Intuitiveness as the Next Stage of Open Data: dataset design and complexity

Sarazin Arthur[a,d,*] and Mourey Mathis[b,d]

[a]Researcher in Design Science, Veltys, France
[b]Researcher in Data science, The Hague University of Applied Sciences, Netherlands
[d]Researcher associated with the UNESCO Chair "AI and Data Science for Society"
[*]Corresponding author : asarazin@veltys.com

Abstract

Intuitive open datasets, which can adapt to the level of data literacy and the needs of the user, represent the next stage of open data. They do not only provide broader access to data, but also unlock the underlying information, knowledge, and reuse potential. In this paper, we present a conceptual meta-design framework for redesigning existing datasets and propose a first instantiation through the *intuitiveness* package and interface. This framework aims at empowering more data users to extract value from open data. Our framework is flexible and can be applied to any new or existing dataset to enhance its intuitiveness. Through this paper, we contribute to the open data community by offering a practical approach to designing and redesigning intuitive datasets and advancing the state of openness.

Keywords: open data; meta-design; data literacy; framework

1 Introduction

1.1 Context and motivation

This conceptual meta-design framework aims at demonstrating how design can assist in producing useful open datasets and related data products. The framework and the associated package and interface (*intuitiveness*) was created to support the open data community in their efforts to broaden the number of data publics (Ruppert, 2012) who use data to support decision-making, create new services and products, or produce innovative information and knowledge (Safarov et al., 2017).

1.2 Problem statement

The conceptual meta-design framework meets the need for a method to design intuitive datasets, that is datasets whose shape can adapt to the data literacy level and the need of

the user. There is a very diverse range of data users whose data literacy and needs differ greatly considering data: some will only look for one piece of information in the dataset while others will use data as a core artifact of a data product they are making. Yet, these diverse needs and data literacy levels have not been considered by open data producers while designing datasets (Dymytrova et al., 2018).

1.3 Contributions

In this paper, we make the following contributions:

1. We propose a conceptual meta-design framework based on five levels of data abstraction, enabling dataset designers to adapt complexity to user needs.
2. We provide a formal definition of dataset complexity and demonstrate mathematically how transitions between abstraction levels reduce complexity.
3. We implement this framework as a Python library (`intuitiveness`) and validate it through a real-world case study with a major international logistics operator.
4. We propose an interface that demonstrate the value of the `intuitiveness` package by making all CSVs files from the french public open data platform data.gouv.fr accessible to the package.
5. We offer practical guidelines for open data platforms to design intuitive datasets that address societal issues such as global warming, health, and public transparency.

1.4 Paper organization

The remainder of this paper is organized as follows. Section 2 reviews related work on data literacy, open data reuse, and human-data interaction. Section 3 presents our conceptual meta-design framework with five levels of abstraction. Section 4 provides the formal complexity analysis. Section 5 describes the implementation and case study. Section 6 discusses implications and limitations. Finally, Section 7 concludes the paper and outlines future work.

2 Related Work

Our work on intuitive datasets draws from several research streams: adaptive data visualization, user modeling and assessment, documentation design, and human-data interaction. We review each in turn, highlighting how existing approaches inform our meta-design framework.

2.1 Adaptive data visualization

A growing body of research addresses the challenge of adapting visualizations to users with varying levels of expertise. Poetzsch et al. (2020) propose a taxonomy for adaptive data visualization in analytics applications, distinguishing between user traits (e.g., statistical expertise, graphical literacy) and user states (e.g., monitoring vs. analysis tasks). Their empirical evaluation reveals that monitoring tasks with higher data complexity receive better suitability ratings, while analysis tasks require richer interactive features such as

filtering, brushing, and drill-down capabilities. This suggests that different abstraction levels may be appropriate for different analytical intents—a principle we formalize in our framework. Steichen et al. (2013) demonstrate that eye gaze data can predict users' current visualization tasks and cognitive abilities, including perceptual speed, visual working memory, and verbal working memory. Their work enables real-time adaptive interventions such as highlighting relevant elements or de-emphasizing non-relevant data to reduce cognitive load. Notably, they find that users with low perceptual speed particularly benefit from adaptive assistance, reinforcing the need for interfaces that can dynamically adjust complexity.

Amyrotos (2021) critiques the prevalent "one-size-fits-all approach" in data visualization tools and proposes a human-centered adaptive visualization framework. This framework incorporates a multi-dimensional user model considering cognitive factors, domain expertise, and task context. A data visualization engine then recommends best-fit visualizations, while an intelligent analytics component continuously refines the user model through interaction tracking. This work underscores the importance of moving beyond static dataset presentations toward dynamic, user-responsive designs.

2.2 Visualization recommenders and progressive disclosure

Selecting appropriate visualizations poses significant challenges for non-expert users. Mutlu et al. (2016) address this through VizRec, a recommender system that suggests personalized visualizations by combining perceptual guidelines with user preferences. The system uses tag vectors to describe visualization content for content-based filtering and quality ratings for collaborative filtering. By reducing combinatorial explosion through perceptual constraints, VizRec alleviates choice overload—a key barrier to data accessibility for users with limited visualization literacy.

Cockburn et al. (2014) examine the broader challenge of supporting novice-to-expert transitions in user interfaces. They document systems like FollowUs, which integrates online tutorials within applications and enables community contributions, leading to higher task completion and lower frustration. The Chronicle system visualizes user workflows via a zoomable timeline, supporting reflection on interaction strategies. These mechanisms for progressive skill development complement our approach of progressive complexity reduction.

2.3 Dashboard Design and User-Centered Challenges

Alhamadi et al. (2022) provide empirical insights into the challenges of user-centered dashboard design. Through interviews with dashboard developers, they identify a significant gap between users' visual literacy and dashboard requirements. Their work categorizes three adaptation mechanisms: *customization* (user-initiated modifications), *personalization* (system-driven adjustments at load time), and *automatic adaptation* (real-time updates based on user models). Developers report implementing practices such as minimal default charts, consistent color schemes, role-tailored filters, and explicit interpretation of visualizations.

Crucially, Alhamadi et al. find that users often struggle not only with visualization com-

plexity but also with understanding data provenance and trusting displayed information. They recommend layered documentation—high-level summaries, in-situ definitions, and links to machine-readable metadata—to address these concerns. This resonates with our goal of designing datasets whose structure can reveal or hide complexity based on user needs.

2.4 User Modeling and Literacy Assessment

Effective adaptation requires accurate assessment of user capabilities. Steichen et al. (2013) pioneer the use of behavioral telemetry for user modeling, demonstrating that gaze patterns can infer cognitive abilities with accuracy significantly above baseline. Prior work they reference shows that mouse click behavior and visualization selections can reveal user expertise and suboptimal usage patterns.

Amyrotos (2021) extends this with reflective analytics and learning-curve modeling to refine user models over time. The goal is a "generic data visualization engine" that renders appropriate visualizations based on data characteristics, user models, and task specifications. Such systems move toward the vision of intuitive datasets that automatically calibrate their presentation to each user's proficiency level.

2.5 Positioning Our Contribution

Existing work focuses predominantly on adapting *visualizations* and *interfaces* to user characteristics. However, comparatively little attention has been paid to adapting the *underlying dataset structure* itself. Our framework addresses this gap by proposing that datasets can be designed with multiple levels of abstraction, enabling not just different visual presentations but fundamentally different data structures optimized for users at different literacy levels.

While adaptive visualization systems adjust how data is shown, our approach adjusts what data is shown and how it is organized. This represents a shift from presentation-layer adaptation to data-layer adaptation. By formalizing complexity levels and reduction mechanisms, we provide dataset designers with a principled method for creating intuitive open datasets that can serve diverse data publics—from citizens seeking a single fact to data scientists building complex products.

3 Conceptual Meta-Design Framework

To create the conceptual meta-design framework we used the design science research methodology (Hevner et al., 2004) and applied the Hierarchical design pattern (Vaishnavi & Kuechler, 2015). This pattern uses the divide and conquer strategy to design a complex system. It designs a system (the conceptual meta-design framework) by decomposing it into subsystems (five conceptual design frameworks to design each of the five levels of abstraction of one dataset), designing each of them before designing the interactions between them.

We also ensured, following the recursive principle of granular computing that secures a high level of human-data interaction (Wilke & Portmann, 2016), that datasets of an

upper level of abstraction could be constructed by human extrapolation of datasets of lower levels of abstraction.

We consider five levels of abstraction for every dataset, hence five conceptual design frameworks:

- **Level 4**: Data made of unlinkable and multi-level datasets
- **Level 3**: Data made of linkable and multi-level datasets
- **Level 2**: Data made of a single dataset with several entities and attributes
- **Level 1**: Data made of a single entity and several attributes, or of a single attribute and several entities
- **Level 0**: Data made of a single entity, a single attribute, and a single value—corresponding to the classical definition of data as a triplet entity-value-attribute (Redman, 1997), also considered as the fundamental information granule.

These five levels were designed according to the five stages of an intuitive process (Csikszentmihalyi, 1997) :

- **Level 0** corresponds to the preparation stage, that is where the user becomes interested in some datum (e.g. he discovers that 65 percent of the population uses chatGPT on a daily basis)
- **Level 1** corresponds to the incubation stage, where the user unfold the datum and makes unusual connections (e.g. the people using chatGPT on a daily basis should be the youngest)
- **Level 2** corresponds to the insight stage, where the users starts to put pieces together (e.g. the users compares the 'age' and the 'daily gpt use' column inside a data table to see if there is a pattern between the two)
- **Level 3** corresponds to the evaluation stage, where the user puts his insight on a trial (e.g, the user challenges his intuition that people using chatGPT must be the youngest by collecting a different dataset with the propability of being a chatGPT daily users depending on your age class)
- **Level 4** corresponds to the elaboration stage where the user will use other logical mechanisms than data aggregation to make sense of different datasets (e.g. the user might look for demographic distributions to see what proportion the youngest represent out of an entire population and then precise that if 65 percent of the population are daily users of chatGPT, the users are mostly the youngest, showing the technology has not spread yet to the entire population)

According to these definitions, data will be qualified as intuitive if it allows to go up the level ladder in order to elaborate from a single value or, in the opposite direction, go down the level ladder to unconver an insight that will trigger curiosity or decision-making.

All in all, this paper sets forth a meta-design framework that will enable data modelers, architects, analysts and data platform editors to put data users in a position of experiencing the "flow", that is a mental state "expected to occur when individuals perceive

greater opportunities for action than they encounter on average in their daily lives, and have skills adequate to engage them" (Nakamura & Csikszentmihalyi, 2014)

3.1 Level 0: The Datum

Data made of a single entity, attribute, and value corresponds in our framework to a Level 0 dataset, with no complexity. It can also be referred to as a "datum." A datum is defined as the smallest informational granule or the fundamental particle of data science. In computer science, the datum is defined as a triple entity-attribute-value. It can be represented by a table with a single cell (see Figure 1).

	Michael
Weight (kg)	45

Figure 1: Table view of Level 0 data

At the lowest abstraction level, Level 0, data has no complexity as no interpretation is required for its comprehension. This type of data is commonly referred to as "raw data." For instance, in our example, Michael weighs 45 kg, which is a fact.

3.2 Level 1: Single Entity or Single Attribute

To move from Level 0 to Level 1 of abstraction, we need to find a common element among several data. It can be a common attribute ('weight'), a common entity ('Michael'), or a common value. As a result, we can obtain a table with a single entity defined by multiple attributes, or a table with a single attribute and multiple entities (see Figure 2). These tables are the primary form of what we call 'data' (plural of 'datum').

	Weight (kg)	Height (cm)
Michael	45	150

	Weight (kg)
Michel	45
Giselle	60

Figure 2: Table view of two Level 1 data

3.3 Level 2: Single Dataset with Multiple Entities and Attributes

To increase complexity and move from Level 1 to Level 2, it logically involves adding attributes and/or entities to the table (see Figure 3).

	Michael	Giselle
Weigh (kg)	45	60
Height (cm)	150	178

Figure 3: Table view of Level 2 dataset

3.4 Level 3: Linkable Multi-Level Datasets

At a higher level of abstraction, data transition from one to many linkable datasets and from one to many levels of entities and/or attributes. This results in linkable datasets, consisting of multiple levels of entities and multiple levels of attributes (see Figure 4).

	Michael		Giselle			
	2015	2016	2015	2016	2015	2016
Weight (kg)	45	45	60	59		
Height (cm)	150	152	178	185		

Figure 4: Enter Caption

In the above example, we have two levels of entities: individuals on one side, and years on the other. We also have one level of attribute: individual physical characteristics (weight and height).

3.5 Level 4: Unlinkable Multi-Level Datasets

At the highest level of abstraction, data transition from definable complexity to undefinable complexity. These are multi-level data tables that cannot be linked based on current scientific knowledge. These multi-level tables are characterized by the fact that their junction cannot be represented in the form of tables, since their level of complexity is indefinable. At best, they could be represented by a disconnected graph.

It has become commonplace to state that there is an ever-increasing amount of data available, which implies an underlying structure of extreme intelligence that can link data

together. This structure would enable the discovery of fascinating knowledge and the creation of innovative applications. In this article, we assume that the reality of newly available data is quite different: there is no apparent structure to link them together, or if it does exist, it is indefinable based on current scientific knowledge. From our conceptual meta-design framework's perspective, all newly available datasets are Level 4 data.

Indeed, the available data, in their vast majority, do not share any attributes or common elements that would allow them to be linked together. For example, the link between daily shopping baskets in supermarkets and monthly water consumption in surrounding households has not been established. It may not exist, but certainly the complexity of the link is indefinable based on current scientific knowledge.

3.6 Summary

We have shown in this section that data can be represented according to five levels of abstraction. Level 0 and Level 4 are purely theoretical in nature, being respectively the domain of machines and human beings. Our main focus is to establish the conceptual framework of intuitive data, which are defined as data whose level of abstraction and complexity can adapt to the data literacy level of the user.

In the following section, we formally demonstrate that transitioning from a higher abstraction level to a lower level decreases the complexity of the data, and we determine to what extent this complexity decreases.

4 Formal Complexity Analysis

4.1 Definition of Dataset Complexity

The complexity of a dataset can be measured by the number of relationships that can be extracted from it. In this framework, we consider that the order of complexity ($C()$) associated with a dataset relates to how fast the complexity increases with the size of the dataset.

Let us consider the derivation of the order of complexity for each level:

Level 0: The complexity of a single data point is equal to zero. There is no complexity associated with a single point of information. The order of complexity is $C(0)$.

Level 1: For a variable or a single vector of values, there is only one way to interpret the data: we look at how all data points compare to each other. Since there is only one way to look at the data, the order of complexity is $C(1)$.

Level 2: For a (single-level) table, we can consider the following: (1) we can interpret each row/column independently; (2) we can combine one row (or column) with one or more rows (or columns) to study the relationship they have.

The overall number of combinations we can make in a table with n rows is: $2^n - 1$. We define the order of complexity as how fast the complexity grows with each new row:

$$2^{n+1} - 1 - (2^n - 1) = 2^{n+1} - 2^n = 2^n(2 - 1) = 2^n \tag{1}$$

The order of complexity is: $C(2^n)$.

Level 3: For a multi-level table, the complexity depends on the number of rows/columns (n) and the number of groups for each level (g). The total number of possible combinations is: $2^{ng} - 1$.

Considering the growth of complexity when adding a new group:

$$2^{n(g+1)} - 1 - (2^{ng} - 1) = 2^{n(g+1)} - 2^{ng} = 2^{ng}(2^n - 1) \tag{2}$$

The order of complexity is: $C(2^{ng}(2^n - 1))$.

Level 4: Unlinkable tables correspond to infinite complexity: $C(\infty)$.

4.2 Complexity Reduction

We examine how to reduce the order of complexity by transforming a higher level into a lower one. We consider single-level reductions and measure the relative reduction:

$$\Delta C = \frac{C_{\text{after}} - C_{\text{before}}}{C_{\text{before}}} \tag{3}$$

4.2.1 Level 4 to Level 3

Going from an infinitely complex dataset to a measurably complex dataset is, by definition, an almost perfect reduction of complexity.

4.2.2 Level 3 to Level 2

For the upper bound:

$$\Delta C = \lim_{g \to \infty} \frac{2^n - 2^{ng}(2^n - 1)}{2^{ng}(2^n - 1)} = -100\% \tag{4}$$

For the lower bound (simplest Level 3: $g = 2$, $n = 2$):

$$\Delta C = \frac{1}{2^{2(2-1)}} - 1 = \frac{1}{4} - 1 = -75\% \tag{5}$$

The reduction is bounded: $\Delta C \in [-100\%; -75\%]$.

4.2.3 Level 2 to Level 1

$$\Delta C = \lim_{n \to \infty} \frac{1 - 2^n}{2^n} = -100\% \tag{6}$$

For the lower bound (simplest Level 2: $n = 2$):

$$\Delta C = \frac{1}{2^2} - 1 = -75\% \tag{7}$$

The reduction is bounded: $\Delta C \in [-100\%; -75\%]$.

4.2.4 Level 1 to Level 0

$$\Delta C = \frac{0 - 1}{1} = -100\% \tag{8}$$

Any complexity reduced to Level 0 corresponds to a 100% reduction.

5 Case Study and Implementation

We implemented this method in a Python library (`intuitiveness`), applied it to a dataset from a major international logistics operator and developed an interface.

5.1 Case study

5.1.1 Problem context

The organization faced an overwhelming amount of metadata on their indicators, coming from different sources and formats, making it difficult to manage their data ecosystem effectively. Their core challenge was: **given these metadata, how to identify which indicators to delete for operational efficiency while maintaining analytical capabilities?** With 8,368 indicators scattered across multiple sources, there was no intuitive way to determine which were essential and which were redundant or obsolete.

5.1.2 The Descent Phase (L4 → L0)

Step 1: L4 → L3 (Graph Construction) We modeled the raw "unlinkable" files into a knowledge graph, transforming a Level 4 dataset into Level 3. The graph revealed 40,279 relationships among 8,368 indicators (48 connections per indicator on average).

Step 2: L3 → L2 (Domain Isolation) We queried the graph to isolate indicators by domain: revenues, volumes, and employees (ETP). This categorical structure provided the first layer of intuitive organization.

Step 3: L2 → L1 (Feature Extraction) We extracted indicator names to analyze naming conventions and identify duplicates.

Step 4: L1 → L0 (Atomic Metric) We derived the atomic metric: *number of revenue indicators*. This precise formulation captures a business diagnostic—an overproduction of indicators—and served as the ground truth for the audit.

5.1.3 The Ascent Phase (L0 → L3)

Step 5: L0 → L1 (Reconstructing the Vector) From the atomic metric, we reconstructed a vector of naming signatures by extracting structural features from each indicator name.

Step 6: L1 → L2 (Initial Classification) We added categories to the indicators:

- `business_objects`: volume, revenue, ETP
- `calculated`: binary flag (raw data vs. calculated metric)
- `weight_flag`, `rse_flag`, `surcharges_flag`

Step 7: L2 → L3 (Analytic Dimensions) We added analytic dimensions:

- `client_segmentation`: which client segments?
- `sales_location`: geographic usage?

- `product_segmentation`: which products?
- `financial_view`: financial perspective?
- `lifecycle_view`: business lifecycle stage?

5.1.4 Results

The **descent** (L4 → L0) moved the organization from chaos to a clear atomic metric. The **ascent** (L0 → L3) produced intuitive Level 3 tables answering the business question.

The Level 3 table reveals clusters of indicators sharing identical analytic dimensions:

Indicator	Object	Client	Location	Product	Financial	Lifecycle
`CA|4p|12caps`	revenue	All	Global	All	operational	current
`CA|4p|11caps`	revenue	All	Global	All	operational	current
`CA|4p|10caps`	revenue	All	Global	All	operational	current

Table 1: Example of redundant indicators sharing all analytic dimensions

These three indicators share **all six analytic dimensions**—candidates for consolidation. By grouping indicators with identical dimension profiles, the organization can:

1. **Identify redundancy clusters**
2. **Select representatives** per cluster
3. **Delete duplicates** with confidence

This demonstrates the power of the **Descent-Ascent cycle**: transforming "data swamps" into "intuitive datasets."

5.2 Implementation : the Intuitivness package

We implemented the framework as `intuitiveness`, an open-source Python package that operationalizes the descent-ascent methodology through three core capabilities: AI-assisted data modelling, semantic domain matching, and interactive navigation.

5.2.1 Adaptive data structures.

The package represents each abstraction level as a typed Python class. Level 4 wraps dictionaries of DataFrames representing disconnected CSV files. Level 3 holds NetworkX graphs or multi-level DataFrames with explicit relationships. Level 2 contains single pandas DataFrames for domain-specific tables. Level 1 stores pandas Series as feature vectors. Level 0 encapsulates scalar values with optional parent references enabling reverse navigation. A unified `Redesigner` class dispatches transitions between adjacent levels, enforcing the constraint that users cannot return to Level 4 once they descend—ensuring commitment to the linking decisions that transform chaos into structure.

5.2.2 AI-powered entity discovery.

The L4→L3 transition—often the most daunting for non-experts—leverages large language models to suggest data models from raw CSVs. The system extracts column names, data types, and sample values, then prompts an LLM (supporting both local models via

Ollama and cloud models via OpenAI) to propose entities, key properties, and relationships. Users review suggestions through an interactive schema visualization before graph construction. For users preferring manual control, a rule-based generator creates star schemas from user-specified entity lists. A three-tier relationship discovery system identifies connections between files: name heuristics (∼5 ms) detect common ID patterns; value overlap analysis (∼100 ms) computes Jaccard similarity on stratified samples; semantic embeddings (∼2 s) find conceptually related columns when exact matches fail.

5.2.3 Semantic domain matching.

The L3→L2 transition employs embedding-based categorization using the multilingual-e5-small model**?**. Users specify target domains in natural language; the system first attempts keyword matching against domain-specific vocabularies, then batch-processes unmatched items through semantic similarity computation. This hybrid approach handles multilingual datasets and accommodates users unfamiliar with precise terminology—a query for "revenus" correctly matches columns labelled "CA" (*chiffre d'affaires*) or "income."

5.2.4 Navigation and persistence.

A `NavigationSession` class enforces the framework's rules: entry exclusively at L4, vertical-only movement, and no return to L4 once departed. The system tracks accumulated outputs at each visited level, enabling export of complete analysis trails. A branching `NavigationTree` structure preserves alternative exploration paths, supporting time-travel to previous decision points. Sessions persist across browser restarts through localStorage serialization with version checking and corruption detection.

5.2.5 Interactive interface.

An accompanying Streamlit application provides guided workflows for each transition: file upload with drag-and-drop, entity discovery with confidence indicators, relationship confirmation with sample matches, domain categorization with keyword/semantic toggles, column extraction with data type handling, and aggregation selection. Ascent-phase forms support enrichment function selection (L0→L1), dimension addition with preview (L1→L2), and drag-and-drop relationship building (L2→L3). A sidebar decision tree visualizes exploration branches; a JSON exporter produces audit trails compatible with visualization tools. The interface supports French and English with runtime language switching.

5.2.6 Data.gouv.fr integration.

To demonstrate real-world applicability, we integrated search capabilities for the French national open data platform. Users search datasets using natural language queries, preview metadata and sample records, then import CSV resources directly into the redesign workflow—applying the full descent-ascent cycle to any of the platform's 45,000+ datasets.

6 Discussion

6.1 Implications for Practice

The conceptual meta-design framework and its implementation as the `intuitiveness` Python package offer several practical pathways for adoption by open data platforms, dataset designers, and data literacy initiatives.

6.1.1 For Open Data Platforms

International open data platforms such as data.gouv.fr, UIS.stat, or the World Bank Open Data can integrate the framework as a "data redesign plugin." Our interface demonstrates this feasibility by making all CSV files from data.gouv.fr accessible through the `intuitiveness` package, allowing users to query datasets in natural language and navigate through abstraction levels. Specifically, platforms could:

- **Expose multiple abstraction levels**: Rather than providing a single download option, platforms could offer users the choice to access data at L0 (atomic metrics for quick facts), L1 (feature vectors for trend analysis), L2 (filtered tables for domain exploration), or L3 (linked multi-level datasets for advanced analytics).
- **Implement guided descent workflows**: Following our Q&A approach, platforms could guide users through progressive complexity reduction, asking questions such as "What is your main indicator of interest?" and "Which domains are relevant to your analysis?"
- **Enable user-driven ascent**: Once users reach L0 ground truth, platforms could support intentional reconstruction with user-specified analytic dimensions, producing datasets tailored to specific audiences or use cases.

6.1.2 For Dataset Designers

Data architects and analysts can apply the framework to redesign existing datasets or design new ones with intuitiveness in mind:

- **Start with L0**: Define the atomic metric(s) that capture the essential truth of the data before adding complexity. This "ground truth first" approach ensures that even the most complex datasets can be traced back to fundamental, interpretable values.
- **Document complexity levels**: For each dataset, explicitly document what L1, L2, and L3 representations exist and how to navigate between them. Our data model validation tools (`validate_data_model`) can verify that transitions are well-defined.
- **Use semantic matching for domain isolation**: The L3→L2 transition employs embedding-based categorization (using models like `intfloat/multilingual-e5-small`) to group data by semantic similarity, not just exact keyword matching. This approach handles multilingual datasets and accommodates users who may not know the precise terminology.

6.1.3 For Data Literacy Initiatives

Educational programs and training initiatives can leverage the framework to scaffold learning:

- **Progressive skill development**: The five abstraction levels align with increasing data literacy competencies. Beginners can start at L0 (understanding single facts) and progressively advance to L3 (working with linked, multi-level structures).
- **Flow-based learning experiences**: Drawing on Csikszentmihalyi's flow theory, the framework positions data users to experience optimal challenge-skill balance at each level, reducing frustration and increasing engagement.
- **Adaptive interfaces**: Following recommendations from the visualization literacy literature (Alhamadi et al., 2022; Amyrotos, 2021), our Streamlit interface implements progressive disclosure, minimal default visualizations, and contextual guidance that adapts to user actions.

6.1.4 Integration with Knowledge Graphs

The framework's L4→L3 transition leverages Neo4j and knowledge graph technologies to transform unlinkable datasets into structured, queryable graphs. Organizations can:

- Use LLM-assisted entity discovery to automatically suggest data models from raw CSV files.
- Map columns to entities rather than entire files, enabling finer-grained graph construction.
- Query the resulting knowledge graph using Cypher to isolate domain-specific subsets for further analysis.

Our case study with the international logistics operator demonstrated that this approach can handle 8,368 indicators across multiple sources, producing actionable redundancy clusters within a single descent-ascent cycle.

6.2 Limitations

While the framework provides a principled approach to designing intuitive datasets, several limitations should be acknowledged.

6.2.1 Theoretical Constraints

- **L4 and L0 as theoretical extremes**: Level 4 (unlinkable datasets) and Level 0 (atomic datums) represent boundary conditions that are more useful as conceptual anchors than practical states. L4 assumes complete disconnection between datasets, which is rarely absolute in practice—some implicit relationships often exist. Similarly, L0's single-value representation may oversimplify nuanced phenomena.

- **Complexity formula assumptions**: The complexity order calculations assume that all possible combinations of rows/columns are equally meaningful. In practice, many combinations may be semantically invalid or analytically irrelevant. The formal bounds (75%–100% reduction per level) represent mathematical upper limits, not guaranteed practical outcomes.

- **Tabular data focus**: The framework was developed primarily for tabular data (CSV files, relational tables). Extension to other data modalities—such as time series, geospatial data, or unstructured text—would require additional abstraction mechanisms.

6.2.2 Package and interface implementation Constraints

- **LLM dependency for entity discovery**: The L4→L3 transition relies on large language models (via Ollama or OpenAI) to suggest entities and relationships. LLM outputs can be inconsistent, requiring manual validation and editing of the generated data model. Users without access to suitable LLMs may find this step challenging.

- **Semantic matching thresholds**: The embedding-based domain categorization (L3→L2) requires threshold tuning. A threshold too low produces false positives; too high yields missed matches. The optimal threshold varies by domain and language, complicating cross-domain generalization.

- **Neo4j infrastructure requirement**: The knowledge graph functionality requires a running Neo4j instance. While Docker deployment simplifies setup, this dependency may be prohibitive for users seeking a lightweight, standalone solution.

- **Performance at scale**: The current implementation processes datasets in memory and executes Cypher queries sequentially. For very large datasets (millions of rows), batch processing, pagination, or distributed computation would be necessary.

6.2.3 Validation Scope

- **Single case study**: The empirical validation is based on a single case study with a logistics operator. While the results are promising (40,279 relationships discovered, actionable redundancy clusters identified), broader validation across diverse domains, dataset sizes, and user populations is needed to confirm generalizability.

- **No user study**: We have not yet conducted formal user studies to measure intuitiveness gains, task completion times, or user satisfaction across different data literacy levels. The claim that the framework produces "intuitive" datasets remains theoretically grounded but empirically untested with end users.

- **Ascent phase less mature**: While the descent phase (L4→L0) is well-developed with comprehensive tooling, the ascent phase (L0→L3) relies on manually specified dimensions and enrichment functions. Automated dimension suggestion based on user intent remains an open research challenge.

6.2.4 Scope Boundaries

- **Data quality assumed**: The framework assumes input data is reasonably clean and well-structured. It does not address data quality issues such as missing values, inconsistent encodings, or schema drift. Preprocessing steps are the user's responsibility.
- **Static snapshot**: The current implementation operates on static dataset snapshots. Real-time or streaming data would require additional mechanisms for incremental updates and complexity recalculation.
- **Single-user workflow**: The framework supports individual users navigating abstraction levels. Collaborative features—such as shared navigation sessions, annotation, or version control of redesigned datasets—are not yet implemented.

7 Conclusion and Future Work

The Data Redesign Method provides a rigorous path out of the "data swamp." By quantifying complexity and enforcing a descent to atomic levels before any ascent, organizations can create datasets that adapt to the data literacy level of their users.

This methodology, implemented as a Python package, can be used by designers, data scientists, and citizens dealing with real-world data. International open data platforms such as UIS.stat or the World Bank Open Data can use it to design data redesign plugins that increase dataset intuitiveness.

7.1 Future Work

[This section requires development with specific future directions.]

Acknowledgements

The authors thank Datactivist and the UNESCO Chair "AI and Data Science for Society" for supporting this research.

Funding Statement

This research was funded by Datactivist and the UNESCO Chair in AI and Data Science for Society through the Dataflow project.

Competing Interests

The authors declare no competing interests.

References

Alhamadi, M., Alghamdi, O., Clinch, S., & Vigo, M. (2022). Data quality, mismatched expectations, and moving requirements: The challenges of user-centred dashboard design. *Proceedings of the Nordic Human-Computer Interaction Conference*, 1–14. 10.1145/3546155.3546708

Amyrotos, C. (2021). Adaptive Visualizations for Enhanced Data Understanding and Interpretation. *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, 291–297. 10.1145/3450613.3459657

Cockburn, A., Gutwin, C., Scarr, J., & Malacria, S. (2014). Supporting Novice to Expert Transitions in User Interfaces. *ACM Computing Surveys*, 47(2), 1–36. 10.1145/2659796

Csikszentmihalyi, M. (1997). Flow and the Psychology of Discovery and Invention. *New York: HarperPerennial*.

Dymytrova, V., Larroche, V., & Paquienséguy, F. (2018). Cadres d'usage des données par des développeurs, des data scientists et des data journalistes [Research report]. Retrieved from `https://hal.science/hal-01730820`

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105.

Mutlu, B., Veas, E., & Trattner, C. (2016). VizRec: Recommending Personalized Visualizations. *ACM Transactions on Interactive Intelligent Systems*, 6(4), 1–39. 10.1145/2983923

Nakamura, J., & Csikszentmihalyi, M. (2014). The concept of flow. In *Flow and the foundations of positive psychology: The collected works of Mihaly Csikszentmihalyi* (pp. 239–263). Springer.

Poetzsch, T., Germanakos, P., & Huestegge, L. (2020). Toward a Taxonomy for Adaptive Data Visualization in Analytics Applications. *Frontiers in Artificial Intelligence*, 3. 10.3389/frai.2020.00009

Redman, T. C. (1997). *Data Quality for the Information Age*. Artech House.

Ruppert, E. (2012). Doing the transparent state: Open government data as performance indicators. In *A World of Indicators* (pp. 51–78). Cambridge University Press.

Safarov, I., Meijer, A., & Grimmelikhuijsen, S. (2017). Utilization of open government data: A systematic literature review. *Information Polity*, 22(1), 1–24.

Steichen, B., Carenini, G., & Conati, C. (2013). User-adaptive information visualization: using eye gaze data to infer visualization tasks and user cognitive abilities. *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, 317–328. 10.1145/2449396.2449439

Vaishnavi, V. K., & Kuechler, W. (2015). *Design Science Research Methods and Patterns*. CRC Press.

Wilke, G., & Portmann, E. (2016). Granular computing as a basis of human–data interaction: A cognitive cities use case. *Granular Computing*, 1, 181–197.