

ParallelSimulations
+ scenarios: list(WaterDistributionNetworkScenarioSimulator)
+ __init__(scenarios) + run_all(n_max_cpu_cores, max_memory_consumption)

ScenarioConfigurationParser
+ parse_config(): list(WaterDistributionNetworkScenarioSimulator)

epyt_flow.data.networks
+ load_hanoi(demand_profile, download_dir): WaterDistributionNetworkScenarioSimulator
+ load_itown(demand_profile, download_dir): WaterDistributionNetworkScenarioSimulator
+ load_net1(download_dir): WaterDistributionNetworkScenarioSimulator

epyt_flow.data.scenarios
+ load_leakdb(scenario_id, download_dir): ScenarioResults
+ load_battledim(download_dir): ScenarioResults

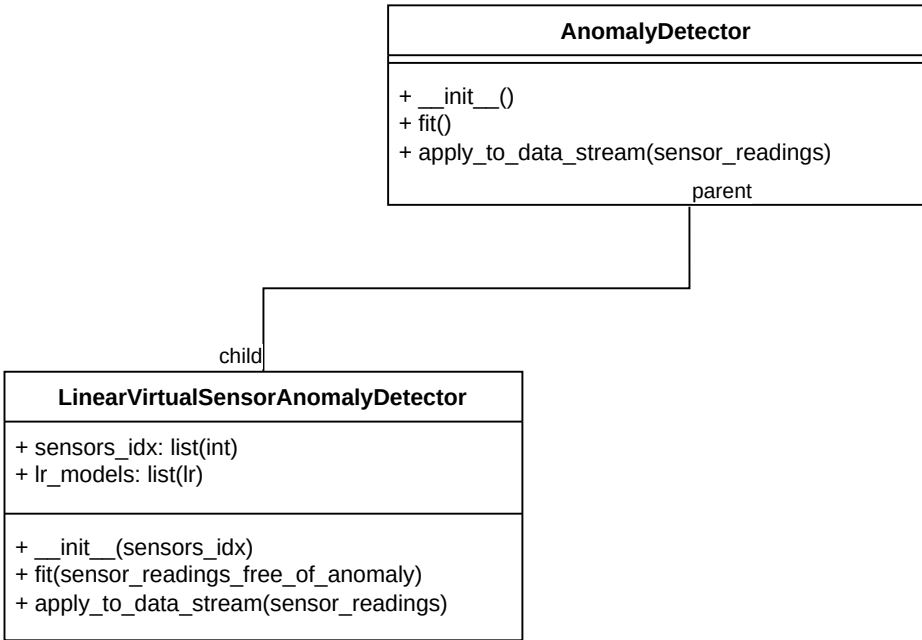
ScenarioResults
+ senario_info: ScenarioInfo + sensor_readings
+ load(file_in) + to_numpy() + export(DataExport) + get_anomalous_time_points(): list(int)

ScenarioInfo
+ inp_file: str + msx_file: str + general_params: dict + sensor_config: SensorConfig + epanet_events: list(EpanetEvent) + sensor_events: list(SensorReadingEvent)
+ __init__(...)

WaterDistributionNetworkScenarioSimulator
+ epanet: EpyT-MSX + sensor_config: SensorConfig + epanet_events: list(EpanetEvent) + sensor_events: list(SensorReadingEvent)
+ __init__(inp_file, msx_file) + get_estimate_memory_consumption() + get_scenario_info(): ScenarioInfo + set_general_parameters(demand_model, quality_type, time_steps, ...) + randomize_demands() + add_demand_uncertainty(Uncertainty) + add_epanet_event(EpanetEvent) + add_leakage(Leak) + add_sensor_reading_event(SensorReadingEvent) + add_sensor_fault(SensorFault) + add_sensor_uncertainty(SensorUncertainty) + add_simple_chlorine_injection(node_id, source_type, pattern) + run_simulation(DataExport, hyd_export=False) + run_advanced_quality_simulation(HydData, DataExport)

ScenarioVisualizer
+ scenario: WaterDistributionNetworkScenarioSimulator
+ __init__(WaterDistributionNetworkScenarioSimulator) + plot_network_topology() + plot_animation(time_window)

epyt_flow.anomaly_detectors
+ get_linear_detector(): AnomalyDetector
+ ...
+



SensorConfig
+ pressure_sensors: list(int) + flow_sensors: list(int) + ...

