An Exploration into Scientific Narrative Of WorkFLow for Learning, Analytics and Knowledge Engineering

Javed I. Khan[1] and Philip Thomas[2]
e-mail: Javed@ kent.edu | plthomas@ kent.edu

[1] Internetworking and Media Communications Research Laboratories
Department of Computer Science

[2] Information Systems Division
Kent State University
233 MSB, Kent, OH 44242
March 2025

1. Introduction

Modern scientific projects are increasingly collaborative, data-intensive, and computationally distributed. A single study may span laboratories, instruments, software environments, and analytical frameworks across continents. Yet, despite this interconnected nature of contemporary science, there remains no universal, machine-readable narration language to describe *how* scientific work actually happens — how people, computation, and instruments cooperate to produce reproducible knowledge.

SNOWFLAKE — *Scientific Narrative Of WorkFLow for Learning, Analytics and Knowledge Engineering*— is proposed as that language. SNOWFLAKE is a structured descriptive and executable language for representing, analyzing, sharing, and even operationalize the complex lifecycle of a scientific project — encompassing human, computational, machine, and data components. SNOWFLAKE provides a structured, declarative way to document and encode every element of a scientific workflow:

- **Human Work Elements** (researchers, analysts, field scientists)
- **Computational Work Elements** (software, models, algorithms)
- **Machine Work Elements** (instruments, sensors, hardware)
- **Information Product Elements** (data, models, results, and scientific thoughts)
- **Linker Elements** (that coherently connect them for many purposes).

Each workflow along with its elements described in SNOWFLAKE becomes a *scientific object* — uniquely identifiable, interoperable, and semantically interpretable. The language defines not only the *structure* of a workflow but also the *conditions*, *resources*, *roles*, and *cdependencies* that give it operational meaning. SNOWFLAKE therefore acts as both a **language** and a **framework** — bridging human understanding with computational execution. Just as a snowflake crystallizes unique yet symmetrical structure, each SNOWFLAKE document captures a scientific process in all its individuality while preserving universal form. In short, **SNOWFLAKE** transforms scientific workflows from

informal narratives into **structured, computable knowledge** — enabling science to describe itself.

2. Related Work

Scientific workflow description languages (SWDL) have evolved over the past decade to enhance computational reproducibility, data provenance, and cross-platform interoperability in research automation. Indeed, in last decade, there has been amazing progress in attempt to capture science workflow. These include Common Workflow Language (CWL) (Amstutz *et al.*, 2016), Workflow Description Language (WDL) (Voss *et al.*, 2017), Nextflow (Di Tommaso *et al.*, 2017), Pegasus (Deelman *et al.*, 2015), Galaxy (Afgan *et al.*, 2018), RO-Crate (Soiland-Reyes *et al.*, 2022), and the recently proposed SWIRL interoperability framework (Goble *et al.*, 2024).

Each system was developed for a specific scientific purpose and has since evolved within distinct ecosystem contexts. CWL was designed as a *platform-neutral standard* for workflow description and has become central to FAIR-data infrastructures such as ELIXIR, WorkflowHub, and EOSC-Life (Goble *et al.*, 2020; Sufi *et al.*, 2023). WDL, developed by the Broad Institute, underpins biomedical analysis platforms like Terra, BioData Catalyst, and Gen3 Commons, where standardized genomics workflows are executed at national scale (Broad Institute, 2024). Nextflow has become one of the most widely adopted frameworks for *reproducible, containerized, and scalable data-intensive workflows*, driven by the nf-core community and integrations with AWS Batch, Google Cloud Life Sciences, and Nextflow Tower (Di Tommaso *et al.*, 2017; Ewels *et al.*, 2020). Pegasus serves as a mature engine for *high-performance and distributed computing*, supporting production deployments at LIGO, NSF XSEDE, and DOE facilities, emphasizing workflow mapping, provenance, and fault-tolerant execution (Deelman *et al.*, 2015; Vahi *et al.*, 2020). Galaxy provides a *graphical, user-friendly environment* for accessible and reproducible data analysis, widely deployed through public portals such as UseGalaxy.org, UseGalaxy.eu, and NIH Galaxy, where it supports large communities of life scientists and educators (Afgan *et al.*, 2018; Jalili *et al.*, 2020). RO-Crate complements these systems by providing a *metadata-packaging standard* that encapsulates digital research artifacts for FAIR compliance and interoperability across repositories like Zenodo, DataCite, and ARDC (Soiland-Reyes *et al.*, 2022). Finally, SWIRL represents an emerging *interoperability-oriented representation language* designed to unify workflow ecosystems through a shared intermediate scientific abstraction (Goble *et al.*, 2024).

These languages excel in computational automation, orchestration, reproducibility however, these exhibit limited semantic expressiveness and epistemic transparency.

Declarative standards such as CWL and WDL are now cornerstones of reproducible research but remain primarily machine- and data-focused, offering few constructs for representing human or institutional roles. Nextflow and Pegasus deliver *high execution fidelity* across distributed environments but provide minimal scaffolding for ethical or Open Science narratives. Galaxy democratizes access through graphical interfaces but sacrifices formal extensibility. Metadata-focused models like RO-Crate and SWIRL enhance interoperability but stop short of capturing human, ethical, or interdisciplinary contexts.

AS evident, most existing SWDL frameworks remain centered on machine-executable and data-centric representations, leaving gaps in modeling the science itself- the story of human participation, scientific rationale, ethical context, and interdisciplinary innovation narratives. —dimensions that the proposed SNOWFLAKE framework seeks to unify within a single, semantically rich schema.

In contrast to these SWDL frameworks- as we will see SNOWFLAKE extends beyond machine/computation/digital focused execution description to integrate computational, human, and semantic dimensions of scientific workflows—enabling richer representation of *intent, responsibility, collaboration, and reproducibility* across the full lifecycle of research.

3. Design of SNOWFLAKE

The **SNOWFLAKE schema** (Scientific Narrative Of WorkFLow for Learning, Analytics and Knowledge Engineering) is designed as a *meta-representation framework* for describing every dimension of a research project — not just its computations, but also the **human, institutional, informational, and ethical contexts** that make science understandable, communicable, reproducible and interpretable. A scientific workflow is not just a series of computing- it must be understood in the context of scientific knowledge it adds to, and the larger collaborative human knowledge processes involved.

3.1. Formal Definition and Distinction of the SNOWFLAKE

Classically, a workflow has been defined as a "*formal specification of a process that determines the ordered sequence of computational or data manipulation tasks, their dependencies, and the flow of control and information among them*" (Taylor et al., 2007; van der Aalst et al., 2003; Deelman et al., 2005). Such definitions emphasize **process automation**, **task coordination**, and **computational reproducibility**, serving primarily the needs of **machine-oriented scientific workflows** such as those implemented in CWL, WDL, Pegasus, or Galaxy (Gil et al., 2007; Curcin and Ghanem, 2008). Within this paradigm, workflows are treated as **execution graphs**—efficient but semantically shallow structures that capture

how computation occurs while completely omitting human factors, intentions and important information such as why, by whom, and under what scientific or ethical rationale it is performed.

In contrast, the **SNOWFLAKE Workflow**—*formally defined as a structured and semantically unified schema that integrates human, computational, machine, and informational processes into a single representation of scientific intent, execution, and accountability*—extends the classical concept beyond automation. It reconceptualizes workflows not merely as process control models but as **knowledge representation frameworks** that capture the epistemic, organizational, and ethical context of scientific research project.

3.2. Architecture of SNOWFLAKE

Each SNOWFLAKE workflow is instantiated as a graph of interlinked object classes—**WorkFlow (W)**, **HumanWorkElement (H)**, **ComputationalWorkElement (C)**, **MachineWorkElement (M)**, **InformationProductElement (I)**, and **Link (L)**—providing a **multi-actor ontology** that models the full lifecycle of a scientific process. Unlike traditional workflow description languages that focus on execution provenance and data lineage, SNOWFLAKE introduces attributes for research hypothesis, theoretical framework, innovation, FAIR indices, and ethical-environmental review, thereby achieving both **computational reproducibility** and **scientific interpretability**.

In formal terms, while the classical workflow may be represented as a directed acyclic graph $G = (T, E)$, where T denotes tasks and E the control or data dependencies, the SNOWFLAKE workflow extends this to a heterogeneous, typed graph

$$G_{SNOW} = (V, E, \Sigma_V, \Sigma_E)$$

where $\Sigma_V = \{W, H, C, M, I\}$ defines vertex classes (human, computational, machine, informational) and $\Sigma_E = \{L\}$ defines link semantics capturing execution, data-flow, accountability, and ethical relations. This model allows not only automation and reproducibility but also traceable understanding— enabling cross-domain reuse, meta-analysis, and Open Science compliance at the institutional level.

In SNOWFLAKE, a **Workflow** serves as the *top-level narrative container* that encapsulates:

- The **scientific purpose** (research question, hypothesis, and theoretical framework),

- The **sequence and interrelation** of human, computational, and machine tasks,
- The **information products** created, consumed, or transformed, and
- The **ethical, environmental, and institutional contexts** surrounding the research.

A Workflow thus provides both **procedural structure** (the order and dependency of actions) and **epistemic context** (the intent, innovation, and accountability) — bridging the gap between *automation schema* and *scientific narrative*.

4. Elements of SNOWFLAKE

4.1. WorkFlow (W) — Project-Level Schema

The WorkFlow (W) element represents the *root scientific process*—a complete research activity or experiment, encompassing its intent, methodology, execution, and evaluation. It functions as the meta-node that unifies all other process components (human, computational, machine, informational, and linking). Each workflow instance encapsulates descriptive attributes such as its scientific abstract, research question, theoretical framework, innovation vector, ethical and environmental reviews, data stewardship, and FAIR/reproducibility indices. Formally, $W = \langle ID, D, H, C, M, I, L \rangle$, where these components define the composite structure of a self-describing, semantically linked scientific workflow.

4.2. HumanWorkElement (H) — Human Task or Cognitive Role

The HumanWorkElement (H) models *human participation and accountability* within a workflow. It captures the intellectual, manual, or supervisory actions performed by individuals or teams—such as designing protocols, verifying outputs, annotating data, or approving results. Unlike traditional workflows that abstract human input as opaque or external, SNOWFLAKE explicitly encodes each actor's role, competency level, accountability entity, task time, tools, environment, and verification method. Formally:

$$H_i = \langle Role, Action, Context, Time, Competency, Verification \rangle$$

Where every human instance contributes both process and epistemic traceability to the workflow.

4.3. ComputationalWorkElement (C) — Algorithmic or Software Process

The ComputationalWorkElement (C) represents any *algorithmic, code-based, or software-driven operation*—from data preprocessing to machine learning training or simulation. It defines the computational function, parameters, algorithms,

resource usage, and reproducibility metadata (e.g., container versions, runtime environment, code availability). Attributes such as AlgorithmicDescription, ComputeResource, ExecutionTime, SoftwareMaturity, and VerificationMethod ensure both performance accountability and code-level reproducibility. Formally:

$$C_j = \langle Function, Parameters, Resources, Environment, Verification, FAIR_C \rangle$$

Extends classical machine-task nodes into self-describing, FAIR-aware computational entities.

4.4. MachineWorkElement (M) — Physical or Instrumental Operation

The MachineWorkElement (M) describes *hardware-dependent or physical system operations*—for instance, a GPU cluster performing inference, a lab instrument capturing spectra, or an IoT node collecting sensor data. Each instance models the instrument type, capacity, location, operating environment, calibration, consumables, responsible entity, and machine-level FAIR index (M-FAIR). Formally:

$$M_k = \langle Device, Capacity, Environment, Calibration, Accountability, FAIR_M \rangle,$$

It ensures that physical instrumentation is not only operationally, but also semantically integrated into the reproducibility framework.

4.5. InformationProductElement (I) — Data or Knowledge Artifact

The InformationProductElement (I) captures the *data entities, models, or knowledge artifacts* produced, consumed, or transformed during workflow execution. Each instance may represent raw datasets, derived models, simulation outputs, documents, or analytical results. Attributes such as SchematicDescription, FileSize, CurationExpertise, ValidationMethod, and I-FAIR Index ensure that data are findable, accessible, interoperable, and reusable. Formally,

$$I_l = \langle Content, Schema, Size, Validation, Custodian, FAIR_I \rangle,$$

It make every data element a traceable and reusable unit of scientific evidence.

4.6. Link (L) — Relational and Causal Connector

The Link (L) element formalizes the relationships and dependencies among all other object classes in SNOWFLAKE. Links define not just execution order (as in classical DAGs) but also semantic, ethical, and accountability relations between workflow components—such as "produces," "verifies," "executes-on," "resides-on," or "supervises." Each link is typed and labeled ($LinkType, LinkLabel$) and

connects two instances $(src, sink)$, maintaining metadata on cardinality, validation, and ownership. Formally,

$$L_m = \langle src, sink, type, label, verification \rangle,$$

It provides a relational fabric that enables multi-level reasoning, provenance tracing, and process validation.

4.7. Attributes and Semantics Perimeter

We have designed an extensive +150 attribute schema which works as a semantic container defining these five elements of SNOWFLAKE. The document presents the structural design and semantic container – as defined by the **attribute specifications** of the SNOWFLAKE schema, defining each entity and its associated descriptors in a **machine-interpretable format**. Reader can examine the list of attributed, their semantics, and the encoding methodology in the associated Technical Report (Khan, & Thomas & Prithula, 2025 [18]). Each attribute is assigned a unique identifier, structure, and value constraint to ensure consistency, interoperability, and semantic traceability across diverse research environments. It serves as a technical reference for **encoders, system designers, and developers** implementing SNOWFLAKE-compliant registries, workflow capture systems, or data-integration pipelines.

5. Features of SNOWFLAKE

5.1. Traditional Workflow Construction

A **flowchart** in the SNOWFLAKE system can be dynamically generated using the entity and link attributes already defined in Tables 3–8. Each node in the chart corresponds to an entity—**Human Work Element (HWE)**, **Computational Work Element (CWE)**, **Machine Work Element (MWE)**, or **Information Product Element (IPE)**—identified by its InstanceID, Title, and SNOWTypology. The flow relationships between these nodes are drawn from the **Link Elements** (SRCInstanceID, SinkInstanceID, LinkType, and optional LinkLabel), which serve as directed edges describing execution, data, or dependency paths. Logical branching, decisions, and synchronization points are governed by the EpistemicCondition fields on each node (104.3, 105.3, 108.3, 106.4) and by LinkEpistemicCondition (107.3) on edges; these conditions act as guards that enable or block transitions—precisely the role of decision diamonds in classical flowcharts. Additional contextual cues such as ActorRole, ResponsibleEntity, or FunctionalRole can be used to group nodes into **swimlanes**, while NestedCardinality and WorkFlowId organize sub-processes and parent flows. When rendered graphically, this metadata collectively expresses **process order**,

information flow, and **epistemic validation states**, allowing SNOWFLAKE to reproduce—and extend—the semantics of a traditional flowchart within a unified scientific workflow graph.

5.2. Control Flow Representation

SNOWFLAKE can represent control flow, and in fact it can do so in a **more expressive and epistemically rich way** than traditional programming control-flow diagrams.

Control flow describes **how execution proceeds** — what runs next, under what condition, and how loops or branches are handled. SNOWFLAKE already encodes these semantics through its **Processing Elements (HWE, CWE, MWE)** and **Link Elements (LNK)** with **Epistemic Conditions**.

In this sense:

- **Nodes (processes)** = execution blocks
- **Links (connections)** = control edges
- **EpistemicCondition / LinkEpistemicCondition** = guard or branch condition
- **Cardinality & Persistence** = loop, iteration, or concurrency semantics

Thus, a **control-flow graph (CFG)** is naturally **embedded inside** a SNOWFLAKE workflow as its *control subgraph.*

Table-8 How SNOWFLAKE Encodes Control Flow

Control Flow Concept	SNOWFLAKE Representation	Example
Sequential execution	LinkType="Execute-After" between two Processing Elements.	Task B runs after Task A completes.
Conditional branch (if/else)	Multiple outgoing links from a node, each with a different LinkEpistemicCondition.	"If sensor_temp > 50, activate cooler; else log warning."
Loop / iteration	A link from a downstream node back to an upstream node with LinkType="Loop-Back" and a condition.	"Repeat until error_rate < 0.05."
Parallelism / fork-join	One-to-many or many-to-one links using Cardinality or LinkType="Fork"/"Join".	"Run preprocessing on 5 data partitions concurrently."
Synchronization	A Join node whose inbound links must all be valid before execution (EpistemicCondition = all preconditions true).	"Start training only when all sensors calibrated."
Function / subroutine call	A nested WorkFlow (103.3 NestedCardinality) with its own internal flow.	"Invoke SubWorkflow: ModelEvaluation."
Exception / interrupt	A special link with LinkType="Interrupts" or conditional guard.	"Abort if calibration file missing."

Traditional control flow only expresses *temporal or logical* sequencing. SNOWFLAKE introduces **epistemic control flow** — control based on *knowledge state*. That means a process can be enabled not just when a flag or variable is true, but when an **information product (IPE)** has been validated, verified, or peer-reviewed. Such as "Run the publication workflow only after dataset IPE is validated and peer-review approval is true." This goes beyond classical CFGs — it merges **knowledge readiness** with **execution logic**, which is essential for scientific or data-driven workflows.

5.3. Virtual Computing Orchestration

SNOWFLAKE can also be used to represent the mapping of computational processes onto virtual or physical machine hardware by aligning each Computational Work Element's (ComputeResource, ID 105.10) specification with the Machine Work Element's (CapacityDescription, ID 108.9) descriptor through the Executes-on link. The ComputeResource field of a CWE defines the typical node or hardware configuration required for execution —for example "3× Xeon 28c + P100" —expressing the processor count, core size, and accelerator type the computation expects. Conversely, each MWE declares its *CapacityDescription*, such as "20× Xeon 28c + 160× P100," summarizing the actual compute resources available on that machine or virtual instance. During orchestration, the Executes-on links in *mappable* state are resolved by matching *ComputeResourc*e requirements against *CapacityDescription* availability, transitioning through requested → pending → granted → locked as scheduling proceeds. This pairing enables SNOWFLAKE to reason quantitatively about placement, verify resource sufficiency, and translate logical workflows into concrete, capacity-aware deployments across heterogeneous clusters or virtualized infrastructures.

5.4. Epistemic Conditions for IPE

A **powerful and novel feature** of the **SNOWFLAKE** framework is that **conditions** are not restricted to process transitions alone but can be attached to **Processing Elements**, **Information Product Elements (IPEs)**, and even the **Link Objects** that interconnect them. This represents a significant generalization of the classical **Petri net** paradigm—where conditions apply primarily to transitions—into a richer, multi-dimensional **epistemic network**. In SNOWFLAKE, every entity and relationship can embody its own **activation, validation, or dependency condition**, enabling the system to model not just *when processes occur*, but *when information becomes epistemically justified* and *when relationships themselves become valid*. This triadic conditioning mechanism allows SNOWFLAKE to unify **workflow logic, information provenance, and knowledge-state evolution** within

a single, semantically consistent framework—marking a conceptual advance in representing the dynamics of scientific knowledge creation.

Table-8 Interpretation of Epistemic Conditions in SNOWFLAKE

SNOWFLAKE Element	Petri Net Analogue	Condition Type	Interpretation in SNOWFLAKE	Example of Application / Trigger
Processing Element (PE)	**Transition**	**Operational Condition**	Defines when a *process* (human, machine, or computational) is permitted to execute. Conditions may include availability of input data, tool readiness, user authorization, or time constraints.	"Script (PE) A executes only if raw dataset exists, schema matches, and validation flag = true."
Information Product Element (IPE)	**Token / Place Content**	**Epistemic Condition**	Defines when an *information artifact* can be instantiated, updated, or considered epistemically valid. Reflects logical sufficiency, evidential support, or peer validation.	"Derived Genome dataset (IPE) G is "valid" only when calibration model and metadata records are verified."
Link Object (LNK)	**Arc or Flow Connection**	**Relational Condition**	Defines when a *relationship* (e.g., input–output linkage, dependency, or citation) can be asserted as valid. Ensures semantic and temporal coherence between connected elements.	"A 'usesDataFrom' link activates only after both source and target IPEs are validated and temporal overlap is confirmed."

5.5. Representation of Human Work and Role

In classical workflow languages such as CWL or WDL, human actions are invisible to the computational graph; data simply appears as input. RO-Crate and SWIRL add descriptive power—they can record that *Dr. Lee operated the microscope* or *a participant contributed blood samples*—but these remain annotations detached from execution. SNOWFLAKE, however, integrates people as **active epistemic agents**: the experimenter's calibration, the analyst's judgment, and the subject's consent are all encoded as *workflow attributes and conditions*. A SNOWFLAKE workflow may pause until an authorized scientist verifies an image, or refuse to proceed without documented consent. This elevates the workflow from a passive automation script to a **living representation of scientific practice**, where humans, machines, and ethics co-operate inside a single computational grammar.

This incorporation of human role in scientific narration is invaluable. From a technical standpoint, SNOWFLAKE's treatment of human entities introduces a representational and operational advance over all prior workflow architectures. This provides **causal completeness** — every procedural dependency, including manual calibration or verification, becomes a first-class element of execution. This design produces a more **realistic computational model of science**, since experimental outcomes are often contingent on human intervention, interpretation, or ethical compliance. The integration of **AccountabilityEntity** and **CompetencyLevel** attributes enforces verifiable authorship and authorization, turning provenance into a machine-checkable form of accountability. Similarly,

HumanFAIRIndex and **EthicalComplianceCheck** extend reproducibility metrics beyond automation to include cognitive and regulatory traceability, ensuring that both data and decisions are reproducible. As a result, the SNOWFLAKE engine can evaluate human-dependent workflow branches conditionally—pausing, verifying, or rejecting execution based on real-time confirmations. In effect, SNOWFLAKE fuses human cognition, instrumental control, and algorithmic execution into a unified formal system, enabling **reproducibility-aware, ethics-aware, and human-aware computation**

5.6. Representation of Scientific Instruments

SNOWFLAKE treats a workflow not just as a computational process, but as a **scientific knowledge instrument** — one that unifies procedural logic, epistemic intent, and provenance under a single descriptive model.

Unlike CWL, WDL, Nextflow, or Pegasus, which focus on computational *execution*, and unlike RO-Crate or SWIRL, which focus on *metadata interoperability*, SNOWFLAKE introduces an **ontological layer** where every workflow element (task, tool, actor, instrument, environment) carries **semantic attributes** such as *ScientificSignificance*, *FunctionalRole*, *Persistence*, and *HumanFAIRIndex*. This allows a SNOWFLAKE workflow to explicitly recognize both **digital and physical instruments**—for example, identifying an electron microscope as a calibrated data-producing entity, or a computational model as an analytic instrument. Among current workflow and provenance frameworks, most—such as CWL, WDL, Nextflow, Pegasus, and Galaxy—are designed to describe **computational processes**, not **physical instruments**. They can execute data-analysis steps that *use* data from an electron microscope or a refrigerator, but they cannot natively *recognize* or *describe* those instruments themselves. Only **RO-Crate** and the newer **SWIRL** interoperability framework allow instruments to be represented as **research objects** with metadata, identifiers, and provenance. Thus, only these two can formally describe classical scientific instruments within a workflow ecosystem, linking digital computation to the physical apparatus that generated the data. **SNOWFLAKE** surpasses existing workflow metadata frameworks by embedding scientific instruments into the **semantic, procedural, and epistemic fabric** of workflows. While **RO-Crate** and **SWIRL** can *record* instruments as metadata entities for provenance and interoperability, SNOWFLAKE can *instantiate* them as active, parameterized participants in the scientific process—each with role, calibration, accountability, and knowledge purpose. This transforms a workflow from a record of actions into a **living scientific apparatus**—a meta-instrument capable of representing, executing, and reasoning about real instruments and their contribution to discovery. In essence, SNOWFLAKE turns the workflow itself into a **meta-instrument of science**,

capable of describing, executing, and validating knowledge production as a reproducible and FAIR-aligned process.

5.7. KIP Classification of Innovation and Knowledge Engineering

A KIP (Knowledge Information Product) is an *intangible yet identifiable scientific information artifact* representing a **conceptual or cognitive construct** generated, refined, or validated through inquiry. It includes ideas, hypotheses, models, frameworks, and theories — all carrying epistemic value, provenance, and versioning, just like datasets or workflows.

Each KIP has a **persistent identifier**, a **Tier-1 Origin**, a **Representation Form**, and a **Functional Role** within the scientific knowledge cycle.

An innovative feature of **SNOWFLAKE** is the introduction of a formal language for representing and tabulating the **information artifacts** employed in scientific exploration through a structured **KIP (Knowledge Information Product) typology**. To the best of our knowledge, **knowledge artifacts**—including *ideas, insights, hypotheses, postulates, and models*—have not previously been subjected to systematic classification.
Through its **KIP (Knowledge Information Product) typology**, SNOWFLAKE makes it possible to catalogue and analyze *ideas, insights, hypotheses, and models*—the cognitive tools that have long driven exploration but have rarely been formally represented. The **KIP framework** recognizes these constructs as **authentic, reproducible, and citable information entities**, deserving the same care, traceability, and FAIR stewardship as data and software.
By linking conceptual artifacts to every stage of the scientific workflow, SNOWFLAKE opens a pathway toward a new discipline of **scientific knowledge engineering**, where the evolution of thought itself becomes transparent, analyzable, and shareable.

Each of these can be recorded as an Information Product Elements (IPE) along with more traditional items cataloged in earlier efforts. These IPE Instances can be linked as input/output/derivative to various C/M/H/I workflow element instances. Below is a SNOWFLAKE three tire IPE typology scheme. For example, a scientist can provide a plausible interpretation of a trend about reviewing a graph. The interpretation product can be catalogued as "*SNOW.I: M.E.M.KIP-06"*. Thus, SNOWFLAKE can be the facilitator for genesis, search, recognition of innovation products. IT can be foundational towards deeper scientific security.

Table- 101(b) Tier 1 – Classification of Origin of Information (Mode of Generation)

Describes *how* an information product comes into being — from observation, computation, or cognition.

CODE	Category	Subclass Examples	Definition / Scope
E	**Empirical**	*Observational, Experimental, Sensor*	Created by direct measurement or observation of phenomena.
C	**Computational**	*Simulation, Theoretical, Synthetic*	Generated through algorithmic, mathematical, or model-based reasoning.
D	**Derived**	*Analytical, Aggregated, Calibrated*	Produced by transforming or combining existing datasets.
M	**Mental***	*Idea, Hypothesis, Assumption, Model, Interpretation, Insight*	Arises from human cognition and intellectual activity—mental abstraction, pattern recognition, or synthesis of understanding—not directly measurable yet epistemically real.

**This fourth class captures "knowledge artifacts" such as an idea conceived, a hypothesis framed, etc.*

Table- 101(a) The Three Tire SNOWFLAKE Typology for Information Product

Tire	Tier	Conceptual Function	Examples
T1	**Tier 1: Origin**	Human cognitive and inferential generation of new knowledge elements.	Ideas, hypotheses, theories.
T2	**Tier 2: Representation**	Symbolic or textual expressions of conceptual knowledge (statements, equations, diagrams).	Hypothesis text, model equations, conceptual graphs.
T3	**Tier 3: Functional Role**	Frameworks guiding or interpreting empirical data.	Hypothesis guiding experiment, theory interpreting data.

Table-101(d) Tier 3 –Classification of Functional Role (Purpose in the Knowledge Process)

Describes *why* the information exists and *how* it functions in inquiry.

Code	Role	Definition / Scope	Illustrative Examples
P	**Primary (Raw Evidence)**	Direct empirical or computational records representing original observations of phenomena.	Sensor readings, field logs, simulation outputs.
S	**Secondary (Processed Product)**	Analytical, derived, or interpreted outputs created by transforming or analyzing primary sources.	Derived datasets, statistical summaries, fitted models.
R	**Reference (Validation / Calibration)**	Standardized or benchmark information used to compare, verify, or calibrate other data.	Control datasets, reference spectra, gold-standard curves.
C	**Contextual (Metadata / Documentation)**	Descriptive or procedural information that provides context, intent, provenance, or methodological detail.	Experimental design, workflow description, provenance records.
M	**Mental (Cognitive / Theoretical Insight)***	Intangible intellectual artifacts that originate from human reasoning, abstraction, or theorization and structure the understanding of data.	Ideas, hypotheses, conceptual models, theoretical propositions.

Table-101(d) Conceptual Information Product (CIP) Classification Table (v1.0)

KIP ID	Information Product Type	Definition	Typical Origin (Tier 1)	Representation (Tier 2)
KIP-01	**Idea / Insight**	A novel or emergent conceptual realization linking phenomena, often intuitive or qualitative.	**Mental***	Ephemeral → Digital (Textual / Visual
KIP-02	**Hypothesis**	A testable propositional statement predicting a relationship or mechanism between variables.	**Mental***	Digital – Symbolic / Textual
KIP-03	**Assumption**	A foundational premise accepted provisionally to enable reasoning or modeling.	**Mental***	Digital – Textual
KIP-04	**Theoretical Model**	A structured abstraction representing causal or mathematical relationships within a system.	Computational – Theoretical / Menta	Digital – Symbolic / Visual
KIP-05	**Conceptual Framework**	An integrated system of related hypotheses, models, or principles guiding inquiry within a domain.	**Mental***	Digital – Textual / Visual
KIP-06	**Interpretation**	An explanatory mapping of data patterns to meaning or theoretical context.	Derived / Mental	Digital – Textual / Visual
KIP-07	**Prediction / Expectation**	A quantified or qualitative outcome logically implied by a model or hypothesis.	Computational / Mental	Digital – Symbolic / Numerical
KIP-08	**Design / Plan / Protocol**	A structured arrangement of actions, parameters, or configurations to test a hypothesis or produce data.	Conceptual / Cognitive	Digital – Textual / Visual
KIP-09	**Algorithm / Method**	Formalized sequence of operations to derive results or transform data.	Computational – Theoretical	Digital – Symbolic / Code
KIP-10	**Schema / Ontology / Taxonomy**	A structured conceptual model defining entities, attributes, and relationships for knowledge organization.	**Mental***	Digital – Symbolic / Textual
KIP-11	**Inference / Conclusion**	Logical outcome derived from data analysis or reasoning, closing a cycle of inquiry.	Derived / Mental	Digital – Textual
KIP-12	**Principle / Law**	A general and reproducible relationship describing consistent behavior in nature.	**Mental***	Digital – Symbolic
KIP-13	**Paradigm / Theory System**	A comprehensive explanatory architecture integrating multiple models and laws.	**Mental***	Digital – Textual / Visual
KIP-14	**Heuristic / Rule of Thumb**	A simplified mental or procedural shortcut derived from accumulated experience.	**Mental***	Ephemeral → Digital (Textual)
KIP-15	**Question / Problem Statement**	A formal articulation of uncertainty or research objective driving inquiry.	**Mental***	Digital – Textual
KIP-20	**Other**			

6. Dimensions of Scientific Projects

The **SNOWFLAKE** framework, as designed, possesses the structural capacity to represent an exceptionally rich set of **eleven dimensions of a scientific project**, which is precisely what makes it both novel and powerful. Below is a detailed explanation of these eleven dimensions in terms of the SNOWFLAKE schema's semantics ad attribute coverage.

6.1. Understanding Project Goals

Every SNOWFLAKE workflow begins with a clear declaration of purpose. The **WorkFlowTitle**, **WorkFlowDescription**, and **DomainKeywords** together define *what the project aims to accomplish*, *why it matters*, and *in which scientific or applied context it belongs*. By requiring these structured fields, SNOWFLAKE ensures that each scientific project encodes its central research question or engineering goal, allowing others to quickly grasp its intent without ambiguity. The language thus transforms project motivation into machine-readable metadata—something traditional research documentation rarely achieves.

6.2. Capturing Significance and Purpose

Beyond the "what," SNOWFLAKE records the *why*—the broader significance and motivation behind a project. Through attributes such as **WorkflowNarrator**, **CollaborationNetwork**, and **KnowledgeAreas**, each workflow carries a brief narrative about its importance, contributors, and context within a larger scientific endeavor. This helps situate the workflow's purpose within the discipline's evolving landscape, showing how it contributes to ongoing inquiry or practical outcomes. In this sense, SNOWFLAKE not only documents process but embeds the intellectual rationale that drives discovery.

6.3. Representing the Approach and Methodology

The heart of any scientific project lies in *how* it is carried out. SNOWFLAKE expresses this through the **FunctionalDescription** attribute present in every class of work element—human, computational, or machine. These fields describe, in the language of the domain, what each component does, how it transforms inputs into outputs, and what assumptions or methods underlie it. Together with **ParametricDescriptors** and **CardinalityConditions**, these entries form a blueprint of the project's procedural logic, allowing others to reconstruct or simulate the original methodology.

6.4. Expressing the Mode of Work

SNOWFLAKE models a project as a directed workflow graph, making explicit the *mode of work*—the flow of tasks, dependencies, and conditions. Links defined by **LinkType**, **CardinalityType**, and **Execution Conditions** show how actions depend on one another, whether steps occur sequentially or in parallel, and under what conditions particular branches execute. This structure captures not just static organization but the dynamic behavior of a scientific project: its rhythm, synchronization, and flow of information across collaborators and systems.

6.5. Documenting Human Resource Utilization

Science is, at its core, a human enterprise. SNOWFLAKE recognizes this by explicitly representing the roles and contributions of people within the workflow. Each **HumanWorkElement** includes attributes such as **ActorRole**, **ActorLocation**, **TaskTime**, and the total **#HumanWorkerInstances** involved. These attributes map who performs each task, from where, and for how long. The result is a formal record of human participation and expertise—a basis for understanding the intellectual and labor structure of modern collaborative science.

6.6. Capturing Computational Resource Utilization

Most contemporary scientific projects rely on computational systems. SNOWFLAKE captures this through **ComputationalWorkElements**, which specify the **ComputeResource**, **ExecutionTime**, **Memory**, **Environment**, and **Tools** associated with each computational task. This level of detail enables assessment of computational efficiency, cost, and scalability, while allowing workflows to be re-executed or ported to different environments. It also supports meta-analysis: comparing resource profiles across projects to understand computational demands of scientific discovery.

6.7. Recording Machine and Instrument Utilization

In addition to computation, many projects depend on laboratory or field instruments. **MachineWorkElements** in SNOWFLAKE include **CapacityDescription**, **AllocatedConsumables**, **Persistence**, and **Environment**, creating a comprehensive profile of each instrument or device. This ensures that physical context—how sensors, microscopes, or robots contribute—is not lost in abstraction. By formalizing machine roles, SNOWFLAKE bridges the gap between digital data and physical observation, completing the loop of reproducibility.

6.8. Describing Data and Information Flow

Scientific knowledge is carried through data, and SNOWFLAKE models it explicitly via **InformationProductElements**. These define the structure (**SchmaticDescription**), size (**Filesize**), and transformation characteristics (**ManipulationTime**, **MemoryNeed**) of each data artifact. Links between elements describe how information moves between humans, software, and instruments. This converts data lineage—often implicit—into a traceable, queryable network of information, enabling provenance tracking and downstream validation.

6.9. Encoding Collaboration and Roles

Collaboration lies at the heart of modern science. SNOWFLAKE makes the extent and nature of collaboration explicit through attributes like **CollaborationNetwork**, **ActorRoleSet**, **ProtocolOwner**, and **ProtocolDesigner**. These describe institutional and interpersonal connections, ownership of methods, and flow of responsibility. By doing so, SNOWFLAKE transforms the invisible web of teamwork into structured metadata—revealing how ideas, skills, and authority move across boundaries in a scientific enterprise.

6.10. Representing Duration and Temporal Scope

Time is an essential dimension of any project. SNOWFLAKE captures it at multiple scales through **TaskTime**, **ExecutionTime**, **Persistence**, and **WorkflowLifetime**.

Each element carries its own temporal metadata—indicating frequency, repetition, or long-term persistence. This allows projects to be understood not as static graphs but as evolving processes with defined life cycles, from one-time experiments to continuous monitoring networks. Temporal encoding also enables simulation, scheduling, and comparative timing analysis across projects.

6.11. Representing Interdisciplinarity and Knowledge Domains

Finally, SNOWFLAKE acknowledges that contemporary research rarely fits within a single discipline. Attributes such as **KnowledgeAreas**, **DomainKeywords**, and **Tools** provide a high-level description of a project's intellectual and technological domains. This allows workflows to be classified, indexed, and discovered across scientific boundaries, supporting meta-research and cross-disciplinary analytics. By encoding disciplinary diversity as structured metadata, SNOWFLAKE makes visible the converging nature of modern science.

Table-1 shows the representative attribute elements for these dimensions. Together, these eleven dimensions turn a SNOWFLAKE workflow into a **complete conceptual and operational map** of a scientific project — from motivation to execution, collaboration, and knowledge outcome. It is a language not only for *what science does*, but for *how science works*.

Table-1 SNOWFLAKE's Coverage of Science Project's Dimensions

Aspect of Understanding	Where It Appears in SNOWFLAKE	Description / Example
1. Project Goals	WorkFlow Title, WorkFlow Description, DomainKeywords, KnowledgeAreas	Each workflow begins with a textual and conceptual definition of purpose — what the workflow achieves and its significance within a discipline.
2. Scientific Significance / Purpose	WorkFlowDescription, WorkflowNarrator, CollaborationNetwork	Narrative and contextual elements articulate *why* the workflow exists — the motivation, domain importance, and societal or scientific value.
3. Approach and Methodology	FunctionalDescription (in HW, CW, MW, IP)	Every human, computational, machine, and information element documents

		what it does in domain-specific language — collectively defining the method.
4. Mode of Work (how the work proceeds)	Workflow graph structure + CardinalityCondition + LinkType	The directed graph expresses sequencing, concurrency, dependencies, and conditional execution — capturing how the work actually unfolds.
5. Human Resource Utilization	HumanWorkElement. ActorRole, #HumanWorkerInstances, ActorLocation, TaskTime	Specifies the roles, counts, durations, and contexts of human participation — who does what, where, and for how long.
6. Computing Resource Utilization and Requirements	ComputationalWorkElement.ComputeResource, ExecutionTime, Memory, Environment, Tools	Defines compute nodes, resource consumption, execution duration, and software stack — precise mapping of digital workload.
7. Machine / Instrument Resource Utilization	MachineWorkElement .CapacityDescription, AllocatedConsumables, Persistence, Environment	Encodes specifications, consumables, and usage frequency for lab instruments or hardware systems.
8. Data and Information Flow	InformationProductElement.ParametricDescriptors, SchmaticDescription, Filesize, MemoryNeed	Models every data entity, schema, and their interactions within the workflow.
9. Collaboration and Roles	ActorRoleSet, CollaborationNetwork, ProtocolOwner, ProtocolDesigner	Expresses *who collaborates with whom*, the ownership of processes, and institutional or international partnerships.
10. Duration and Temporal Scope	TaskTime, ExecutionTime, Persistence, WorkflowLifeTime	Provides timing, frequency, and persistence — enabling timeline reconstruction and scheduling.
11. Extent of Interdisciplinarity / Knowledge Domain	KnowledgeAreas, DomainKeywords, Tools	Captures the disciplinary and technical diversity of the workflow, defining its multi-domain footprint.

7. Support for Campus Research Support Engineering

The **SNOWFLAKE framework** can transform how campus Research Computing and IT Infrastructure Units onboards and maintains interdisciplinary projects on shared computing clusters. When a principal investigator seeks assistance in migrating a research workflow, engineers must rapidly determine its computational footprint, data handling requirements, compliance obligations, and execution dependencies. SNOWFLAKE offers a **unified, semantically structured metadata layer** that makes this information **immediately discoverable and machine-interpretable**. Its standardized descriptors reveal each project's hardware, software, data, and governance needs, allowing engineers to allocate cluster resources, plan data storage, and configure security policies efficiently—without repeated manual consultations. The following subsections illustrate typical engineering questions and show how SNOWFLAKE attributes provide direct, actionable answers.

7.1. Assessing Computational Environment Requirements

Before migrating a workflow to a shared cluster, systems engineers must understand its computational dependencies and runtime environment. SNOWFLAKE records these details through attributes such as **ComputeResource**, **Environment**, and **Tools** within the *ComputationalWork* layer, complemented by **Infrastructure & Resources** in the *Workflow* layer. Together, these descriptors specify CPU/GPU needs, required operating systems, libraries, and execution environments. By referencing these structured fields, administrators can evaluate compatibility with existing cluster modules, identify software or driver dependencies, and pre-allocate suitable nodes—enabling an automated and frictionless deployment process.

7.2. Evaluating Data Scale, Sensitivity, and Storage Constraints

Understanding the nature and sensitivity of data products is equally essential for cluster integration. SNOWFLAKE captures this through **SchematicDescription**, **FileSize**, and **Environment** in the *InformationProduct* layer, along with **LicenseType** and related provenance attributes. These descriptors reveal schema complexity, dataset volume, expected data persistence, and protection requirements such as encryption or restricted access. IT teams can use them to determine whether parallel file systems, object-store tiers, or controlled data partitions are appropriate, and to design compliant data-movement and retention strategies that uphold confidentiality and regulatory obligations.

7.3. Mapping Execution Order and Scheduling Dependencies

Efficient cluster scheduling requires explicit knowledge of task timing and dependency graphs. Within SNOWFLAKE, **ExecutionTime** attributes across *HumanWork*, *ComputationalWork*, and *MachineWork* entities, combined with **LinkType**, **CardinalityType**, and **CardinalityCondition**, collectively define a directed graph of dependencies and recurrence patterns. These relationships specify which tasks may execute concurrently, which must await completion, and how frequently they recur. Such structured relationships allow engineers to translate scientific workflows directly into **schedulable jobs** in cluster managers like *Slurm*, *PBS*, or *Kubernetes*, ensuring optimal parallelization and resource utilization.

7.4. Security and Compliance Units

For cybersecurity, data protection, and research compliance divisions, SNOWFLAKE provides transparent traceability of accountability, verification, and data-handling practices. Key attributes like **SecurityReview**, **AccountabilityEntity**, and **TaskVerificationMethod** document encryption standards, responsibility assignments, and audit mechanisms. This allows compliance teams to map data flows against institutional or regulatory frameworks such as HIPAA, GDPR, or NIST SP-800-53, ensuring that sensitive data never leaves secure domains. In addition, these descriptors support periodic compliance reviews and incident response investigations without interrupting active research.

7.5. Defining Accountability and Verification Mechanisms

In production environments, it is vital to identify ownership and quality-assurance responsibility for every workflow component. SNOWFLAKE explicitly connects human and organizational roles through attributes such as **ActorRoleSet**, **AccountabilityEntity**, **TaskVerificationMethod**, and **ServiceOwnerAccountability**. These ensure that each computational or experimental process has a designated maintainer and a documented verification mechanism. The result is a clear delineation of support boundaries between the research group and the infrastructure team, establishing an **auditable chain of accountability** for change control, debugging, and performance validation.

7.6. Ensuring Compliance, Ethics, and Open-Science Readiness

Cluster integration must also conform to ethical, regulatory, and institutional policies. SNOWFLAKE centralizes this information using attributes such as **EthicalReview**, **SecurityReview**, **EnvironmentalReview**, and **IRBClassification**, along with **OpenScienceStatement** and **CumulativeFAIRIndex**. These collectively inform engineers about required data-handling safeguards, audit mechanisms, and openness policies before execution.

By embedding these compliance and transparency indicators directly within workflow metadata, SNOWFLAKE bridges the gap between **scientific reproducibility** and **institutional governance**, providing a complete metadata foundation for securely and transparently deploying research projects on campus high-performance computing infrastructure.

7.7. Supporting Post-Onboarding Maintenance and Lifecycle Management

Once a research project is operational on a shared cluster, its maintenance becomes an ongoing challenge involving software updates, user transitions, resource scaling, and compliance re-validation. SNOWFLAKE provides persistent, queryable metadata that simplifies this entire lifecycle. Attributes such as **VersionHistory**, **TaskVerificationMethod**, and **AccountabilityEntity** allow administrators to track component revisions and re-certify dependencies after upgrades or hardware changes. **Persistence**, **Environment**, and **ExecutionTime** descriptors support continuous performance monitoring—highlighting when workloads deviate from expected run times or when scaling is required.

Moreover, **ServiceOwnerAccountability** and **ActorRoleSet** attributes maintain institutional memory of who is responsible for each module, even after project personnel change, while **SecurityReview** and **EthicalReview** fields ensure that access policies and compliance documents remain synchronized with institutional standards. In effect, SNOWFLAKE functions as a **living operational record**—enabling reproducible reruns, efficient debugging, and accountable maintenance across the project's full lifespan, well beyond its initial onboarding.

8. Other Institutional Usage of SNOWFLAKE

The SNOWFLAKE framework benefits a broad range of institutional units beyond Research Computing and IT Infrastructure Support, across a university by converting complex research activities into structured, interoperable, and intelligible narratives. Each unit—academic, administrative, or outreach-oriented—derives distinct value from the same metadata fabric, reducing duplication of effort and enabling more coherent institutional knowledge flows. Below is a representative set of parties.

8.1. Institutional Review Board (IRB) and Human-Subjects Oversight

SNOWFLAKE streamlines ethical oversight by embedding IRB-related information directly into the workflow metadata. Attributes such as **EthicalReview**, **IRBClassification**, and **HumanFAIRIndex** identify whether human participants are involved, the level of review required (exempt, expedited, or full), and whether anonymization or consent mechanisms are implemented. When an IRB officer reviews a new project submission, these descriptors enable rapid pre-screening of research involving

personal data or clinical samples, reducing compliance delays and ensuring early identification of projects requiring formal ethical approval.

8.2. University Libraries and Data Stewardship Offices

University libraries and research-data offices can use SNOWFLAKE as a metadata gateway for digital preservation and open-access publication. Through **InformationObject**, **SchematicDescription**, **LicenseType**, and **FAIRIndex**, the library gains a structured overview of datasets, formats, and rights of use. When researchers deposit project data or software, librarians can directly ingest SNOWFLAKE metadata into institutional repositories or national data commons (e.g., Zenodo, Figshare) with consistent FAIR-aligned documentation. This reduces curatorial overhead and ensures long-term findability, accessibility, and reusability of research outputs.

8.3. Communications, Media, and Public Relations Offices

University communications and media offices often struggle to translate technical projects into narratives accessible to the public and funding stakeholders. SNOWFLAKE simplifies this task by making contextual information such as **ScientificSignificance**, **BroaderImpact**, and **OpenScienceStatement** machine-readable and discoverable. Media professionals can identify projects demonstrating innovation, societal relevance, and collaboration from these metadata fields and rapidly develop feature stories or press releases. The ability to access verified information directly from structured project descriptors reduces dependence on time-intensive interviews and ensures scientific accuracy in outreach materials.

8.4. Academic Departments and Curriculum Committees

Academic departments benefit from SNOWFLAKE's capacity to connect ongoing research projects to instructional objectives and experiential learning opportunities. Attributes such as **KnowledgeAreas**, **MethodologyExperimentalDesign**, and **Tools** allow faculty to identify workflows suitable for classroom demonstrations, capstone projects, or undergraduate research modules. This integration strengthens the research–teaching continuum, allowing departments to embed real-world data processing and reproducibility practices into the curriculum. Students, in turn, gain direct exposure to authentic workflows aligned with institutional expertise.

8.5. Technology Transfer and Intellectual Property (IP) Offices

SNOWFLAKE assists technology-transfer and intellectual-property units in identifying novel and commercializable outputs within the university's research portfolio. Fields such as **InnovationVector**, **ProtocolOwner**, and **ProtocolType** document methodological originality, ownership, and reuse rights, respectively. These descriptors allow IP officers to assess whether a workflow component constitutes a new algorithm, device, or method suitable for patenting or licensing. By automating discovery of potential innovations, SNOWFLAKE accelerates technology assessment pipelines and reduces missed opportunities for early-stage IP protection.

8.6. Institutional Assessment and Strategic Planning Offices

University leadership and strategic-planning offices rely on accurate metrics to evaluate research performance and interdisciplinary engagement. SNOWFLAKE's attributes—particularly **CollaborationNetwork**, **WorkflowFAIRIndex**, and **InnovationReadinessLevel**—allow aggregation of cross-project indicators such as disciplinary diversity, openness scores, and technology maturity. These metrics provide empirical evidence for accreditation, benchmarking, and institutional rankings. The structured metadata also reveals trends in collaboration networks, guiding investment in emerging research domains and interdepartmental programs.

8.7. Principal Investigators, Postdoctoral Fellows, and Students

SNOWFLAKE can indeed be most helpful to the core research teams- to better organize, communicate, and sustain knowledge continuity across time much more effectively than it is possible today- and dramatically increasing team's scientific productivity. Most of today's team communication is personality driven, often information needed to be effective is often inherited from personal communication, fragmented notes, often student mentees develop unclear expectations. In contrast SNOWFALKE can enable investigator and student alike to enter a project environment where **their role, purpose, and interconnections are articulated in a living digital record**. Not only, an individual's own role, SNOWFLAKE links each one role in the broader context of project's science- where human expertise, computational execution, and data provenance are narrated within one coherent semantic framework- this can enormously facilitate scientific communication among the core team. For example, a new graduate student joining a multidisciplinary lab, rather than piecing together instructions from scattered emails and hallway conversations, can use SNOWFLAKE dashboard to see themselves identified in the **ActorRoleSet** as *"Data Analyst – Validation Stream,"* with a **FunctionalDescription** that explains their role, contribution, and other entities they need to connect. The **CompetencyLevel** entry outlines the skills expected—say, Python scripting and statistical modeling—while the **TaskVerificationMethod** clarifies how their work will be reviewed and integrated by the supervising postdoctoral researcher. The project narrative, stored under **WorkFlowAbstract** and **MethodologyExperimentalDesign**, contextualizes why their role matters within the scientific aim. Student would no longer need to decode informal lab culture to find out what they are supposed to do; mentors could track progress transparently and collaborations could scale without losing institutional/project memory for evolving projects. By treating the human dimension of research as a formal, describable part of the workflow—on par with datasets and code—SNOWFLAKE a new level of clarity which is not available in any current Workflow language or framework.

9. Cost of Maintaining SNOW for Each Project

The cost of maintaining a SNOW (Scientific Narrative of Workflow) record for each project is relatively modest compared to the administrative and communication efficiencies it provides. Most of the data elements in the SNOW structure—such as *WorkFlowTitle*, *ScientificDescription*, and *Milestones*—already exist in grant proposals, project management reports, or publications. Thus, the incremental cost lies primarily in metadata curation and periodic updates. In practice, a trained data steward or graduate assistant can complete an initial SNOW entry in **2–3 hours** using structured templates, with an annual update taking less than **one hour** per project.

At scale, for institutions hosting hundreds of active projects, this translates to a lightweight metadata layer that can be integrated with existing research information systems (e.g., Symplectic, Pure, or local grant-tracking portals). The operational expense per project is estimated at **$50–$75 per year**, depending on automation levels, storage architecture, and whether the metadata entry is self-declared or centrally validated.

Moreover, once SNOW records are embedded in the research workflow—from proposal submission to publication—the cost curve declines sharply. Automated extraction from grant documents, lab notebooks, or repositories can populate up to **70% of the attributes**, leaving only interpretive fields (e.g., *BroaderImpact*, *OpenScienceStatement*) for manual input. This hybrid model keeps SNOW maintenance sustainable, minimizes researcher burden, and ensures continuous metadata freshness.

Finally, the return on investment is high: SNOW records not only support **IT onboarding** and **media communication** but also enhance **grant compliance, reproducibility, and institutional reporting**. The small cost of maintaining SNOW per project is thus best viewed as a shared infrastructure investment—comparable to maintaining a DOI registry or ORCID system—that pays dividends in visibility, accountability, and interoperability across research domains.

10. Conclusions

The project has been supported by funding from the National Science Foundation NSF Award#2201558, and NSF Award# 1925678, engineering time contributed by the Division of Information Technology, location and engineering time donated by Department of Computer Science.

11. References

[1] van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B. and Barros, A.P. (2003) 'Workflow patterns', Distributed and Parallel Databases, 14(1), pp. 5–51., Available at: https://doi.org/10.1023/A:1022883727209.

[2] Deelman, E., Singh, G., Su, M.-H., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Vahi, K., Berriman, B., Good, J., Laity, A., Jacob, J.C. and Katz, D.S. (2005) 'Pegasus: A framework for mapping complex scientific workflows onto distributed systems', Scientific Programming, 13(3), pp. 219–237., Available at: https://doi.org/10.1155/2005/914734.

[3] Taylor, I.J., Deelman, E., Gannon, D. and Shields, M. (eds.) (2007) Workflows for e-Science: Scientific Workflows for Grids. London: Springer., Available at: https://doi.org/10.1007/978-1-84628-757-2.

[4] Curcin, V. and Ghanem, M. (2008) 'Scientific workflow systems—Can one size fit all?', in Proceedings of the Cairo International Biomedical Engineering Conference (CIBEC 2008). Cairo: IEEE, pp. 1–9., Available at: https://doi.org/10.1109/CIBEC.2008.4786085.

[5] Afgan, E., Baker, D., Batut, B., Van den Beek, M., Bouvier, D., Čech, M., Chilton, J., Clements, D., Coraor, N., Grüning, B.A. et al., 2018. "The Galaxy platform for accessible, reproducible and collaborative biomedical analyses." Nucleic Acids Research, 46(W1), W537–W544. DOI: 10.1093/nar/gky379.

[6] Amstutz, P., Crusoe, M.R., Tijanić, N., Chapman, B., Chilton, J., Heuer, M., Soiland-Reyes, S. and Khan, F., 2016. Common Workflow Language, v1.0. figshare. DOI: 10.6084/m9.figshare.3115156.

[7] Broad Institute, 2024. Terra Platform and Cromwell Workflow Engine Documentation. Available at: https://terra.bio.

[8] Deelman, E., Vahi, K., Juve, G., Rynge, M., Callaghan, S., Maechling, P.J., Mayani, R., Chen, W., Ferreira da Silva, R., Livny, M. and Wenger, K., 2015. "Pegasus, a Workflow Management System for Science Automation." Future Generation Computer Systems, 46, pp.17–35. DOI: 10.1016/j.future.2014.10.008.

[9] Di Tommaso, P., Chatzou, M., Floden, E., Barja, P.P., Palumbo, E. and Notredame, C., 2017. "Nextflow enables reproducible computational workflows." Nature Biotechnology, 35(4), pp.316–319. DOI: 10.1038/nbt.3820.

[10] Ewels, P.A., Peltzer, A., Fillinger, S., Patel, H., Alneberg, J., Wilm, A., Garcia, M.U., Di Tommaso, P. and Nahnsen, S., 2020. "The nf-core framework for community-curated bioinformatics pipelines." Nature Biotechnology, 38(3), pp.276–278. DOI: 10.1038/s41587-020-0439-x.

[11] Goble, C., Soiland-Reyes, S., Hardisty, A., Huber, W., Barthel, R. and Castro, L.J., 2024. "SWIRL: A Scientific Workflow Interoperability Representation Language." arXiv preprint arXiv:2403.08521. Available at: https://arxiv.org/abs/2403.08521.

[12] Goble, C.A., Cohen-Boulakia, S., Soiland-Reyes, S., Garijo, D., Gil, Y., Crusoe, M.R. and Peters, K., 2020. "FAIR computational workflows." Data Intelligence, 2(1–2), pp.108–121. DOI: 10.1162/dint_a_00033.

[13] Jalili, V., Afgan, E., Gu, Q., Clements, D., Blankenberg, D., Goecks, J., Taylor, J. and Nekrutenko, A., 2020. "The Galaxy platform for accessible, reproducible, and collaborative biomedical analyses: 2020 update." Nucleic Acids Research, 48(W1), W395–W402. DOI: 10.1093/nar/gkaa434.

[14] Soiland-Reyes, S., Bacall, F., Owen, S., Lusher, S., Goble, C.A. et al., 2022. "Packaging research artefacts with RO-Crate." Data Science Journal, 21(1), p.10. DOI: 10.5334/dsj-2022-010.

[15] Sufi, S., Sansone, S.A., Goble, C.A., Hardisty, A. and Cohen-Boulakia, S., 2023. "EOSC-Life WorkflowHub: FAIR sharing and discovery of computational workflows." Frontiers in Research Metrics and Analytics, 8, 1082443. DOI: 10.3389/frma.2023.1082443.

[16] Vahi, K., Rynge, M., Juve, G., Mayani, R., Silva, R.F., Deelman, E. and Livny, M., 2020. "Pegasus: Enhancing the Performance and Reliability of Scientific Workflows on HPC and Distributed Systems." Computing in Science & Engineering, 22(3), pp.52–63. DOI: 10.1109/MCSE.2020.2968421.

[17] Javed I. Khan and Philip Thomas, *An Exploration into Scientific Narrative Of WorkFLow for Learning, Analytics and Knowledge Engineering*, Technical Report 2025-03-01 Internetworking and Media Communications Research Laboratories, Department of Computer Science, Kent State University, available from: http://medianet.kent.edu/technicalreports.html

[18] Javed I. Khan and Philip Thomas, *Semantic Container for Scientific Narrative Of WorkFLow for Learning, Analytics and Knowledge Engineering*, Technical Report 2025-03-02 Internetworking and Media Communications Research Laboratories, Department of Computer Science, Kent State University, available from: http://medianet.kent.edu/technicalreports.html