Comments on Themenbaumstruktur

Nghia Duong-Trung

22.03.2025

1 Introduction

The current solution ("Themenbaumstruktur") proposes a method for generating a three-level topic tree ("Themenbaum") for educational content by means of AI-driven prompts. While it addresses some practical considerations (e.g., the choice between a single-prompt workflow and iterative "prompt loops"), it does not fully map onto the IEEE 1484.12.1-2020 Learning Object Metadata (LOM) standard[1][2] [1, 2]. Additionally, data engineers may face uncertainty about whether the proposed solution can be reliably extended or aligned with the structural rigor typically required in large-scale metadata repositories.

Here, I provide a detailed commentary on the approach described in *Themenbaumstruktur* (an attachment outlining a method for AI-based generation of a hierarchical topic tree). It also proposes a revised prompt pattern that adheres more closely to the IEEE 1484.12.1–2020 Learning Object Metadata (LOM) standard. The intended audience is data engineers who are unsure about extending the current method to a rigorous metadata framework.

2 Comments on Potential Issues in *Themenbaumstruktur*

2.1 Limited Use of LOM Metadata Fields

The attachment focuses on building a three-level taxonomy (main categories, subcategories, and further subcategories), but does not systematically reference the nine top-level categories of IEEE LOM:

> `General, Lifecycle, Meta-Metadata, Technical, Educational,`
> `Rights, Relation, Annotation, Classification.`

As a result, key descriptors such as typical learning time or rights statements are not addressed, potentially reducing interoperability with established systems.

2.2 Lack of Explicit Data Type Definitions

The JSON structure in *Themenbaumstruktur* introduces simple fields like ``` ``title'' ``` or ``` ``shorttitle'' ``` without enforcing data types (*LangString*, *Vocabulary*, *DateTime*, etc.). IEEE LOM elements are typically more strictly defined, making

[1] `https://standards.ieee.org/ieee/1484.12.1/7699/`

[2] `https://www.researchgate.net/publication/379104313_Kompendium_Didaktische_Metadaten`

it important to handle items like `language` as valid ISO codes or `status` as enumerated values.

2.3 Risk of Duplicate or Synonymous Categories

While the document acknowledges the possibility of repetitive or near-duplicate topics at different levels, it provides no strict mechanism for preventing such duplication in a large-scale, automated environment. Relying solely on iterative prompts to discover and remove duplicates could still lead to inconsistencies across bigger repositories.

2.4 No Formal Mapping to LOM `Relation` or `Classification`

The hierarchical notion of "subcollections" in the original approach does not map to standard LOM fields such as `classification.taxonPath` or `relation` (`isPartOf`, `requires`, etc.). Without adopting these LOM constructs, further reuse or crosswalks with other metadata standards could become more cumbersome.

3 Revised Prompt Pattern Aligned with IEEE LOM

In order to address the issues above, we propose a more structured, LOM-centric prompt pattern. This pattern instructs the AI to produce *pure JSON* with nested categories that mirror the LOM standard as laid out in IEEE 1484.12.1–2020.

Listing 1: Example Prompt Pattern in JSON Form, Aligned with IEEE LOM

```
You are a helpful AI assistant trained to generate metadata in compliance
     with IEEE 1484.12.1--2020.

Please respond with **pure JSON** (no code fences, no Markdown) for a
    single "learning object"
that includes, at minimum, the following LOM categories and sub-elements:

{
  "general": {
    "identifier": "UniqueID-123",
    "title": "Descriptive title (LangString)",
    "language": "en",
    "description": "Short definition. Relevance. Features. Application.",
    "keyword": ["Keyword1", "Keyword2"]
  },
  "lifecycle": {
    "version": "1.0",
    "status": "draft"
  },
  "technical": {
    "format": "text/html",
    "location": "http://example.com/resource.html"
  },
  "educational": {
    "learningResourceType": ["lecture", "exercise"],
    "interactivityLevel": "medium",
    "typicalLearningTime": "PT2H"
  },
  "rights": {
    "cost": "no",
    "copyrightAndOtherRestrictions": "yes",
    "description": "CC BY-NC 4.0"
  },
  "relation": [
    {
      "kind": "isPartOf",
      "resource": {
        "identifier": "UniqueID-Parent"
      }
    }
  ],
  "classification": [
    {
      "purpose": "discipline",
      "taxonPath": [
        {
          "source": "MyThemenbaum",
          "taxon": [
            {
              "id": "1",
              "entry": "Main Category"
            },
            {
              "id": "1.1",
              "entry": "Subcategory"
            }
          ]
        }
      ]
    }
  ]
}

If you cannot create this structure, return an empty JSON object.

Instructions:
1) Adhere to the specified field names and nesting (LOM hierarchy).
2) Use valid ISO 639-1 codes for "language" (e.g., "en", "de").
3) Acceptable values for "status" in "lifecycle" are ["draft", "final", "
    revised"].
4) Follow the format "PTxHxM" for "typicalLearningTime" (ISO 8601
    durations).
5) For "relation.kind", use LOM-like relationships (e.g., "isPartOf", "
    requires").
6) For "classification", embed hierarchical paths using "taxonPath"
    arrays.
```

3.1 Key Advantages of this Pattern

- **LOM Category Coverage:** Ensures that the AI output covers core LOM categories: `general`, `lifecycle`, `technical`, `educational`, `rights`, `relation`, and `classification`.
- **Data Type Enforcement:** By specifying ISO standards (e.g., `ISO 639-1` for `language` and ISO 8601 duration strings for `typicalLearningTime`), data engineers can reliably parse metadata fields.
- **Hierarchical Encoding:** The `classification.taxonPath` sub-element provides a proper LOM-based mechanism for storing a tree or taxonomy.
- **Relation vs. Classification:** If your topics need cross-linkages, the `relation` element can capture relationships such as "requires" or "isPartOf" more explicitly than a purely nested JSON approach.

4 Compatibility with IEEE 1484.12.1–2020

Note that there are two versions of IEEE 1484.12.1: IEEE 1484.12.1–2002 and IEEE 1484.12.1–2020.

4.1 Unchanged Core Structure

The updated (2020) version of IEEE 1484.12.1 maintains the same overall structure as the earlier standard. The nine top-level categories, their naming, and primary sub-elements remain recognizable and valid. Hence, any solution conforming to the original LOM structure is still fundamentally compatible with the 2020 revision.

4.2 Potential Refinements in the 2020 Revision

The 2020 update introduces finer guidance and best practices. While these additions do not invalidate existing LOM metadata, data engineers may want to:

- Check for any revised vocabularies or recommended enumerations, such as standard `role` types for `Lifecycle.Contribute`.
- Investigate improved coverage of *accessibility* features or specialized fields for emerging use cases.
- Explore crosswalks to other standards (e.g., Dublin Core, xAPI) that the 2020 LOM updates may better support.

4.3 Overall Assessment

Because the core categories and structure were preserved, your revised JSON prompt pattern remains valid for 1484.12.1–2020. At most, you will need to incorporate the standard's newer recommendations on recommended best practices, any updates to sub-element definitions, or extended usage guidelines.

5 Remarks for the Teams

If your primary objective is to build an AI-driven workflow for generating a "Themenbaum" while also ensuring reliable, standards-based metadata, you should:

- **Adopt LOM Categories Directly:** Match the JSON output to LOM's `general`, `technical`, `educational`, `classification`, etc.
- **Use `classification` for Hierarchies:** Encode multi-level topic structures using `taxonPath`.
- **Validate Data Types:** Enforce ISO 639-1 for languages, ISO 8601 for times, and relevant enumerations (e.g., `status`).
- **Leverage `relation` for Dependencies:** If certain topics are part of a larger unit or require another unit, capture that via `kind` fields in `relation`.
- **Iterative Prompting:** For deeper taxonomies, an iterative approach can still be used, but each prompt must produce valid LOM JSON to maintain consistency at scale.

By following the proposed prompt pattern, we can transform the original *Themenbaumstruktur* proposal into a more rigorously structured, LOM-compliant workflow that remains interoperable with other repositories and metadata systems, including those adhering to the 1484.12.1–2020 last version.

References

[1] Ieee standard for learning object metadata. IEEE Std 1484.12.1-2020 pp. 1–50 (2020). https://doi.org/10.1109/IEEESTD.2020.9262118

[2] Oellers, M., Rörtgen, S.: Kompendium: Didaktische Metadaten. : (2024)