LePa: The AIR Conceptual Model

1 Introduction

In this document we define the AIR model developed to capture an ontology of different types of Educational resources (ERs) and their relationships. We will use this conceptual model to describe concepts related to learning in the setting of a course, such as curated sequence of ERs referred to as *learning paths*.

2 Primary Entities

An *Educational Resource* (ER) is defined as any research, teaching or learning "material" used in any learning capacity. A complete list of possible ER types is provided by the *Dublin Core Metadata Initiative* (DCMI), which identifies e.g. datasets, images, interactive resources, software, sound, text etc. as types of resources. The Dublin Core Metadata Initiative (DCMI)[3] is one of the most common metadata frameworks used today due to its wide use in library sciences.

For the purposes of the AIR model, we identify two types of ERs:

1. An *atomic ER* is a single resource (e.g. a text document, a video).
2. A *composite ER* consists of a set of atomic ERs. This set of ERs can consist of one (e.g. a text document) or more (e.g. two text documents and a video) atomic ERs.

All educational resources must have non-empty values for the following properties:

- `identifier` – "An unambiguous reference to the resource within a given context". A unique integer that identifies the ER.
- `title` – "A name given to the resource". The title to be displayed for the ER.
- `type` – "The nature or genre of the resource". The type of composite ER (see Section 2.1) or the type of atomic ER (.txt, .pdf etc.).

Additionally, they must have a non-empty value for at least one of the following properties:

- `description` – A text description for the resource.
- `url` – The URL associated with this ER.

The AIR model also defines two special types of non-ER entities:

1. The start node. The title, type, and description of this node is "start". There is at most 1 start node in a given dataset. The start node has no relationships to other nodes except maybe for a `comesBefore` relationship to exactly one composite ER. There is exactly one composite ER that `comesAfter` the start node.
2. The end node. The title, type, and description of this node is "end". There is at most 1 start node in a given dataset. The end node must have exactly one `comesAfter` relationship to a composite ER and no other relationships.

2.1 AIR Model

The AIR model defines three distinct types of composite ER:

1. An instructional ER (iER) consists of one or more instructional course materials, such as lecture slides, videos, discussions, or any optional materials for the course. An iER is meant to prepare the learner for the next activity.

2. An activity ER (aER) consists of any activity that can be assessed (e.g. homework, quizzes, exams). This may include several atomic ERs.

3. A rubric ER (rER) consists of a set of instructions to inform how an activity is to be assessed [1, 2, 11], as well as any related atomic ERs. A rubric ER also has an optional property called `weight` whose value is the weight of this activity towards the overall course grade, e.g. 0.4 for an activity worth 40% of the final mark.

3 Relationships Among Entities

Relationships involving atomic ERs are derived from the Dublin Core metadata schema. Additional relationships have been specifically defined for the needs of the AIR model at the composite level.

There are three types of relationships (edges) between ERs (nodes). These are: 1) Atomic-Atomic, 2) Atomic-Composite , 3) Composite-Composite.

3.1 Atomic-Atomic ER Relationships

There can be several relationships between atomic ERs.

- A `requires` B: Educational resource A requires educational resource B to support its function, delivery or coherence [10]. In other words, B is a prerequisite of A. The inverse of `isRequiredBy`.

- B `isRequiredBy` A: The inverse relation of `requires` [8]. Included for the sake of completeness. It will normally not be part of an extracted course dataset.

- A `references` B: Educational resource A references, cites, or otherwise points to educational resource B [9]. The inverse of `isReferencedBy`.

- A `isReferencedBy` B: The inverse relation of `references` [7]. Included for the sake of completeness. It will normally not be part of an extracted course dataset.

- A `isFormatOf` B: Educational resource A is substantially the same as educational resource B, but in another format [5]. E.g. B is a video and A is the audio transcript of A.

3.2 Atomic-Composite ER relationships

The relationships between an atomic and a composite ER are as follows: [6]

- B `isPartOf` A: Composite ER A includes educational resource B [6]. For example, B is a video that is part of iER A that also includes other components, such as readings. The inverse of `hasPart`.

- A `hasPart` B: The inverse relation of `isPartOf` [4]. Included for the sake of completeness. It will normally not be part of an extracted course dataset.

Note that `isPartOf` can technically be a relation between two composite ERs in the case where several levels of containment exist. At the moment, there is no use case for something like that, so this remains as an atomic-composite relation, but it may be revisited.

3.3 Composite-Composite ER relationships

Composite-composite relations can be inferred from their contents. For example, if iER A contains ER a, iER B contains ER b, and a `requires` b, then A `requires` B as well. As a result, any atomic-atomic relation can be a composite-composite relation as well.

There are also some relations that apply only at the composite level. We first address relations that apply to course sequencing:

- A `comesAfter` B: Composite educational resource A is assigned in the context of a course at a later time than educational resource B. Does not imply a prerequisite relation between A and B (if such a relation exists, the `requires` relation must be used). The inverse of `comesBefore`.
- B `comesBefore` A: The inverse relation of `comesAfter`. Included for the sake of completeness. It will normally not be part of an extracted course dataset.

An important constraint for the `comesAfter` relation is that at most one composite ER can come after a given composite ER. In other words, it is not possible to have both X `comesAfter` A, and Y `comesAfter` A.

It is important to note that course datasets must include the minimal relations required to ensure proper sequencing in a course, i.e. transitive `requires` relations between ERs are not part of the dataset. Tools using the dataset, such as the LePa Visualizer, can infer such higher order relations if needed for the views they intend to present.

The following relations relate to assessments:

- R `assesses` A: Composite ER R (which must be of type rER) is used to assess composite ER A (which must be of type aER). The inverse of `isAssessedBy`.
- A `isAssessedBy` R: The inverse relation of `assesses`. Included for the sake of completeness. It will normally not be part of an extracted course dataset.

4 Further Constructions

The entities and relations described above can be used to construct different views for the components of a course. Some examples are listed below.

4.1 Milestone

A course “milestone” consists of an activity (aER) and a rubric (rER) to be employed by a marker/grader to assess said activity. This could be, for example, a homework or quiz within the course. In Figure 1, rER_1 is used to assess aER_1. Completing this combination of aER and rER constitutes the completion of a milestone within the course.

A milestone can be represented as a single node. This node can “expand” and “collapse” to show fine vs coarse level of detail. An extended view would include the atomic-ER(s) hosted within the composites of the shown aER and rER, semi-collapsed view would show just the aER and rER associated, and a fully collapsed view would showcase just the milestone.

4.2 Instructional Path

In between milestones, learners engage with instructional material. An iER encapsulates a set of atomic instructional ERs that can be engaged with in any order. In the case where ordering is required, this can be modelled as a sequence of iERs connected with requires relations. For higher level views, this sequence can be abstracted as a single node referred to as an *instructional path.*

Instructional path nodes have `contain` relations to their constituent iERs.

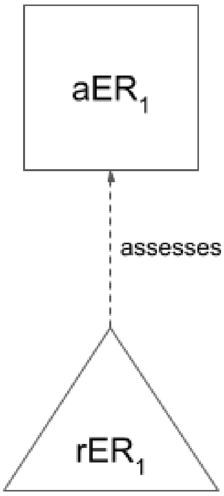

Figure 1: Milestone

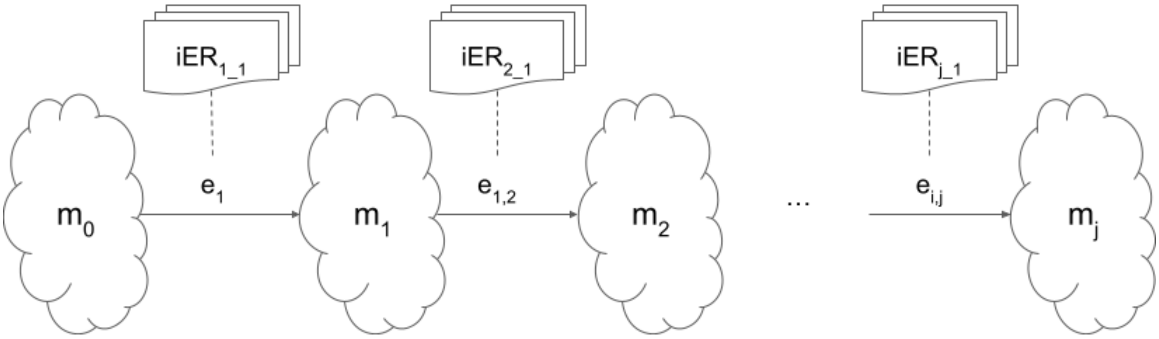

Figure 2: Learning Path

4.3 Learning Path

A "learning path" holds a course's content; milestones and instructional material. This content can be viewed in a linear, metro map or adaptive manner. A linear path goes from one milestone to the next in a linear manner, a metro map adds more variability where there could be multiple paths through the content dictated by need or preference, while an adaptive manner gives suggestions for potential content.

Figure 2 shows a potential learning path starting with m_0, an optional activity to assess the learner's current knowledge, in which each milestone is accompanied by a set of instructional materials and there are m_j milestones.

4.4 Summative Evaluation Path

A "summative evaluation path" is a type of learning path in which each of the milestones m_i have a weight of w_i. This weight indicates the given grade earned from each milestone. These weights can then be calculated together to get the overall grade for the course.

References

[1] Heidi Goodrich Andrade. "Using rubrics to promote thinking and learning." In: *Educational leadership* 57.5 (2000), pp. 13–19.

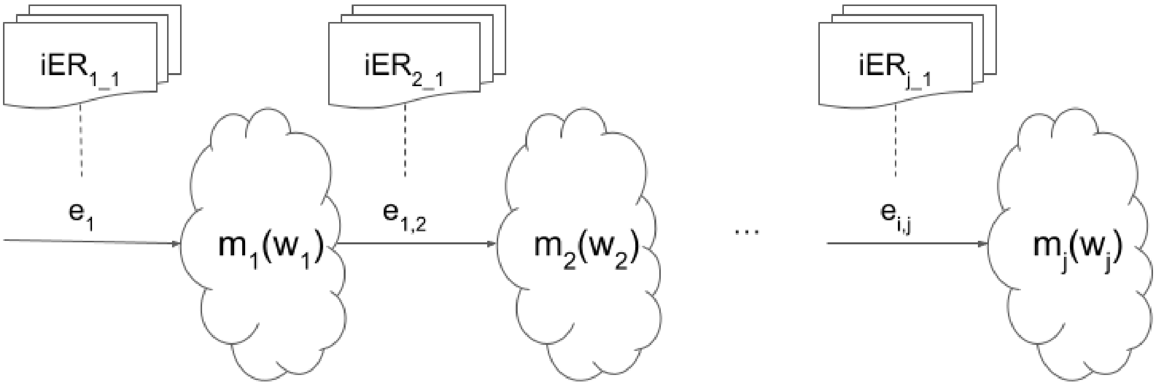

Figure 3: Summative Evaluation Path

[2] Phillip Dawson. “Assessment rubrics: towards clearer and more replicable design, research and practice”. In: *Assessment & Evaluation in Higher Education* 42.3 (2017), pp. 347–360. DOI: 10.1080/02602938.2015.1111294. eprint: https://doi.org/10.1080/02602938.2015.1111294. URL: https://doi.org/10.1080/02602938.2015.1111294.

[3] DublinCore. *DublinCore*. 2022. URL: https://www.dublincore.org/ (visited on 10/18/2022).

[4] DublinCore. *Has Part*. 2022. URL: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/terms/hasPart/ (visited on 10/18/2022).

[5] DublinCore. *Is Format Of*. 2022. URL: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/terms/isFormatOf/ (visited on 10/18/2022).

[6] DublinCore. *Is Part Of*. 2022. URL: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/terms/isPartOf/ (visited on 10/18/2022).

[7] DublinCore. *Is Referenced By*. 2022. URL: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/terms/isReferencedBy/ (visited on 10/18/2022).

[8] DublinCore. *Is required by*. 2022. URL: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/terms/isRequiredBy/ (visited on 10/18/2022).

[9] DublinCore. *References*. 2022. URL: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/terms/references/ (visited on 10/18/2022).

[10] DublinCore. *Requires*. 2022. URL: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/terms/requires/ (visited on 10/18/2022).

[11] Barbara M Moskal and Jon A Leydens. “Scoring rubric development: Validity and reliability”. In: *Practical assessment, research, and evaluation* 7.1 (2000), p. 10.