oaisq.org

The Open Autonomous Intelligence Initiative

QUESTION

How do we make AuI inspectable, interoperable, and accountable?

ANSWER

Create object-oriented models of AuI as explicit operational entities.

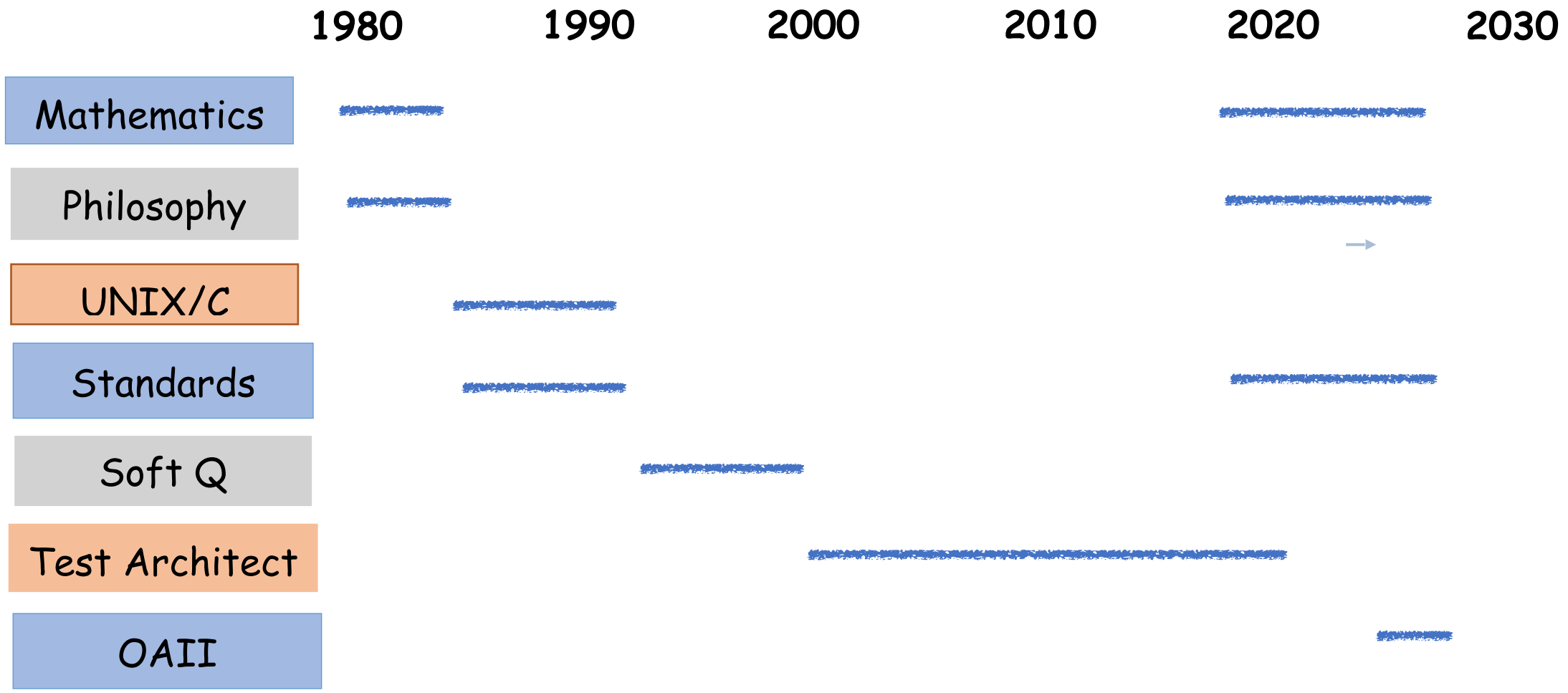
EXAMPLE

Model: Structural Intelligence Framework
Conformant Application: Edge-primary Personal Event Recognition (PER)

Read-ahead deck for AI / Data Science discussion

Working Draft Version 061926

BACKGROUND



I am qualified to be an advocate for open object-oriented standards.

CONCEPTUAL EVOLUTION

oaisq.org

OAII BASE MODEL

Entity, Device, Sensor, Signal, Knowledge, Agent, Log, Interface

UNITY POLARITY AXIOMS (UPA)

Philosophy, Dualistic Monism Axioms, Theorems, Worlds

AXIOMS OF INTELLIGIBILITY AND MIND (AIM)

Philosophy of Mind
Phenomenal-Psychological
Polarity, Covariance
Clinical Psychology

POLARITY MODELING FRAMEWORK (PMF)

Concept Papers:
Knowledge, Process, Context, Transformation, Regulation, Worlds

First Pivot: UPA to AIM

Clarified philosophical scope, De-emphasize theorems and geometry

Second Pivot: AIM to PMF

Removed ontological scope
Focus on Cognitive Science & Clinical Psychology

Current (Last?) Pivot

PMF + OAII → Structural Intelligence Framework
Example-Focused
Location before World

200+ posts published at oaisq.org

THE PROBLEM

oaisq.org

The internal structures of AI are often too opaque, vendor-specific, or task-specific to support public trust, certification, or interoperability.

EXPLAINABILITY

Evidence, assumptions, alternatives, or policy constraints are not inspectable.

VENDOR LOCK-IN

Sensors, models, logs, agents, policies, and interfaces typically evolve as vendor-specific processes, hindering interoperability.

GOVERNANCE

Increasingly recognized in policy, legal, and ethics communities. Without standard objects and logs, it is hard to know what a system is doing or why.

This is a modeling, architecture, QA, standards, deployment and governance issue.

Accountable AI needs certifiable open, standard, shared structures.

OAII ADVOCACY

oaisq.org

OAII is advocating for a way that AuI systems should be modeled: open enough to inspect, standard enough to interoperate, object-oriented enough to implement, and accountable enough to trust.

OPEN

Inspectable structures, Explicit assumptions, Transparent reasoning Reviewable logs.

STANDARD

Interoperable models, common interfaces, certification-friendly testing.

OBJECT-ORIENTED

Implementable entities: device, sensor, signal, knowledge, interface, agent, transformation, log

ACCOUNTABLE

Policy-governed, Explicit constraints Traceable decisions Auditable behavior

An example object model + conforming application are needed to demonstrate conceptual and technical feasible.

EXAMPLE AuI MODEL

oaisq.org

The Structural Intelligence Framework models AuI in terms of structured relationships

STRUCTURE

Behavior expressed through organized Entity Locations and relationships.

CONTEXT

Signals have Entity Location specific meaning

KNOWLEDGE

Stable contextual occurrences of signal patterns learned over time.

TRANSFORMATION

Modeled as changes in state, relation, or condition.

REGULATION

Policies and constraints guide interpretation, action, and safe behavior.

TEMPORAL PROJECTION

Time is handled as an interpretive projection over structured change.

EDGE INTELLIGENCE

Inference and response remain as local and privacy-aware as practical.

These are not pipeline stages.
They are primary, co-present structural aspects of a SIF-compliant system.

SIF APPLICATION - OAII ARCHITECTURE

oaisq.org

SIF Application

Regulation

- Transformation
- Event
- Context
- Temporal Projection
- Report
- Log

OAII

Entity

Device

- ESP32
- Pi Zero
- Pi 5

Sensor

- BLE
- USB
- WiFi
- mmWave

Signal

- Presence
- Image
- Vibration
- Motion

Knowledge

- Location
- Policy
- Routine
- Event

Agent

- Event
- Learning
- Recognition
- Alert

Interface

- API
- Voice
- Text
- Display

Log Level

- Information
- Warning
- Error
- Debug

COMPREHENSIVE EXAMPLE

oaisq.org

NOT A MINIMUM VIABLE PRODUCT

A worked example that can demonstrate structural, accountable AuI.

NOT NECESSARILY MINIMAL

Ambitious enough to show the value of open object-oriented models.

CONCEPT → FORMAL SPECIFICATIONS → EXAMPLE → OPEN SOURCE

CREDIBILITY BASE

The work draws on development, systems engineering, QA, network management, protocols, standards, and LLM-assisted conceptualization.

WHAT SUPPORT ADDS

Grassroots reviewers and builders can challenge assumptions, test interoperability, and make the public-benefit path plausible.

A rigorous example grounded in a published open source framework.

PERSONAL EVENT RECOGNITION (PER)

oaisq.org

PER is a proposed AuI system approach in which sensing, interpretation, and first-line response occur primarily on local edge devices. It recognizes meaningful personal events from signals in context while keeping human-relevant decisions inspectable and policy-governed.

FEASIBLE NOW

Affordable high-performance edge devices + small learning data sets enable local processing.

OPEN-SOURCE MODELS

Speech/audio, vision, signal processing, and lightweight inference components are widely available.

PRIVACY

Local processing reduces cloud exposure. Consent-aware policies control sensitive household data.

INTEROPERABILITY

Shared classes and interfaces can integrate devices and software from multiple vendors.

PER has product potential — it can support aging-in-place and related personal assistance use cases with timely, context-aware responses. For now it serves as an example of an AuI based on the Structural Intelligence Framework.

EXAMPLE USE CASE

oaisq.org

The Primary User begins a normal morning routine: bed/sleep area → bathroom → kitchen → medication/ coffee/breakfast. While showering, a phone call, cell phone alert, or front door entry occurs, kitchen water is detected, shower water is constant, and the shower time alert threshold is reached.

BATHROOM START

UNEXPECTED SIGNAL(S)

BATHROOM TIMEOUT

HYPOTHESIS

REGULATED ACTION

POSSIBLE OUTCOMES

Non-primary user entry, Routine resumes normally, Routine is aborted, Potential distress

WHY THIS IS HARD

Another person, Interruption changes expected durations, Temporal order is not meaning

LOG

Signals, locations, time. Event and Policy references, Agent decisions and actions

The structure of change cannot be represented based on completion time or time order.

SUMMARY AND NEXT STEPS

oaisq.org

SUMMARY

Problems: Vendor Lock-in, Transparency, Accountability, Privacy, Existential?
Solution: Open, Standard, Object-oriented models
Qualifications: Standards, Development, Quality Assurance
Example: Edge-primary PER for aging in place.

NEXT STEPS

Comments: Is this conceptually and technically feasible?
Collaboration: Design, specify, develop, test, support, social media, marketing
Specifications: OAII base classes and PER subclasses
Presentations: OAII Base classes and PER subclasses
Pseudo code
Agile iterations
GitHub

APPENDIX A: OAII BASE CLASSES

oaisq.org

ENTITY

oaisq.org

The Entity class serves as the foundational base class. It represents the abstract concept of "things" that can exist and have properties, behaviors, and relationships.

Attributes: ID, Name, State, Status, Location

Methods: get_attribute, set_attribute, to_string, initialize, activate, suspend, terminate

Notifications: entityCreated, entityDeleted, entityDiscovered, entityStateChange, entityConfigUpdate, entityStatusChange, entityStateChange, entityAlarmRaised, entityAlarmCleared

As a base class, Entity provides a generic blueprint for all entities in the system. Its primary responsibility is to ensure that derived classes maintain consistent structure and behavior. Subclasses will further refine and add to these attributes based on their specific roles.

HOST

oaisq.org

A Host is a physical, virtual, or logical OAII Entity that provides a managed location for Sensors, Knowledge, Agents, Interfaces, Logs, or autonomous-intelligence processing.

Sensors

provide power
- provide GPIO/I2C/SPI/UART/USB

Personal Computer

\