Risk Model Module Documentation

1 Overview

This module defines classes and functions for risk modeling and calculation, focusing on asset-level risk assessment across various hazards, scenarios, and years. It includes base classes for risk models, risk measure calculators, and implementations for asset-level risk assessment.

2 Key Components

2.1 Data Structures

2.1.1 `BatchId` (NamedTuple)

- `scenario` (str): The scenario being modeled
- `key_year` (Optional[int]): The year of the scenario (None for historical)

2.1.2 `MeasureKey` (NamedTuple)

- `asset` (Asset): The asset being assessed
- `prosp_scen` (str): The prospective scenario
- `year` (int): The year of assessment
- `hazard_type` (type): The type of hazard

2.1.3 `Measure` (dataclass)

- `score` (Category): The risk category
- `measure_0` (float): A numerical measure of risk
- `definition` (ScoreBasedRiskMeasureDefinition): Reference to the measure definition

2.2 Classes

2.2.1 `RiskModel` (Base Class)

Base class for risk models that use hazard and vulnerability models.

Methods:

- `__init__(self, hazard_model: HazardModel, vulnerability_models: VulnerabilityModels)`
- `calculate_risk_measures(self, assets: Sequence[Asset], prosp_scens: Sequence[str], years: Sequence[int])` (abstract)
- `_calculate_all_impacts(self, assets: Sequence[Asset], prosp_scens: Sequence[str], years: Sequence[int], include_histo: bool = False)`
- `_calculate_single_impact(self, assets: Sequence[Asset], scenario: str, year: int)`

2.2.2 `RiskMeasureCalculator` (Protocol)

Protocol defining the interface for risk measure calculators.

Methods:

- `calc_measure(self, hazard_type: Type[Hazard], base_impact: AssetImpactResult, impact: AssetImpactResult) -> Optional[Measure]`
- `get_definition(self, hazard_type: Type[Hazard]) -> ScoreBasedRiskMeasureDefinition`
- `supported_hazards(self) -> Set[type]`
- `aggregate_risk_measures(self, measures: Dict[MeasureKey, Measure], assets: Sequence[Asset], prosp_scens: Sequence[str], years: Sequence[int]) -> Dict[MeasureKey, Measure]`

2.2.3 `RiskMeasuresFactory` (Protocol)

Protocol for factories that create risk measure calculators.

Methods:

- `calculators(self, use_case_id: str) -> Dict[Type[Asset], RiskMeasureCalculator]`

2.2.4 `AssetLevelRiskModel` (Inherits from RiskModel)

Implements risk calculation at the asset level.

Methods:

- `__init__(self, hazard_model: HazardModel, vulnerability_models: VulnerabilityModels, measure_calculators: Dict[type, RiskMeasureCalculator])`
- `calculate_impacts(self, assets: Sequence[Asset], prosp_scens: Sequence[str], years: Sequence[int])`

- `populate_measure_definitions(self, assets: Sequence[Asset]) -> Tuple[Dict[Type[Hazard], List[str]], Dict[ScoreBasedRiskMeasureDefinition, str]]`
- `calculate_risk_measures(self, assets: Sequence[Asset], prosp_scens: Sequence[str], years: Sequence[int])`

2.3 Key Functions

- `calculate_impacts`: Calculates impacts for given assets, scenarios, and years.
- `populate_measure_definitions`: Populates measure definitions for given assets.
- `calculate_risk_measures`: Calculates risk measures for given assets, scenarios, and years.

3 Detailed Class Descriptions

3.1 RiskModel

Base class for risk models, providing common functionality for impact calculation.

3.1.1 Key Methods:

- `_calculate_all_impacts`: Calculates impacts for all scenarios and years using concurrent execution.
- `_calculate_single_impact`: Calculates impacts for a single scenario and year.

3.2 AssetLevelRiskModel

Implements asset-level risk calculation.

3.2.1 Key Features:

- Uses concurrent execution for impact calculations.
- Supports multiple hazard types, scenarios, and years.
- Aggregates risk measures across different calculators.

3.2.2 Key Methods:

- `calculate_impacts`: Calculates impacts for given assets, scenarios, and years.
- `populate_measure_definitions`: Generates measure definitions and IDs for hazards and assets.
- `calculate_risk_measures`: Calculates and aggregates risk measures for assets.

4 Usage Example

```
hazard_model = SomeHazardModel()
vulnerability_models = SomeVulnerabilityModels()
measure_calculators = {AssetType: SomeMeasureCalculator()}

risk_model = AssetLevelRiskModel(hazard_model, vulnerability_models, measure_calculators)

assets = [Asset1, Asset2, ...]
scenarios = ["RCP4.5", "RCP8.5"]
years = [2030, 2050]

impacts, measures = risk_model.calculate_risk_measures(assets, scenarios, years)
```

5 Notes

1. The module uses concurrent execution for performance optimization in impact calculations.
2. Risk measures are calculated based on both historical and future impacts.
3. The model supports aggregation of risk measures across different hazards and assets.
4. Error handling is implemented in the concurrent execution part, but could be expanded.

6 Potential Improvements

1. Implement more robust error handling and logging.
2. Add more documentation strings to methods for better inline documentation.
3. Consider implementing caching mechanisms for frequently accessed data.

4. Optimize the aggregation process in `calculate_risk_measures` method.
5. Implement unit tests to ensure the reliability of the risk calculations.