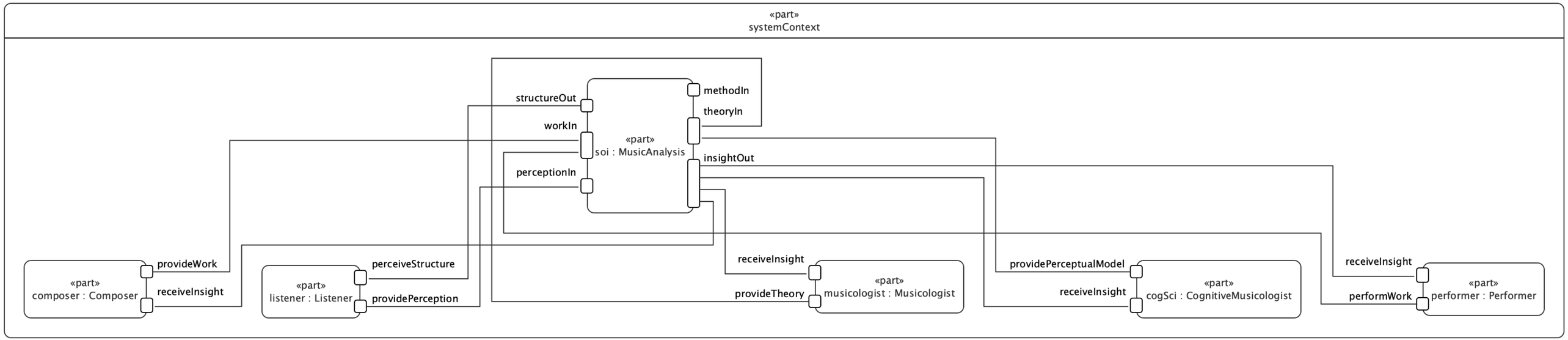
System Model for Music Analysis

The overall goal is to formalize music analysis as a system of interacting conceptual and computational processes that reveal latent musical structures—patterns of symmetry, directionality, organicity, and fractality underlying melodic behavior.

By articulating these structures and their relationships to perception and theory, the model seeks to:

- Bridge musicology and cognitive music science through shared representations.
- Enable performers, listeners, and researchers to experience and interpret the same latent forms.
- Demonstrate that music analysis can generate living musical experiences rather than static descriptions.

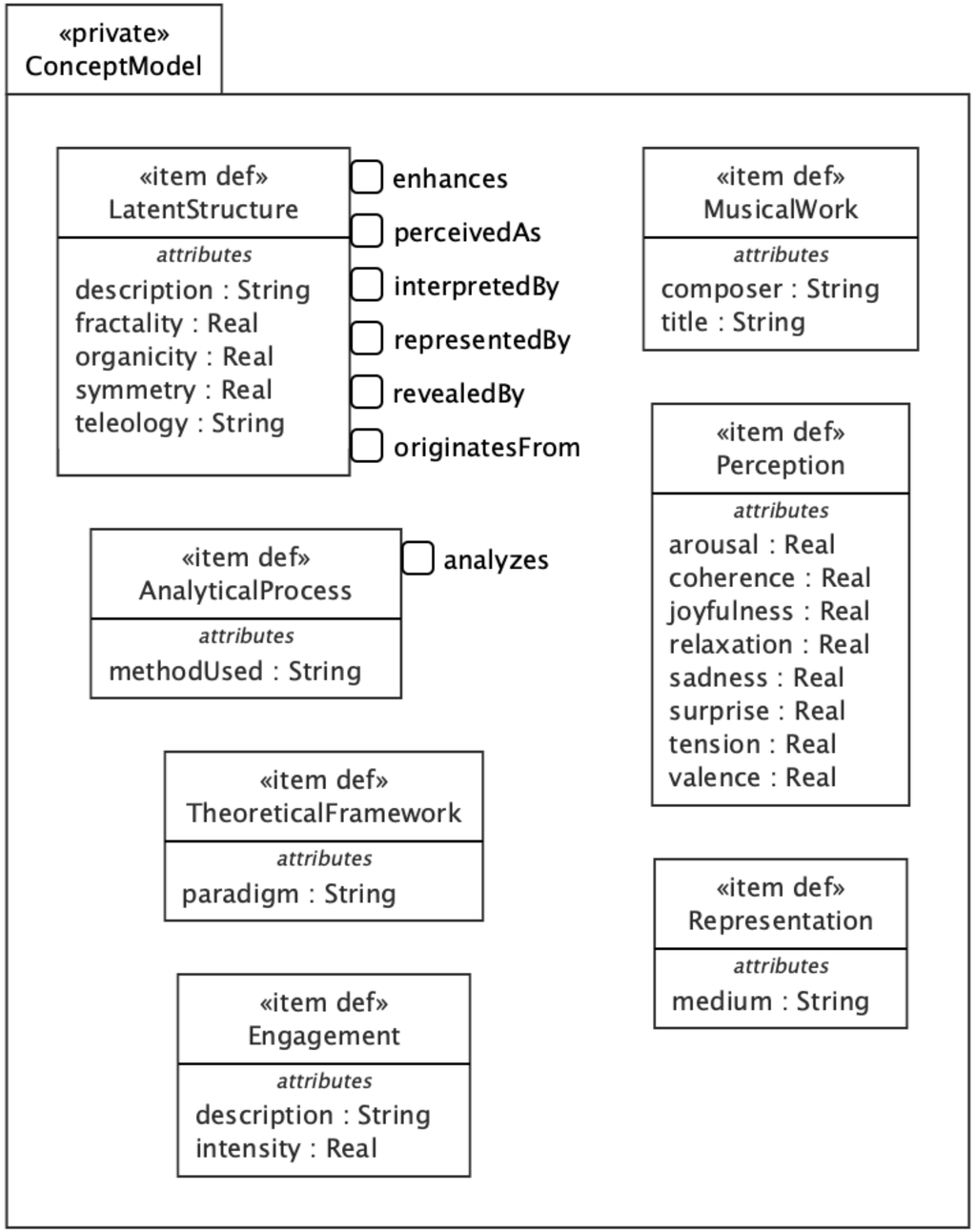
SOI:Music Analysis - Context Model



- Identifies stakeholders—**Composer**, **Performer**, **Listener**, **Musicologist**, **CognitiveMusicologist**—and the System of Interest **MusicAnalysis**.

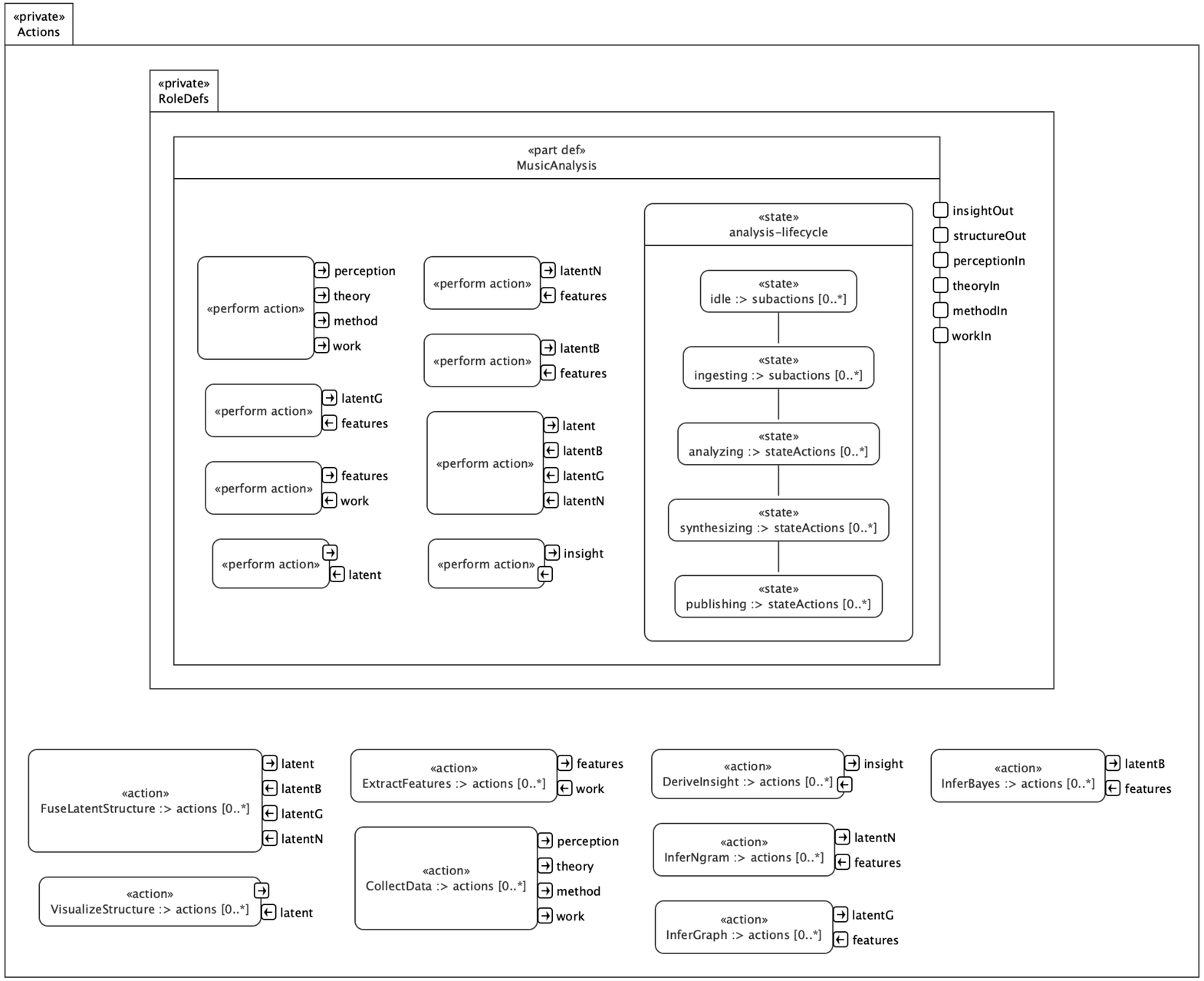
- Describes how information (works, insights, representations) flows among these roles.

SOI:Music Analysis - Concept Model



- Defines key entities: **LatentStructure**, **MusicalWork**, **AnalyticalProcess**, **Representation**, **TheoreticalFramework**, **Perception**, and **Engagement**.
- Captures semantic relationships such as **LatentStructure is revealed by AnalyticalProcess** and **enhances Engagement**.
- Serves as the philosophical map connecting musical meaning to perceptual and analytical activity.

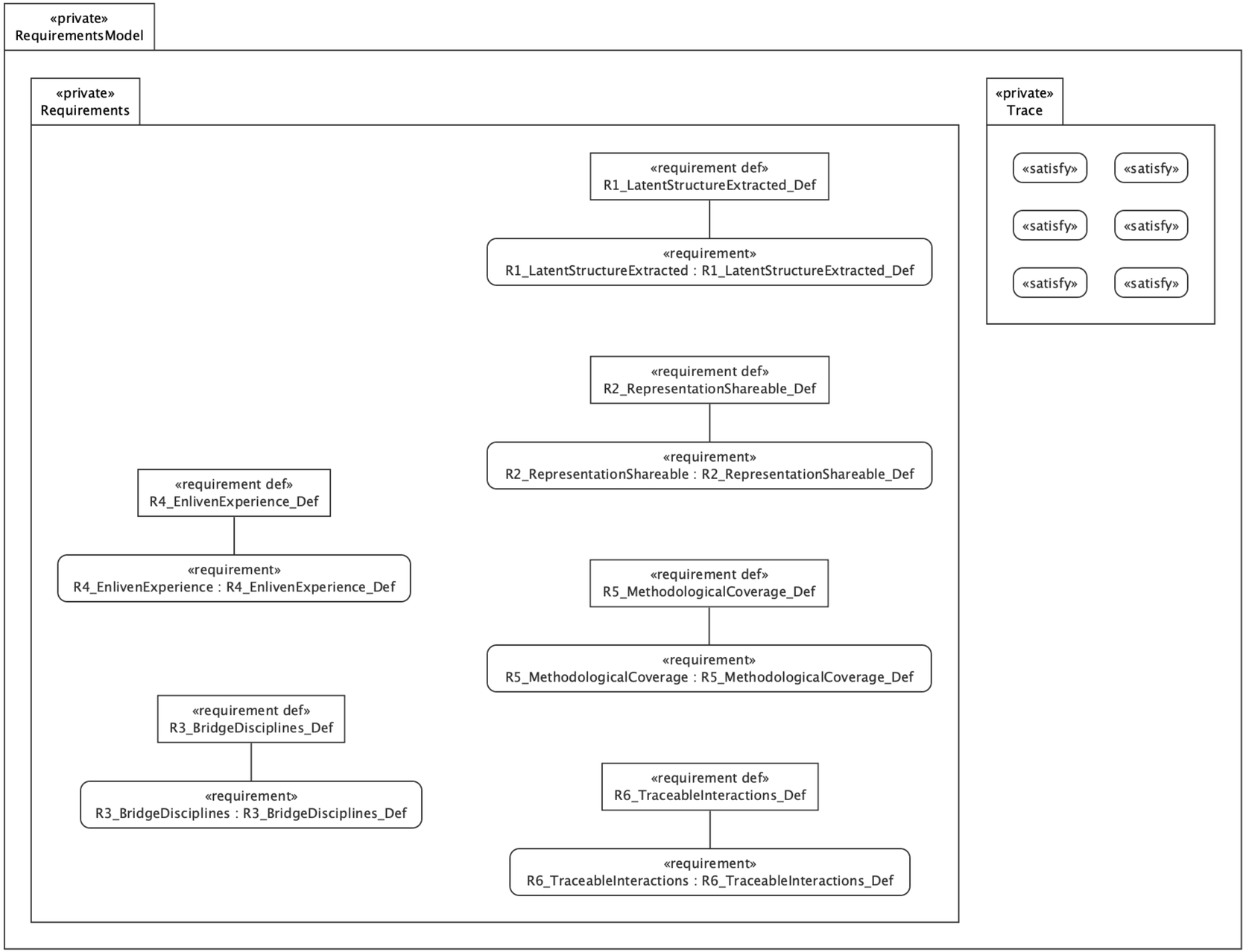
SOI:Music Analysis - Behavior Model



- Models the analytical workflow:
  Collect Data → Extract Features → Infer (N-gram, Graph, Bayesian) → Fuse Latent Structures → Visualize → Derive Insight.

- States lifecycle phases (**idle → ingesting → analyzing → synthesizing → publishing**) showing how the system evolves during analysis.

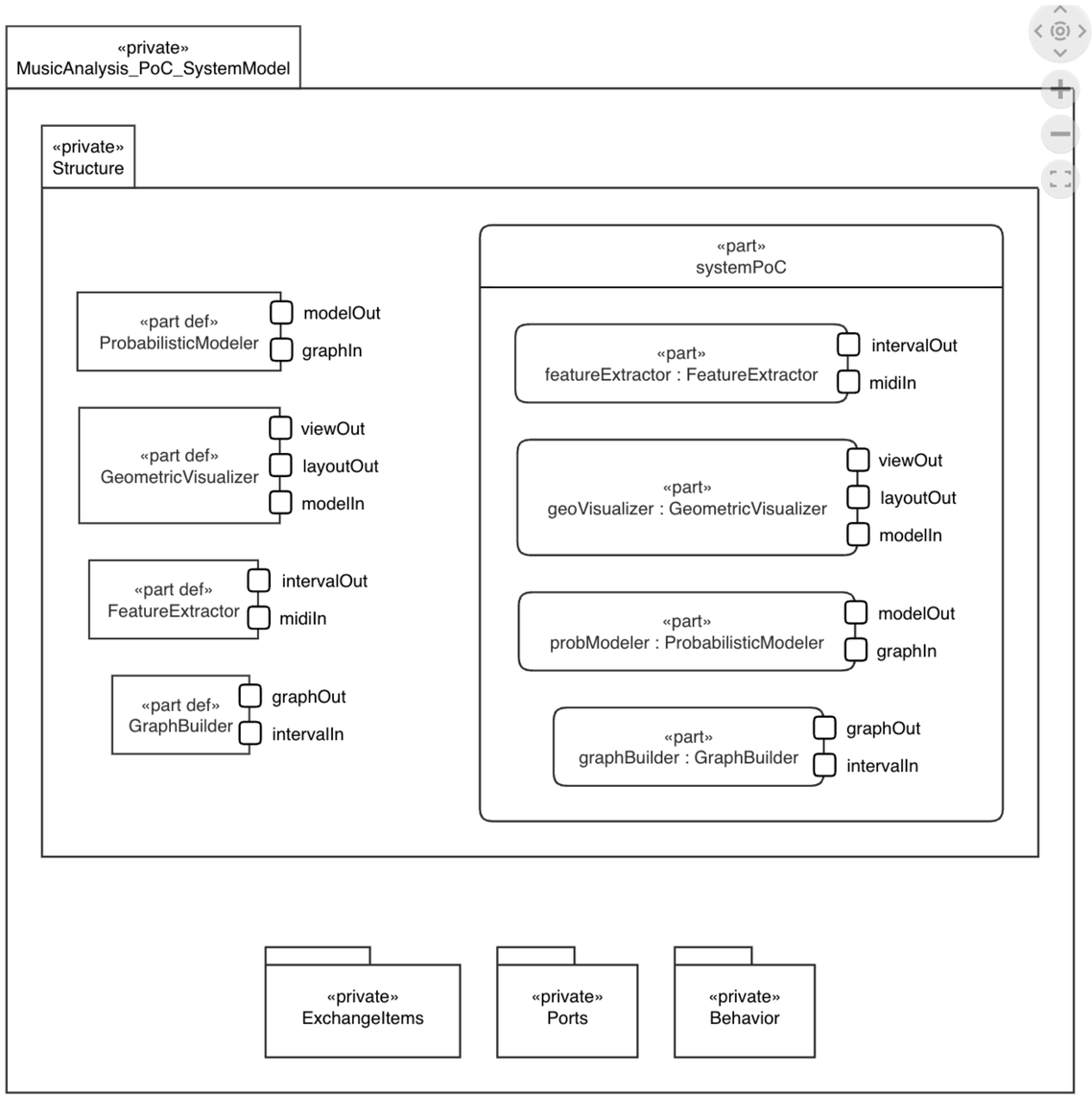
SOI:Music Analysis - Requirement Model



Expresses verifiable objectives:

- **R1 LatentStructureExtracted** – the system can detect latent structure.
- **R2 RepresentationShareable** – results can be shared and understood across roles.
- **R3 BridgeDisciplines** – connects theoretical and perceptual domains.
- **R4 EnlivenExperience** – enhances engagement through analysis.
- **R5 MethodologicalCoverage** – integrates N-gram, graph, Bayesian approaches.
- **R6 TraceableInteractions** – maintains observable information flow among stakeholders.

SOI:Music Analysis - PoC System Structure Model



PoC Implements a small-scale, executable prototype to verify feasibility of the concept.

Focuses on melodic interval (pitch-difference) analysis to demonstrate latent-structure extraction and Tonnetz-like visualization.Expresses verifiable objectives:

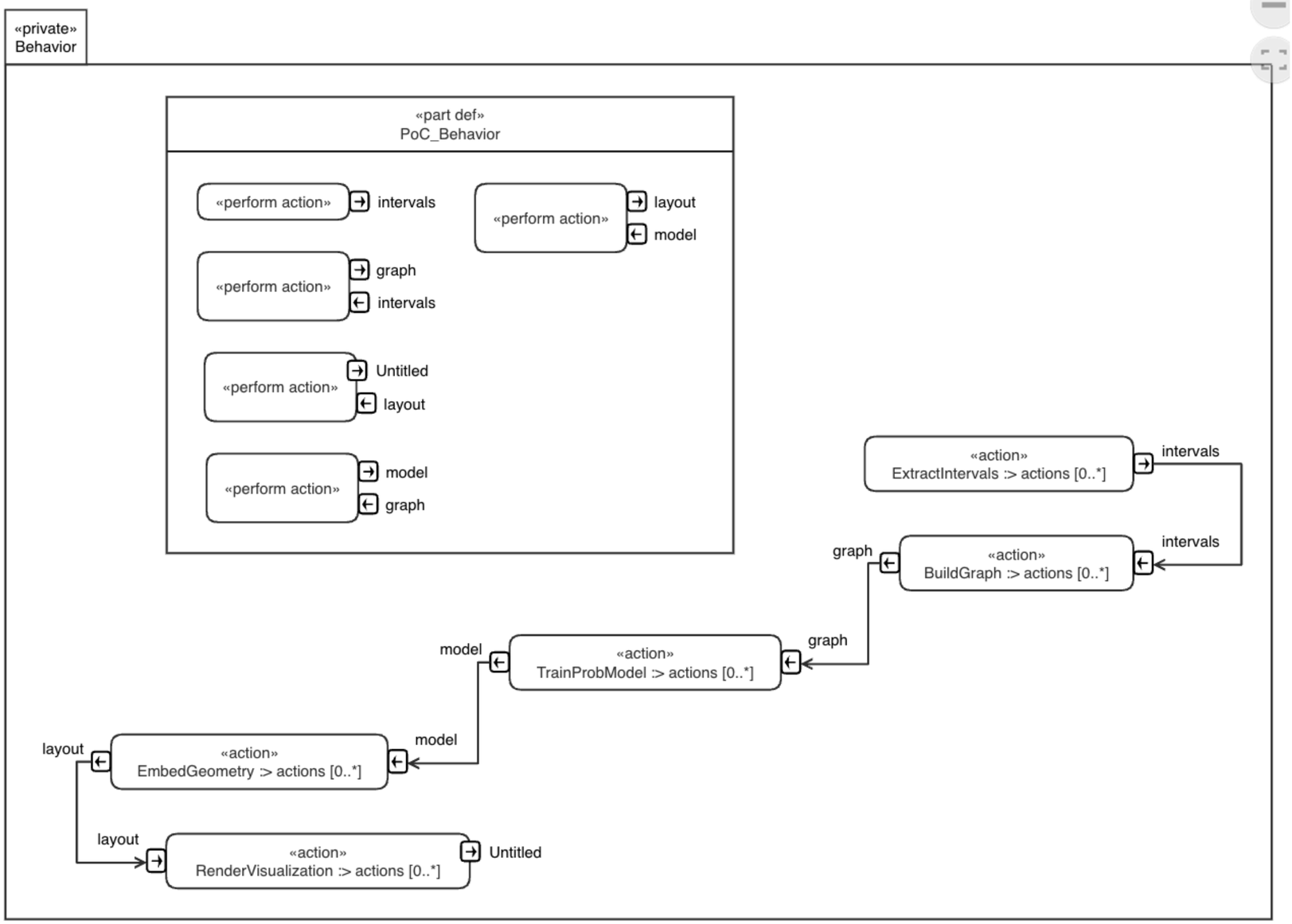
Feature Extractor : Extract melodic intervals (Δpitch N-grams)

Graph Builder : Construct interval-transition graph

Probablistic Modeler : Train HMM / Bayesian hybrid model

Geometric Visualizer : Compute Tonnetz-like embedding (MDS / UMAP

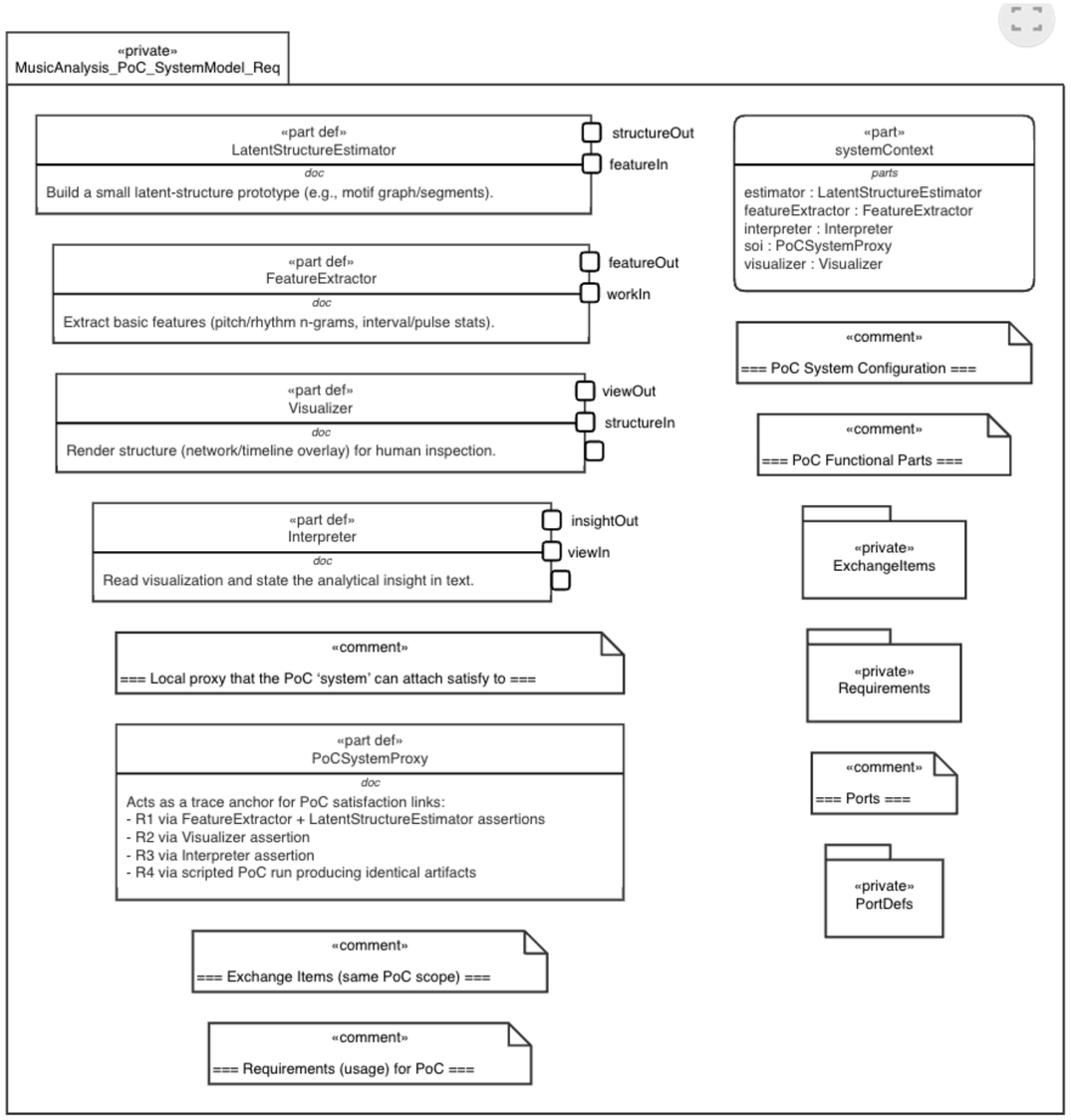
SOI:Music Analysis - PoC System Behavior Model




```
action ExtractIntervals {
    //  "Step 1 — Extract melodic intervals (Δpitch N-grams).";
    out intervals : ExchangeItems::PitchIntervalSeries;
};
action BuildGraph {
    //  "Step 2 — Construct interval-transition graph.";
    in  intervals : ExchangeItems::PitchIntervalSeries;
    out graph     : ExchangeItems::GraphData;
};
action TrainProbModel {
    // "Step 3 — Train HMM / Bayesian hybrid model.";
    in  graph : ExchangeItems::GraphData;
    out model : ExchangeItems::ProbabilisticModel;
};
action EmbedGeometry {
    // "Step 4 — Compute Tonnetz-like embedding (MDS / UMAP).";
    in  model  : ExchangeItems::ProbabilisticModel;
    out layout : ExchangeItems::LatentCoordinates;
};
action RenderVisualization {
    // "Step 5 — Render 2D visualization and output to viewer.";
    in  layout : ExchangeItems::LatentCoordinates;
    out view   : ExchangeItems::Visualization;
};
// /* — Control Sequence (for concept retention) — */;
succession flow s1 from ExtractIntervals to BuildGraph;
succession flow s2 from BuildGraph to TrainProbModel;
succession flow s3 from TrainProbModel to EmbedGeometry;
succession flow s4 from EmbedGeometry to RenderVisualization;

// /* — Dataflow (for drawing)  — */;
flow F1_Intervals
    from ExtractIntervals.intervals
    to   BuildGraph.intervals;
flow F2_Graph
    from BuildGraph.graph
    to   TrainProbModel.graph;
flow F3_Model
    from TrainProbModel.model
    to   EmbedGeometry.model;
flow F4_Layout
    from EmbedGeometry.layout
    to   RenderVisualization.layout;
```

SOI:Music Analysis - PoC System Requirement Model



Define Success Criteria for the PoC :

R1 PatternDetected The PoC shall detect at least one non-trivial latent pattern (e.g., repeated n-gram or graph motif).

R2 ViewGenerated The PoC shall produce a human-inspectable visualization of the detected structure.

R3 InsightRecorded The PoC shall output an insight derived from the visualization

R4 Reproducible Running the PoC twice on the same input shall yield identical outputs (features/structure/view/insight).