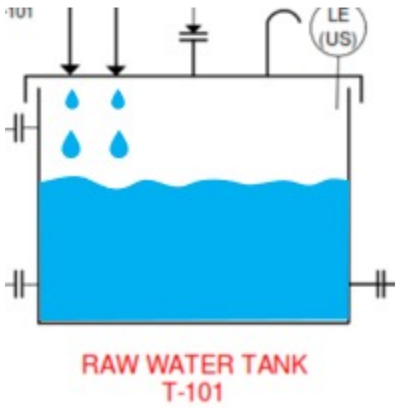
Water tank being drained

- water can flow **out** — P101 OPEN & FIT201 FLOW
- Tank draining — LIT101 DECRESING
- No inflow — FIT101 LOW

Water tank being filled

- water can flow **inside** — MV101 OPEN & FIT101 FLOW
- Tank filling up — LIT101 INCREASING
- No out flow — P101 OFF & FIT201 LOW





Includes
Extends

P1 filling the water tank

Water tank being filled

water can flow inside

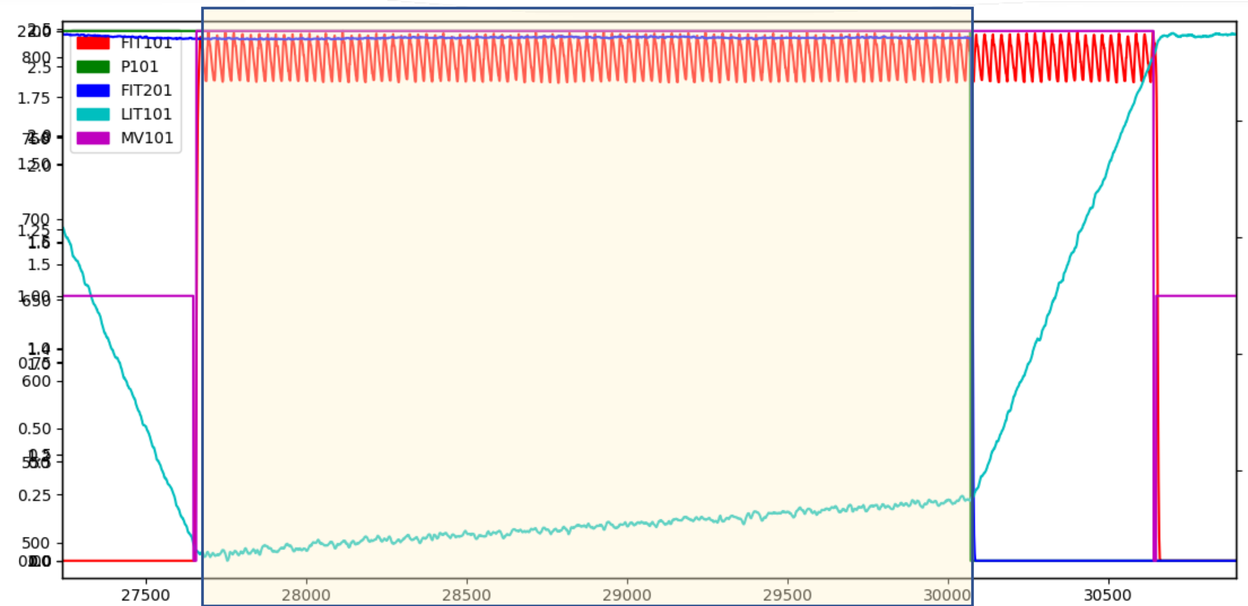
MV101 OPEN &
FIT101 FLOW

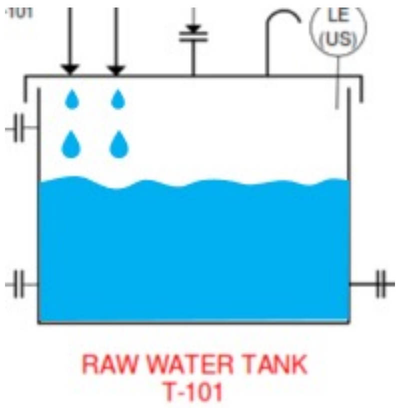
Tank filling up

LIT101
INCREASING SLOW

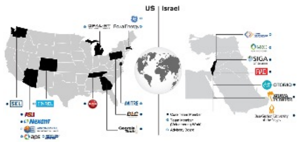
Water can flow out

P101 ON &
FIT201 HIGH





Conclusions



- COPEs – good foundation for representing ICS processes
- A COPE may have several possibilities for defining patterns
  - Usage of different set of sensors
  - Different state of said cope (draining hot water vs draining cold water)
- Needs to improve coverage
- Next steps
  - Implement on additional cases/ICSs
  - Integrate within an anomaly/attack detection task

Thank you!