Enabling FAIR Research Lifecycle Provenance: The RDIP Project-Centric Integration Profile

Suhel Acharya[1], Chutiporn Anutariya[1], Yasuyuki Minamiyama[2], Hideaki Takeda[3], and Vilas Wuwongse[4]

[1] Faculty of Science and Advanced Technology, Asian Institute of Technology, Thailand
{suhel.acharya,chutiporn}@ait.ac.th
[2] Institute of Social Science, the University of Tokyo
minamiyama@iss.u-tokyo.ac.jp
[3] National Institute of Informatics, Japan
takeda@nii.ac.jp
[4] Mahasarakham University, Thailand
vilasw@gmail.com

Abstract. Research outputs, including datasets, software, and publications, are typically described in isolation across repositories and scholarly information systems. This fragmentation makes it difficult to reconstruct the research lifecycle or to support automated reuse in line with the FAIR principles. We present RDIP, a lightweight, project-centric integration profile, composing VIVO, DCAT, PROV-O, BIBO, and CiTO around a central `rdip:ResearchProject` hub. RDIP adds reproducibility-oriented constructs grounded in the ML Reproducibility Checklist and ACM Artifact Review policy (hyperparameters with random seeds, computing environments, evaluation results with dataset split labels, software dependencies, and dataset derivation lineage) together with qualified PROV-O association and activity-sequencing patterns. We validate RDIP through four complementary analyses: (i) a demonstration that five common project-centric provenance queries cannot be answered by DCAT and PROV-O alone; (ii) a 344-triple gold-standard knowledge graph derived from twelve machine-learning papers, against which all eleven competency questions return non-empty results; (iii) a schema-coverage analysis across 95 papers showing $\geq$94% applicability for core classes; and (iv) an LLM-extraction experiment confirming RDIP's utility as a target schema.

Keywords: Research Knowledge Graph · Integration Profile · FAIR Principles · Provenance · Research Data Management · Reproducibility

1 Introduction

The scientific community faces a reproducibility crisis driven by the fragmentation of scholarly knowledge. While research outputs grow exponentially, the contextual links between data, software, and methods remain buried in unstructured text or isolated silos. This semantic gap hinders the automated implementation of

the FAIR principles [29]; for instance, tracing a dataset back to the specific software version used and the researcher accountable for its production often requires manually connecting disconnected text, identifiers, and contributor role statements.

Recent advancements in Large Language Models (LLMs) offer a promising avenue for automated contextual links extraction as a set of metadata elements [5,31], but they require robust target schemas to mitigate hallucination [14]. Existing metadata standards address individual facets of this problem such as DCAT for dataset cataloguing [8], PROV-O for generic provenance [30], VIVO for scholarly profiles [28]. However, no single vocabulary provides a schema that integrates activities, software stacks, methods, hyperparameters, and evaluation results into a coherent, queryable graph.

To address this gap, we present RDIP, a project-centric integration profile [27] for FAIR research lifecycle provenance. Rather than introducing new concepts, RDIP composes established vocabularies; VIVO for people and organisations, DCAT for datasets, PROV-O for provenance, BIBO for publications [6], and CiTO for citation typing [23]; into a unified schema anchored by the `rdip:ResearchProject` class. The profile incorporates reproducibility-oriented constructs: software dependencies with version constraints, computing environments with container image specifications, hyperparameters and random seeds as first-class entities, and evaluation results with dataset split labels.

Our research contributions are: **(1)** An integration profile that bridges the gap between existing vocabularies by providing a project-centric coordination layer for research lifecycle provenance, with constructs for computational reproducibility. **(2)** A demonstration, through five executable SPARQL queries, that DCAT and PROV-O alone cannot answer project-centric reproducibility questions. **(3)** A large-scale evaluation on 95 real machine-learning research papers with associated GitHub repositories, comprising a gold knowledge graph (12 papers, 344-triples, 11 competency questions) and a schema coverage analysis demonstrating $\geq$94% applicability for core RDIP classes. **(4)** An LLM extraction experiment comparing three prompt conditions, showing preliminary evidence that RDIP enables typed relation extraction where simpler vocabularies largely fail.

RDIP is the schema layer of a broader system, the Research Data Intelligence (RDI) Platform, under development at our institution. The platform combines the integration profile presented here with knowledge graph storage, LLM-based metadata extraction, and FAIR-aligned discovery interfaces. This paper scopes its contributions strictly to the integration profile as a target schema (defining what a valid extraction must contain, not how to extract it) and presents its design, evaluation, and evidence of utility for automated extraction. The remaining components of the RDI Platform are the subject of ongoing work and are not evaluated here.

The ontology and all evaluation resources are available at `https://w3id.org/rdip` with persistent URIs and Widoco [11] generated documentation.

2 Related Work

The design of the RDIP integration profile is situated at the intersection of several active research areas, including scholarly ontologies, research data management (RDM), and process-centric or component-centric models for scientific workflows and software. In this section, we analyse representative approaches and position RDIP with respect to the specific gaps they leave at the project-centric, knowledge-layer level.

2.1 Project Identifiers and the Semantic Gap

Recent initiatives such as the Research Activity Identifier (RAiD) [4] have highlighted the importance of "Projects" as the central unit of research administration. RAiD provides a persistent identifier (PID) for a project, linking it to contributors and organisations. However, RAiD functions primarily as a "container" or administrative handle. It does not provide the internal semantic schema to model the granular execution of the research lifecycle, such as the specific software version used in a data cleaning activity.

Similarly, while DCAT [8] excels at cataloguing static resources and PROV-O [30] provides a generic grammar for causality, neither offers a unified schema for computational provenance that is both rigorous enough for queries (e.g., "Find all datasets generated by PyTorch v1.8") and simple enough for automated extraction. RDIP addresses this gap by adopting the "Project" as the semantic root, using it to anchor the specific activities, software agents, data outputs, and reproducibility artefacts into a coherent narrative.

2.2 Foundational and Management-Layer Ontologies

RDIP builds upon a set of widely adopted, community-driven ontologies that each model a specific facet of the scholarly ecosystem. VIVO [28] provides a canonical vocabulary for representing scholars, organisations, and their roles in the research landscape (e.g., `vivo:Person`, `vivo:Organization`). It excels at capturing people and affiliations but does not provide an explicit, provenance-aware link between activities, software, datasets, and outputs at the project level. BIBO [6] offers a standard model for bibliographic resources, but is document-centric, DCAT [8] is the W3C standard for describing datasets, but does not embed them in a project context, PROV-O [30] provides a generic provenance language without prescribing how to anchor activities to a higher-level project, CiTO [23] enables semantically precise citation relations but focuses on inter-entity relationships rather than integrated representation.

Beyond these foundational scholarly ontologies, several models target the management layer of research data rather than the semantic structure of research content. The ontology developed for the National Institute of Informatics's Research Data Cloud (NII RDC) demonstrates how an RDM system uses an ontology to track administrative processes, roles, activities, and policies [17]. Such ontologies typically focus on policies, storage, access rights, and plans, providing

excellent support for compliance and stewardship but only limited insight into the intellectual content and knowledge structures produced by a project. The Project Management Ontology (PMO) [13] models tasks and resources but focuses on management rather than scientific content.

In contrast, the RDIP integration profile provides a complementary knowledge layer for semantic reproducibility. Its purpose is to model the semantic narrative and intellectual content of research: the concepts, methods, software, outputs, and their interrelationships.

2.3 Process-Oriented Models and Application Profiles

Recent research validates the need to model the research process with greater granularity. The FAIR Implementation Profile (FIP) ecosystem models FAIR-supporting resources and profiles for FAIRification across domains [15]. While this confirms our methodology of explicitly representing FAIR-relevant entities, its primary focus is on resources and implementation profiles rather than on an end-to-end, project-centric knowledge layer. Other work has focused on deep models for individual components: The research by [16] presents a detailed ontology specifically for biomedical data analysis software, providing rich descriptions of software functions, dependencies, and versions. Our contribution differs in scope: we integrate a sufficiently detailed representation of software into a larger, comprehensive model of the entire research project, connecting it to activities, teams, and outputs.

Recent work on regulation-driven application profiles demonstrates a complementary approach to vocabulary integration. AIDOC-AP [18] addresses the technical documentation requirements of the EU AI Act Annex IV by composing PROV-O, DCAT, MLSchema, AIRO, and VAIR into a domain-specific application profile. Its methodology extends the NeOn-GPT pipeline [10] with LLM-assisted alignment to reference ontologies and iterative competency-question coverage evaluation. Whereas AIDOC-AP targets regulatory compliance for AI systems (modelling system architecture, risk management, and conformity documentation), RDIP targets the broader research lifecycle across disciplines. The two profiles share a design philosophy: both reuse established vocabularies rather than introducing new ones, and both employ competency questions for validation. However, RDIP focuses on project-centric provenance (activities, datasets, software, methods, people) rather than system-centric compliance.

Overall, the landscape is populated by powerful but specialised ontologies: (i) foundational models for people, publications, datasets, and provenance, (ii) management layer models for data stewardship and planning, and (iii) deep, component-specific ontologies for software or FAIR resources. A critical gap remains for a holistic, project-centric model that integrates these components into a single, coherent knowledge layer. The RDIP integration profile addresses this gap by providing a synthetic model that is explicitly project-centric and process-oriented, and that is purpose-built to serve as a FAIR-compliant schema for AI-driven knowledge extraction and research lifecycle intelligence.

2.4 Metadata Standards and Scholarly Infrastructure

Beyond ontologies, a mature ecosystem of metadata standards underpins FAIR research data management. CERIF [24] provides the EU-recommended CRIS standard; re3data [19] catalogues research data repositories; and OpenAIRE [21] aggregates scholarly metadata using Dublin Core [9], DataCite [7], and CERIF-XML. The Open Research Knowledge Graph (ORKG) [3] represents research contributions as machine-actionable knowledge graphs. At the artefact level, Dublin Core, DataCite, Schema.org, and RO-Crate [25] provide widely adopted baselines, while FAIRsharing [22] curates the broader standards landscape. RDIP complements these infrastructures by offering a project-level provenance model that can be linked to CERIF and OpenAIRE records, expressed as RO-Crate packages, and registered in re3data and FAIRsharing.

Across the four layers (project identifiers, foundational ontologies, process-oriented profiles, and metadata standards) no existing approach provides a project-centric, cross-domain integration layer for research lifecycle provenance. This gap is explicitly addressed by RDIP.

3 Design and Specification of RDIP

The RDIP integration profile was developed following the NeOn [26] methodology, instantiated as a project-centric, process-oriented "knowledge layer" model. The modelling process was guided by expert review from specialists in semantic web and research data management.

3.1 Competency Questions

Table 1 presents eleven competency questions (CQs), organised into four categories. These CQs were formulated from observed requirements in research data management practice: (i) Project-structure queries establish boundaries and responsibility, (ii) Artefact queries identify the software, datasets, and methods involved, (iii) execution-context queries that capture how activities were run, and (iv) Output & Dependency queries that trace results back to their constituent components. This formulation extends the model to category reproducibility (CQ7-CQ9) and dataset-lineage category (CQ10-CQ11), reflecting current practice in computational science and the requirements of open-science workflows. CQ7–CQ11 in particular align with reporting requirements established by the ML Reproducibility Checklist [20] and ACM artifact review policy [2].

3.2 Design Rationale

Traditional provenance models (e.g., PROV-O) are activity-centric, modelling research as an unbounded chain of events. While theoretically complete, such chains are difficult to query for aggregation. By reifying the `rdip:ResearchProject` as the central hub, RDIP imposes a boundary on the graph, enabling data

Table 1: Competency questions driving RDIP's design, organised by category.

Category	CQ	Question
Project Structure	CQ2	What is the ordered, typed activity sequence per project?
	CQ3	Who participated in each project?
	CQ6	Which organisation led each project?
Artefacts	CQ1	What software (and version) was used in each activity?
	CQ4	Which datasets were used or produced by each activity?
	CQ5	What methods were employed in each activity?
Execution Context	CQ7	What hyperparameters and random seeds were used?
	CQ8	What computing environment was used for each activity?
Outputs & Dependencies	CQ9	What evaluation results were produced, on which split?
	CQ10	What is the derivation lineage of each dataset?
	CQ11	What software dependencies does each tool have?

stewards and AI agents to group all related software, datasets, and activities under a single context. This structure is specifically optimised to serve as a target schema for automated extraction systems, ensuring that extracted entities always have a defined parent context.

3.3 The Core RDIP Model

The core of RDIP is the `rdip:` namespace, a small set of classes and properties designed to bridge conceptual gaps between the foundational standards. The `rdip:ResearchProject` class acts as a central hub that connects people (through roles), research activities, datasets, software, methods, and publications. Figure 1 provides an overview of the integration profile, showing how RDIP composes external vocabularies around the central `rdip:ResearchProject` hub. RDIP classes and classes reused from external vocabularies (VIVO, DCAT, BIBO) are distinguished by colour as shown in the legend. The dashed inner box for `rdip:RandomSeed` indicates an `rdfs:subClassOf` relation to `rdip:Parameter`. The nine `rdip:ResearchActivity` subclasses corresponding to research lifecycle stages are listed inline. The `usedDataset/generatesDataset` bidirectional arrow represents two distinct properties from Activity to Dataset (input and output respectively).

A `rdip:ResearchProject` is the central hub, linked to one or more `rdip:ResearchActivity` instances with `rdip:hasActivity` (and its declared inverse `rdip:isPartOfProject`). Activities may use existing datasets, software, and methods (`rdip:usedDataset`, `rdip:usedSoftware`, `rdip:usedMethod`) and generate new datasets, software artefacts, and publications (`rdip:generatesDataset`, `rdip:generatesSoftware`, `rdip:generatesPublication`). Projects are connected to people and organisations through `rdip:hasParticipant`

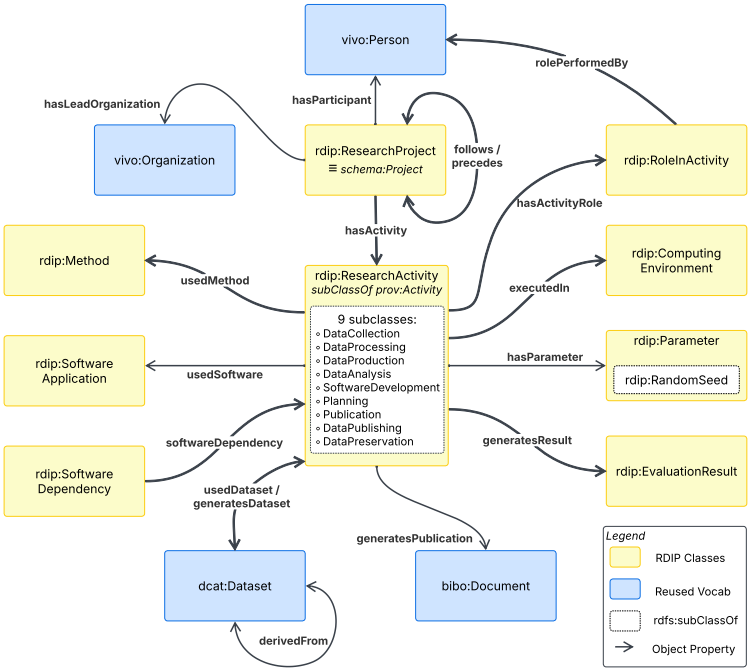

Fig. 1: The core RDIP integration profile.

and `rdip:hasLeadOrganization`. Activities are linked to instances of `rdip:RoleInActivity`, which are in turn `rdip:rolePerformedBy` a `vivo:Person`, capturing the responsibility pattern. Nine activity subclasses capture the research lifecycle stages shown in Figure 1. For full PROV-O compliance, RDIP additionally provides the qualified-association pattern using `rdip:ActivityAssociation` $\sqsubseteq$ `prov:Association`. An activity attributes an agent and a role through a single qualified node.

```
:trainModel a rdip:DataAnalysisActivity ;
    prov:qualifiedAssociation [
        a rdip:ActivityAssociation ;
        prov:agent :alice ;
        prov:hadRole :role_pi ;
        rdip:roleLabel "Principal Investigator" ]
```

Here, the `:trainModel` activity attributes Alice as its agent through a reified rdip:ActivityAssociation node rather than a direct property, allowing her role (Principal Investigator) to be captured alongside the agent link itself. This pattern is what makes role-qualified provenance queries (e.g., "find all PIs of data-analysis activities") expressible in a single SPARQL traversal, which the simpler `hasActivityRole` chain cannot support.

The `hasActivityRole` chain is retained for backward compatibility. Beyond agent attribution, activity execution order is captured by `rdip:precedes` and its inverse `rdip:follows` $\sqsubseteq$ `prov:wasInformedBy`, which turn the activity set into a directed acyclic workflow graph and make CQ2's typed activity sequence answerable in a single SPARQL property path. Table 2 summarises the core classes and properties, including their integration with foundational vocabularies.

Table 2: RDIP: resource counts and relationships with foundational vocabularies.

(a) Resource counts

Component	**Count**
RDIP classes	21
RDIP object properties	25
RDIP datatype properties	44
`owl:equivalentClass` axioms to Schema.org	2
`rdfs:subClassOf` axioms to PROV-O, Schema.org	15
`rdfs:subPropertyOf` axioms to PROV-O, Schema.org, DCAT, Dublin Core, CiTO	30

(b) Property integration with foundational vocabularies (selected)

RDIP property	**Inherits from**
`usedDataset`, `usedSoftware`, `usedMethod`	`prov:used`
`generatesDataset`, `generatesSoftware`, `generatesPublication`, `generatesResult`	`prov:generated`
`derivedFrom`	`prov:wasDerivedFrom`
`isVersionOf`	`prov:wasRevisionOf`, `dcterms:isVersionOf`
`follows`	`prov:wasInformedBy`
`hasParticipant`	`schema:contributor`
`hasLeadOrganization`	`schema:sponsor`
`citesDataset`	`cito:citesAsDataSource`
`identifier`	`dcterms:identifier`, `schema:identifier`
`activityStart` / `activityEnd`	`schema:startDate` + `prov:startedAtTime` / `schema:endDate` + `prov:endedAtTime`

3.4 Reproducibility Constructs

RDIP defines six reproducibility constructs: five classes namely `rdip:Parameter`, `rdip:RandomSeed`, `rdip:ComputingEnvironment` (with `rdip:EnvironmentSpec`), `rdip:EvaluationResult`, and `rdip:SoftwareDependency`, and one property `rdip:derivedFrom`. The selection of these six constructs is grounded in

established reproducibility frameworks. The ML Reproducibility Checklist [20], adopted by NeurIPS and other major venues, requires explicit reporting of hyperparameters, random seeds, computing infrastructure, and evaluation metrics with dataset splits which directly correspond to RDIP's Parameter, RandomSeed, ComputingEnvironment, and EvaluationResult classes. The ACM Artifact Review and Badging policy [2] similarly identifies software dependencies, execution environment, and reproducible computational results as the core artifact components, motivating SoftwareDependency and reinforcing ComputingEnvironment. Gundersen and Kjensmo's taxonomy of AI reproducibility factors [12] groups these dimensions into method, data, and experiment factors which is also captured through RDIP's activity, dataset, and result classes. `ComputingEnvironment` captures host metadata (GPU, CUDA, OS); for byte-level reproducibility, `rdip:EnvironmentSpec` records a container image digest or environment-file URI, supporting reconstruction without out-of-band context. Table 3 summarises the constructs and their mapping to competency questions, grounded in the ML Reproducibility Checklist [20], ACM Artifact Review [2], and Gundersen & Kjensmo's taxonomy [12].

Table 3: Reproducibility constructs and their corresponding competency questions.

Construct	**Class**	**Key Properties**	**CQ**
Hyperparameters	`rdip:Parameter`	`parameterName`, `parameterValue`	CQ7
Random seeds	`rdip:RandomSeed`	`parameterValue`	CQ7
Environment	`rdip:ComputingEnvironment` +`rdip:EnvironmentSpec`	`gpuModel`, `cudaVersion`, `osVersion`; `specType`, `specUri`, `imageDigest`	CQ8
Results	`rdip:EvaluationResult`	`metricName`, `metricValue`, `splitLabel`	CQ9
Lineage	`rdip:derivedFrom`	(Derived Dataset → Source Dataset)	CQ10
Dependencies	`rdip:SoftwareDependency`	`dependencyName`, `dependencyVersion`	CQ11

3.5 FAIR Alignment

RDIP supports the FAIR principles [29] across its layers. Findability is enabled by persistent identifiers (RAiD, DOIs, ORCID, ROR) attached to all core entities via `dcterms:identifier`, with the project hub aggregating descriptive metadata for CQ1–CQ3. Accessibility is supported through `dcat:landingPage` and `rdip:accessLevel` on datasets, with provenance metadata remaining accessible even when payload data are restricted. Interoperability comes from layering over VIVO, DCAT, BIBO, PROV-O, and CiTO, with key classes mapped to Schema.org for JSON-LD exchange. Reusability is enabled by the explicit Dataset → Activity → Method → Project pattern, combined with version, identifier, and role properties supporting CQ5–CQ11. Schema.org enters the profile structurally: `rdip:ResearchProject` ≡ `schema:Project`, with `rdip:Parameter`,

`rdip:EvaluationResult`, and `rdip:Method` aligned through `rdfs:subClassOf` to `schema:PropertyValue`, `schema:Observation`, and `schema:CreativeWork` respectively. This makes RDIP graphs JSON-LD interoperable with Google Dataset Search, schema.org validators, and other Schema.org-aware consumers without conversion.

4 Evaluation

We evaluate RDIP through four complementary analyses: a baseline showing DCAT and PROV-O cannot answer project-centric reproducibility queries; a twelve-paper gold knowledge graph resolving all eleven competency questions; a schema-coverage analysis across 95 papers; and a controlled LLM-extraction experiment over three prompt conditions.

4.1 DCAT+PROV-O Baseline Analysis

To address the question of why existing vocabularies are insufficient, we formulated five SPARQL queries probing key competency questions and evaluated whether they could be answered using only DCAT and PROV-O (Table 4). The queries exercising CQ1, and CQ4 (software-to-dataset coordination), CQ2 (typed activity chain), CQ7, and CQ8 (parameters and environment), and CQ9, and CQ10 (result-to-raw-data lineage) cannot be answered using DCAT+PROV-O constructs alone, while the CQ3 query (person and role) returns only partial results due to the absence of a role vocabulary in PROV-O. This confirms the need for a coordination layer for project-centric reproducibility queries. The full SPARQL queries are available in the project repository.

Table 4: Provenance queries against DCAT+PROV-O versus RDIP.

Query	Competency Queries	DCAT+ PROV-O	RDIP	Baseline failure reason
Q1	CQ1, CQ4	×	✓	`prov:used` is untyped; cannot distinguish software inputs from dataset inputs
Q2	CQ2	×	✓	No activity subclasses; no project boundary
Q3	CQ3	○	✓	No role vocabulary in PROV-O
Q4	CQ7, CQ8	×	✓	No vocabulary for parameters or environments in any standard
Q5	CQ9, CQ10	×	✓	No `EvaluationResult` class; no metric-to-dataset traversal

Remarks: ✓ = Can Answer; ○ = Can Partially Answer; × = Cannot answer.

4.2 Gold Dataset and Competency Question Analysis

Corpus Construction. We assembled a corpus of 95 machine-learning research papers, each with an open-access arXiv PDF and a public GitHub repository. Papers were drawn from an existing reproducibility study [1] and include pre-annotated indicators: Docker support (35%), conda environment (27%), requirements.txt (72%), random seeds in code (35%), and a reproducibility-tier classification. The original study assembled 96

papers; we excluded one due to an inaccessible PDF, yielding 95. Papers were classified into two tiers based on their reproducibility indicators: a full tier (26 papers with five or more reproducibility features, including seeds and containerisation) and a build-only tier (69 papers with basic build infrastructure but fewer reproducibility features).

We partition this corpus into two annotated subsets following standard practice in scientific information extraction [5]: a gold dataset of 12 exhaustively annotated papers, used for the LLM extraction experiment and instantiated as a knowledge graph for competency-question evaluation; and a silver dataset of the remaining 83 papers, used for schema coverage analysis. The distinction reflects annotation depth, not entity quality: every paper in both subsets was reviewed by the domain expert.

Gold Dataset. The 12 gold papers were selected from the full tier to maximise diversity across machine-learning subfields (NLP, computer vision, reinforcement learning, recommender systems, scientific computing, medical imaging, knowledge graph embeddings, digital pathology), reproducibility features, and institutional origin. For each paper, the LLM (Gemini 3 Flash, gemini-3-flash-preview-12-2025) received the paper text together with automatically extracted repository metadata (dependencies from requirements.txt, Docker configuration, random seed values from source code) and produced a structured JSON annotation. The domain expert then exhaustively reviewed and corrected every annotation against the source paper and code repository, resolving all uncertain items. Each gold paper received approximately 30 minutes of thorough verification covering entity completeness, relation correctness, and adherence to the RDIP schema.

Silver Dataset. The remaining 83 papers were processed through the same LLM-assisted pipeline, but the domain expert performed a lighter verification pass of approximately 15 minutes per paper. This verification confirmed schema-level mappings: extracted entities were correctly typed under RDIP classes and relations conformed to the property definitions. However, it did not exhaustively check every entity instance against the source. The silver dataset is sufficient for schema coverage analysis (which asks, does this paper contain information mappable to RDIP class X?) but is not used for the LLM extraction experiment, which requires entity-level ground truth.

All prompts, LLM outputs, repository metadata, and verification records for both datasets are published in the repository to enable independent replication.

Gold Knowledge Graph Statistics The gold dataset was merged into a single RDIP-compliant knowledge graph[5], hereafter the *gold knowledge graph*, comprising 344-triples and 80 resolved relations.

Competency-question coverage All eleven competency questions (CQ1–CQ11) were executed as SPARQL queries against the merged gold knowledge graph in Protégé. Beyond non-emptiness, the domain expert manually verified that each query's returned bindings correctly answered its CQ against the source papers; all eleven CQs returned correct, non-empty results, confirming that RDIP can represent and retrieve the information required by each CQ across twelve real papers from diverse machine-learning subfields.

[5] `https://github.com/open-rdip/rdip-ontology/blob/main/evaluation/validation/combined_gold.ttl`

Table 5: Corpus, gold dataset, and silver dataset statistics.

Metric	Gold dataset	Silver dataset
Papers	12	83
Tier composition	full (12)	full (14) + build-only (69)
Annotation depth	exhaustive (∼30 min)	schema-level (∼15 min)
Used for	LLM experiment, gold KG, CQs	Schema coverage analysis
Gold knowledge graph (materialised from gold dataset)		
Total triples	344	—
Resolved relations	80	—
Avg triples/paper	29	—
Distinct entity types	10	—
CQs answered	11/11	—

4.3 Schema Coverage Analysis

To assess RDIP's applicability at scale, we performed schema-level coverage analysis across the full corpus (gold and silver datasets combined, n = 95). For each RDIP class and relationship, the domain expert recorded, by manual inspection of the paper text and repository, whether the paper contained at least one item of information instantiable under that concept. Coverage for a concept is the percentage of papers in which it was present. Table 6 presents the results stratified by reproducibility tier.

Table 6: RDIP schema coverage across 95 papers, by reproducibility tier.

RDIP Classes & Properties	build only tier (n=69)	full tier (n=26)	Overall (n=95)
`SoftwareApplication`	99%	100%	99%
`SoftwareVersion`	91%	100%	94%
`Dataset`	94%	100%	96%
`Method`	100%	100%	100%
`Activity`	100%	100%	100%
`Person`	100%	100%	100%
`Organization`	100%	100%	100%
`EvaluationResult`	65%	100%	75%
`Parameter`	51%	100%	64%
`RandomSeed`	58%	65%	60%
`ComputingEnvironment`	45%	96%	59%
`derivedFrom`	19%	15%	18%

Core RDIP classes (`Software`, `Dataset`, `Method`, `Activity`, `Person`, `Organisation`) achieve ≥ 94% coverage, confirming the model's broad applicability. Reproducibility-oriented constructs exhibit a clear tier effect: full-tier papers achieve near 100% coverage for Parameters, EvaluationResults, and ComputingEnvironment, while build-only papers show substantially lower rates. This pattern supports RDIP's design rationale i.e., the schema captures reproducibility metadata when documented, and the coverage differential highlights where documentation practices can improve. The low coverage of `derivedFrom` (18%) likely reflects two compounding factors: a genuine gap in

Table 7: LLM experiment: three prompt conditions.

Condition	Schema	Description
C1	Generic	Providing entity names only; no vocabulary definitions
C2	DCAT+PROV-O	Providing standard class and property definitions
C3	RDIP	Providing full RDIP class hierarchy and relationships

Table 8: LLM extraction results across different conditions and entity types.

(a) Aggregate results (Entity and Relation F1)

Condition	Entity F1	Rel F1	Rel TPs
C1 Generic	0.502	0.000	0 / 93
C2 DCAT+PROV-O	**0.570**	0.046	4 / 93
C3 RDIP	0.495	**0.099**	**10 / 93**

(b) Per-entity-type F1 breakdown under relaxed evaluation

Entity Type	C1	C2	C3	C3–C1
Parameter	0.435	0.455	**0.547**	+0.112
EvaluationResult	0.237	**0.296**	0.271	+0.034
Dataset	**0.703**	0.696	0.688	-0.015
Method	0.494	**0.629**	0.490	-0.004
Person	0.864	**0.947**	0.800	-0.064
Software	**0.232**	0.198	0.222	-0.010

scholarly reporting, since most papers do not explicitly document dataset derivation chains, and implicit derivation relationships described narratively rather than through explicit versioning statements, which our present analysis does not attempt to capture. Distinguishing these two explanations is left to future work.

4.4 LLM Extraction Experiment

Design. To evaluate RDIP's utility as a target schema for automated metadata extraction, we conducted a controlled experiment comprising three prompt conditions applied to the gold dataset. Table 7 further describes these three prompt conditions. This design provides each condition with comparable structural depth, the only variable is the vocabulary itself, ensuring a fair comparison.

Each paper was processed independently in a fresh Gemini 3 Flash session using only the paper PDF; no repository metadata was provided to any condition. For scoring, we created a paper-only version of the gold standard where repository-derived entities (e.g., exact dependency versions from requirements.txt, random seed values from source code) were excluded (the full excluded set is enumerated in the repository), as this information is unavailable to extraction prompts operating on paper text alone. This follows best practice in extraction evaluation [5].

Scoring protocols. We report results under three protocols. For entities, we use strict matching (exact normalised string match) and relaxed matching (substring containment or token-overlap ≥50%). For relations, we additionally apply a predicate-normalised protocol that maps PROV-O predicates to RDIP equivalents (e.g., `prov:used` → `rdip:usedSoftware`), isolating vocabulary alignment from extraction quality. Full protocol details are in the project repository.

Results. Table 8 presents aggregate results and per-entity-type breakdowns. Following best practice in information extraction [5], we report both F1 and raw true-positive counts, which we read as a preliminary demonstration given the small absolute counts.

5 Discussion and Limitations

The four evaluation analyses converge on a consistent picture: existing vocabularies cannot answer project-centric reproducibility questions on their own, RDIP's core classes apply broadly across the literature, and RDIP shows early evidence of typed relation extraction that simpler vocabularies miss.

5.1 Why Existing Vocabularies Fall Short

The five-query baseline in Section 4.1 isolates a specific failure mode: not that DCAT or PROV-O lack expressiveness in absolute terms, but that they lack the coordination layer required to bind people, software, datasets, and activities into a single project context. Q2's failure ("typed activity chain") is the cleanest illustration: PROV-O has `prov:Activity` but no semantic typology of research activities, so a SPARQL query over PROV-O alone cannot reconstruct the ordered, typed sequence of stages constituting a research lifecycle. RDIP's contribution is precisely this missing coordination layer.

5.2 Vocabulary Trade-offs in LLM Extraction

The aggregate entity F1 results in Table 8 expose an instructive trade-off. DCAT+PROV-O achieves the highest entity F1 (0.570) because its less constrained vocabulary admits more matches under token-overlap evaluation, while RDIP's typed classes reject ambiguous matches. This trade-off favours RDIP for the use case that motivates the work: knowledge graphs are evaluated by typed relations, not entity lists. On this dimension RDIP leads on small counts (Table 8a), and the per-entity deltas (Table 8b) indicate its gains concentrate on the reproducibility-critical types (`rdip:Parameter`, `rdip:EvaluationResult`) absent from DCAT and PROV-O.

5.3 Implications for AI-driven Metadata Creation

As framed in Section 1, RDIP serves as a target schema rather than an extraction method: it does not specify how an LLM should extract metadata, but constrains what counts as a valid output. This separation matters. The schema's typed structure provides LLMs with explicit extraction targets. The `rdip:SoftwareApplication` class with its version and identifier properties allows distinguishing software instances from dataset identifiers, and the explicit pattern: Dataset → Activity → Software/Method → Project enables extracting workflows rather than isolated mentions. RDIP thus provides a concrete basis

for evaluating automated graph population against CQ1–CQ11 across future LLMs and extraction techniques, independent of which model or prompting strategy is used. This positions RDIP within the broader trend of schema-guided extraction, alongside application profiles such as DCAT-AP for data portals and AIDOC-AP for AI system documentation [18]. Like those profiles, RDIP composes existing vocabularies under a coordination layer rather than introducing competing concepts; unlike them, it targets the research lifecycle across disciplines rather than a specific regulatory or distribution context.

5.4 Limitations

This study consists of three main limitations. First, the LLM extraction experiment uses a single model. Results may differ for other LLMs or for fine-tuned extractors. We mitigate this partially by publishing all prompts and outputs, enabling reproduction with alternative models. Second, annotation was performed by a single domain expert, so we cannot report inter-annotator agreement. The most ambiguous cases (`rdip:Method` vs. `rdip:SoftwareApplication`; activity-boundary delimitation) were resolved via the RDIP class definitions. We mitigate further through tiered verification and published records, though a multi-annotator study remains open. Third, the corpus consists exclusively of machine-learning papers, a domain where reproducibility metadata is unusually rich. Schema applicability in domains with sparser computational artefacts (humanities, qualitative social science) requires separate validation.

6 Conclusion

We have presented RDIP, a project-centric integration profile for FAIR research lifecycle provenance. By composing VIVO, DCAT, BIBO, PROV-O, and CiTO into a coordination layer anchored by `rdip:ResearchProject`, RDIP aims to support provenance queries that no single existing vocabulary addresses on its own, and adds reproducibility constructs largely absent from existing standards. Our evaluation on 95 machine-learning papers suggests broad applicability ($\geq$94% for core classes), all eleven competency questions resolve against a 344-triple gold knowledge graph, and the LLM experiment offers early support for RDIP as a target schema for relation extraction.

Future work proceeds along two tracks. First, we will build out the RDI Platform around RDIP, integrating LLM-based extraction, knowledge graph storage, and FAIR-aligned discovery over the schema layer presented here. Second, we will advance the integration profile itself: connecting it to scholarly infrastructures such as OpenAIRE and CERIF, expressing RDIP knowledge graphs as RO-Crate packages, and adding SHACL constraints for automated validation. We also plan to extend the evaluation to cross-domain corpora.

Disclosure of Interests. The authors have no relevant financial or non-financial interests to disclose.

References

1. Acharya, S., Anutariya, C., Minamiyama, Y., Takeda, H., Wuwongse, V.: A reproducibility corpus of 95 machine-learning papers with github repositories (2026). https://doi.org/10.5281/zenodo.19919042
2. Association for Computing Machinery: Artifact review and badging — version 2.0. https://www.acm.org/publications/policies/artifact-review-and-badging-current (2020)
3. Auer, S., Mann, S.: Towards an open research knowledge graph. The Serials Librarian **76**(1–4), 35–41 (2019). https://doi.org/10.1080/0361526X.2019.1540272
4. Australian Research Data Commons: Research activity identifier (raid). https://raid.org/what-is-raid (2025)
5. Dagdelen, J., Dunn, A., Lee, S., Walker, N., Rosen, A.S., Ceder, G., Persson, K.A., Jain, A.: Structured information extraction from scientific text with large language models. Nature Communications **15**(1), 1494 (2024). https://doi.org/10.1038/s41467-024-45563-x
6. D'Arcus, B., Giasson, F.: Bibliographic ontology (bibo). https://www.dublincore.org/specifications/bibo/ (2016)
7. DataCite Metadata Working Group: Datacite metadata schema for the publication and citation of research data and other research outputs, version 4.6. Tech. rep., DataCite e.V. (2024). https://doi.org/10.14454/mzv1-5b55
8. Dataset Exchange Working Group, W3C: Data catalog vocabulary (dcat) – version 3. https://www.w3.org/TR/vocab-dcat-3/ (2024), w3C Recommendation
9. DCMI Usage Board: Dublin core metadata element set, version 1.1. https://www.dublincore.org/specifications/dublin-core/dces/ (2012)
10. Fathallah, N., Das, A., De Giorgis, S., Poltronieri, A., Haase, P., Kovriguina, L.: NeOn-GPT: A large language model-powered pipeline for ontology learning. In: The Semantic Web: ESWC 2024 Satellite Events. Lecture Notes in Computer Science, vol. 15344, pp. 36–50. Springer (2024)
11. Garijo, D.: Widoco: a wizard for documenting ontologies. In: International Semantic Web Conference. pp. 94–102. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68204-4_9, http://dgarijo.com/papers/widoco-iswc2017.pdf
12. Gundersen, O.E., Kjensmo, S.: State of the art: Reproducibility in artificial intelligence. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018). https://doi.org/10.1609/aaai.v32i1.11503
13. Khadim, A., Perumal, V., Kaliyaperumal, B.: Project management ontology (PMO): A semantic model for representing tasks, milestones and resources in project planning. Journal of Web Semantics **78**, 100759 (2023). https://doi.org/10.1016/j.websem.2023.100759
14. Li, Y., et al.: A survey of graph meets large language model: Progress and future directions. In: Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI) (2024)
15. Magagna, B., Schultes, E., Fouilloux, A., Burger, G., Devriendt, D., Bramley, R., Kuhn, T., Moreira, J.L.R., de Santos, L.O.B., Pires, L.F.: Ontological analysis of FAIR supporting resources in the FIP ecosystem. In: Proceedings of FOIS 2024: FAIR Principles for Ontologies and Metadata in Knowledge Management (2024)
16. Malone, J., Brown, A., Lister, A.L., Ison, J., Hull, D., Parkinson, H., Stevens, R.: The software ontology (SWO): a resource for reproducibility in biomedical data analysis, curation and digital preservation. Journal of Biomedical Semantics **5**(1), 25 (2014). https://doi.org/10.1186/2041-1480-5-25

17. Minamiyama, Y., Hayashi, M., Fujiwara, I., Onami, J.i., Yokoyama, S., Komiyama, Y., Yamaji, K.: Toward the development of NII RDC application profile using ontology technology. In: Proceedings of the Conference on Research Data Infrastructure (CoRDI 2023) (2023). https://doi.org/10.52825/cordi.v1i.260
18. Neumaier, S., Dam, T., Kovac, F.: AIDOC-AP: An application profile for technical documentation of AI systems. https://w3id.org/aidoc-ap (2026)
19. Pampel, H., Vierkant, P., Scholze, F., Bertelmann, R., Kindling, M., Klump, J., Goebelbecker, H.J., Gundlach, J., Schirmbacher, P., Dierolf, U.: Making research data repositories visible: The re3data.org registry. PLOS ONE **8**(11), e78080 (2013). https://doi.org/10.1371/journal.pone.0078080
20. Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché Buc, F., Fox, E., Larochelle, H.: Improving reproducibility in machine learning research: A report from the NeurIPS 2019 reproducibility program. Journal of Machine Learning Research **22**(164), 1–20 (2021)
21. Príncipe, P., Rettberg, N., Rodrigues, E., Elbæk, M.K., Schirrwagen, J., Houssos, N., Jörg, B.: OpenAIRE guidelines: Supporting interoperability for literature repositories, data archives and CRIS. Procedia Computer Science **33**, 92–94 (2014). https://doi.org/10.1016/j.procs.2014.06.015
22. Sansone, S.A., McQuilton, P., Rocca-Serra, P., Gonzalez-Beltran, A., Izzo, M., Lister, A.L., Thurston, M., FAIRsharing Community: FAIRsharing as a community approach to standards, repositories and policies. Nature Biotechnology **37**(4), 358–367 (2019). https://doi.org/10.1038/s41587-019-0080-8
23. Shotton, D., Peroni, S., et al.: CiTO, the citation typing ontology. https://sparontologies.github.io/cito/current/cito.html (2018)
24. Simons, E.: euroCRIS and CERIF: The importance of an international standard metadata model for research information. In: Research Information Systems: Integration for Open Access to Scientific Outputs. pp. 7–16. Slovak Centre of Scientific and Technical Information, Bratislava (2014)
25. Soiland-Reyes, S., Sefton, P., Crosas, M., Castro, L.J., Coppens, F., Fernández, J.M., Garijo, D., Grüning, B., La Rosa, M., Leo, S., Ó Carragáin, E., Portier, M., Trisovic, A., Groth, P., Goble, C.: Packaging research artefacts with RO-Crate. Data Science **5**(2), 97–138 (2022). https://doi.org/10.3233/DS-210053
26. Suárez-Figueroa, M.C., Gómez-Pérez, A., Fernández-López, M.: The NeOn methodology for ontology engineering. In: Suárez-Figueroa, M.C., Gómez-Pérez, A., Motta, E., Gangemi, A. (eds.) Ontology Engineering in a Networked World, pp. 9–34. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24794-1_2
27. Van Nuffelen, B.: DCAT-AP 3.0.1. Application profile specification, SEMIC (2025)
28. VIVO Project: VIVO Integrated Semantic Framework (VIVO-ISF) Ontology. https://bioportal.bioontology.org/ontologies/VIVO-ISF (2015), accessed: 2026-04-27
29. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., et al.: The FAIR guiding principles for scientific data management and stewardship. Scientific Data **3**, 160018 (2016). https://doi.org/10.1038/sdata.2016.18
30. World Wide Web Consortium: PROV-O: The PROV ontology. https://www.w3.org/TR/prov-o/ (2013), w3C Recommendation
31. Xu, D., Zhang, X., Chen, W., Zhang, Y., et al.: Large language models for generative information extraction: A survey. Frontiers of Computer Science (2024). https://doi.org/10.1007/s11704-024-40555-y