From PDF Assessments to LMS Deployment: A Model-Driven QTI-Based Framework

Atefeh Nirumand[1], Renzo Degiovanni[1], and Jordi Cabot[1,2]

[1] Luxembourg Institute of Science and Technology, Luxembourg
[2] University of Luxembourg, Luxembourg
{atefeh.nirumand, renzo.degiovanni, jordi.cabot}@list.lu

Abstract. Learning Management Systems (LMSs) increasingly rely on digital assessments to support automated evaluation, content reuse, and flexible learning scenarios. While the IMS Question and Test Interoperability (QTI) specification provides a standardized, platform-independent format for representing assessment items, its practical adoption remains limited due to fragmented tool support, partial specification coverage, and insufficient integration with execution environments. In addition, assessment content is frequently authored and distributed in document-oriented formats, particularly PDF files, which lack explicit structural and semantic information and are therefore difficult to transform into standardized, machine-processable representations.

This paper proposes a hybrid transformation pipeline that combines large language model (LLM)–based document interpretation with a QTI-based metamodel and deterministic model-driven engineering (MDE) techniques to address these challenges. The approach recovers structured assessment items from unstructured documents and reliably transforms them into standardized, LMS-ready representations. Evaluation on real-world repositories, including IMS QTI examples and the Canterbury Question Bank, demonstrates correct semantic preservation, reliable transformation behaviour, and successful import of the generated assessments into an LMS. These findings establish a solid foundation for future extensions to additional LMS platforms.

Keywords: Learning Management System (LMS) · IMS Question and Test Interoperability (QTI) · Model-Driven Engineering (MDE).

1 Introduction

The utilization of digital assessments, encompassing quizzes, examinations, and self-evaluation activities, has witnessed a substantial augmentation within higher education and online training environments, particularly within the domain of Learning Management Systems (LMSs) [9]. These assessment artifacts are frequently authored, revised, and reused over time, often across different tools and platforms. In order to facilitate reuse, maintainability, and structured processing of assessment content, the use of standardized representations of assessment items is imperative [16]. The IMS Question and Test Interoperability (QTI) specification[3] was introduced to address this need by providing a platform-independent format for the representation of assessment items and tests.

Despite its maturity and expressive power, the practical adoption of QTI remains limited. Existing authoring tools and import mechanisms frequently support only legacy QTI versions, provide only partial coverage of the specification, or lack reliable transformation pipelines to LMS execution environments [8,2]. As a result, even widely used LMS platforms exhibit limited and inconsistent support for importing QTI 3.0 content. This has the effect of impeding the direct reuse of standardized assessment materials [8,16]. Consequently, assessment content is often restricted to proprietary formats or manually re-authored, thereby increasing development effort and the risk of semantic inconsistencies.

A further challenge arises at the authoring stage, as assessment content is often created in document-oriented formats such as PDF. While human-readable, these formats lack explicit structural and semantic information, making systematic transformation into standardized representations like QTI non-trivial. Addressing this gap requires techniques for extracting assessment structure and semantics from unstructured or semi-structured documents.

Model-Driven Engineering (MDE) offers a principled approach to addressing these challenges by elevating models to first-class artifacts throughout the software lifecycle [3]. Its emphasis on abstraction, explicit metamodeling, and automated model-to-model and model-to-text transformations makes it particularly well suited for operationalizing complex domain standards such as QTI within educational software systems.

In this work, we propose a transformation pipeline for the automated generation of LMS-ready assessment content from document-based sources, following the conceptual sequence PDF $\rightarrow$ QTI $\rightarrow$ LMS. The approach incorporates an LLM-based component to interpret and structure assessment content from PDF documents, producing QTI 3.0–compliant XML representations. The resulting QTI artifacts are subsequently processed through deterministic, model-driven transformations, ensuring reliable and reproducible execution in LMS platforms. This is done in collaboration with the company Open Assessment Technologies S.A. (OAT[4]) that provides advanced assessment solutions for education in 194 countries and more than 30 languages, which helped us to review and validate the proposed transformations.

The approach is evaluated using real-world repositories, including the official IMS QTI examples [4] and the Canterbury Question Bank [5]. The results demonstrate effective semantic mapping, coverage of key constructs, and successful import into an LMS environment. The pipeline is implemented on top of the BESSER [1] model-driven framework.

The remainder of this paper is organized as follows. Section 2 introduces background concepts, followed by the proposed approach in Section 3 and tool support in Section 4. Evaluation results are presented in Section 5. Related work

[3] https://www.imsglobal.org/spec/qti/v3p0/oview
[4] https://www.taotesting.com/

is reviewed in Section 6, followed by generalization, limitations, and threats to validity in Section 7. Section 8 concludes the paper.

2 Background

This section introduces QTI and Learning Management Systems.

2.1 IMS Question and Test Interoperability (QTI)

The IMS Question and Test Interoperability (QTI) specification[5] defines a standardized, platform-independent data model for representing assessment items, tests, response processing, and outcomes. Developed by the IMS Global Learning Consortium, QTI aims to support exchange, reuse, and long-term preservation of assessment content across heterogeneous educational systems. It provides both an abstract conceptual model and an XML-based serialization, supporting a wide range of question types. While successive versions have increased expressiveness, they have also introduced substantial implementation complexity [8].

2.2 Learning Management Systems and Moodle

Learning Management Systems (LMSs) are central to contemporary higher education and online training, providing integrated support for course management, learning content delivery, and assessment execution. In the realm of open-source LMS platforms, Moodle is one of the most widely adopted and extensively deployed systems worldwide. Moodle supports a variety of assessment types and relies on its own XML-based formats for the import and export of question [11].

3 Approach

This section presents the proposed transformation pipeline, which facilitates the automated generation of LMS-ready assessment content from document-based sources such as PDF files. The approach combines LLM-based content extraction with deterministic, model-driven transformations, ensuring both flexibility at the input level and reliability at the execution level.

The pipeline follows the sequence PDF $\rightarrow$ QTI $\rightarrow$ LMS (Figure 1), using QTI as a standardized pivot representation to decouple document interpretation from LMS-specific execution formats. It is organized into two phases: (i) PDF-to-QTI and (ii) QTI-to-LMS transformation. The approach adheres to a strict separation of concerns, employing probabilistic techniques only for document interpretation, while all subsequent transformations rely on explicit models and deterministic model-driven techniques to preserve assessment semantics. QTI pivots non-deterministic document interpretation from deterministic transformations; skipping it may reduce reproducibility and interoperability.

[5] https://www.imsglobal.org/spec/qti/v3p0/oview

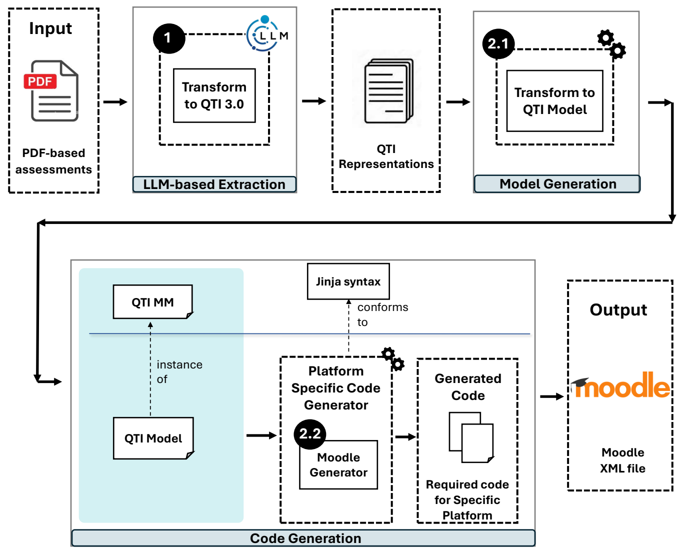


Fig. 1. Overview of the Proposed Assessment Transformation Pipeline.

3.1 PDF-to-QTI Transformation

In the first phase, an LLM-based module transforms assessment content from PDF documents into QTI 3.0–compliant XML. PDF text is extracted page-by-page using *pdfplumber*. The extracted text is provided to the LLM via a carefully designed prompt that enforces well-formed QTI output and specifies required constructs such as items, interactions, and response declarations. In Listing 3.1, an excerpt of the prompt utilized in this process is presented.

Although this step introduces limited non-determinism, the LLM is used exclusively for recovering structure and semantics from unstructured content. The generated QTI files are subsequently validated for syntactic correctness and treated as standardized input for the deterministic transformation stages. This design also enables extensibility to additional input modalities without affecting the deterministic transformation stages.

```
You are an expert in QTI 3.0. Convert examination content into fully valid QTI 3 XML.
STRICT REQUIREMENTS:
1. Output **only** well-formed QTI 3 XML (no markdown, no explanations, no comments).
2. The final output must be **valid UTF-8 encoded XML**.
3. Follow the official QTI 3.0 specification and metamodel provided,
including: <qti-assessment-test>, <qti-test-part>, <qti-assessment-section>,
<qti-assessment-item>, <qti-item-body> (**must be containing the sentence of the
question**), <qti-choice-interaction>, <qti-simple-choice>, response declarations,
outcome declarations, modal feedback, and all required attributes (identifier, title,
cardinality, baseType, etc.)
4. If the input contains multiple questions, produce a single
   <qti-assessment-test> containing multiple <qti-assessment-item> elements.
5. Use the example QTI XML provided as a structural reference and follow the
   metamodel requirements exactly.
6. Ensure that:
   - Every interaction has matching response declarations.
   - Every correct answer is represented in <qti-correct-response>.
   - All identifiers are unique and valid.
   - If there are choices with empty (e.g., "") text, consider these choices.
7. Before finalizing, internally validate the generated XML against the QTI 3 schema.
```

Listing 3.1. Prompt for Schema-Compliant QTI 3.0 Assessment Generation.

Figure 2 illustrates an assessment item related to a PDF and used as input to this phase. The outcome is shown in Listing 3.2, which demonstrates the structured QTI representation produced by the LLM, including (1) the correct-response specification (lines 7-11), (2) the question text (line 14), (3) the definition of answer choices (lines 15-26), and (4) the feedback (lines 28-30) elements.

633573
Suppose you try to perform a binary search on a 5-element array sorted in the reverse order of what the binary search algorithm expects. How many of the items in this array will be found if they are searched for?

a. 5

b. 0

*c. 1

d. 2

e. 3

f. "
g. "
h. "
i. "
j. "
General Feedback:

Only the middle element will be found. The remaining elements will not be contained in the subranges that we narrow our search to.

Fig. 2. Example of PDF-Based Assessment Item.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <qti-assessment-test  Other info omitted for brevity>
3    <qti-test-part identifier="testPart-1" Other info omitted for brevity>
4      <qti-assessment-section Other info omitted for brevity>
5        <qti-assessment-item title="633573" Other info omitted for brevity>
6          <qti-response-declaration Other info omitted for brevity>
7            <qti-correct-response><qti-value>QUE_2001_A3</qti-value>
8            </qti-correct-response>
9          </qti-response-declaration>
10         <qti-outcome-declaration Other info omitted for brevity/>
11         <qti-item-body>
12           <p>Suppose you try to perform a binary search on a 5-element array sorted
                 in the reverse order of what the binary search algorithm expects. How
                 many of the items in this array will be found if they are searched
                 for?</p>
13           <qti-choice-interaction responseIdentifier="QUE_2001_RL" maxChoices="1"
                 minChoices="1">
14             <qti-simple-choice identifier="QUE_2001_A1"><p>5</p></qti-simple-choice>
15             <qti-simple-choice identifier="QUE_2001_A2"><p>0</p></qti-simple-choice>
16             <qti-simple-choice identifier="QUE_2001_A3"><p>1</p></qti-simple-choice>
17             <qti-simple-choice identifier="QUE_2001_A4"><p>2</p></qti-simple-choice>
18             <qti-simple-choice identifier="QUE_2001_A5"><p>3</p></qti-simple-choice>
19             <qti-simple-choice identifier="QUE_2001_A6">"</qti-simple-choice>
20             <qti-simple-choice identifier="QUE_2001_A7">"</qti-simple-choice>
21             <qti-simple-choice identifier="QUE_2001_A8">"</qti-simple-choice>
22             <qti-simple-choice identifier="QUE_2001_A9">"</qti-simple-choice>
23             <qti-simple-choice identifier="QUE_2001_A10">"</qti-simple-choice>
24           </qti-choice-interaction>
25         </qti-item-body>
26         <qti-modal-feedback identifier="QUE_2001_ALL" outcomeIdentifier="FEEDBACK"
               showHide="show">
27           <p>&lt;p&gt;Only the middle element will be found. The remaining elements
                 will not be contained in the subranges that we narrow our search
                 to.&lt;/p&gt;</p>
28         </qti-modal-feedback>
```

```
      </qti-assessment-item>
      <!-- Other info omitted for brevity -->
    </qti-assessment-section>
  </qti-test-part>
</qti-assessment-test>
```

Listing 3.2. Generated QTI 3.0 Representation of a PDF-Based Assessment Item.

3.2 QTI-to-LMS Transformation

This phase has two steps: (i) transform QTI XML to a model-based representation, and (ii) transform the model to a concrete LMS format.

Text-to-Model Transformation To enable LMS-independent processing, QTI XML from the previous phase is parsed into a model based on QTI 3.0. Rather than operating directly on XML syntax, this representation captures the core semantic concepts of assessment items, such as questions, responses, scoring rules, and feedback, in an explicit and analyzable form.

The QTI-based metamodel underlying this representation is designed using *Domain Engineering* principles to capture the essential semantics required for assessment execution. This abstraction enables clearer reasoning about assessment logic and facilitates deterministic downstream transformations. A parser traverses the validated QTI and instantiates model elements, preserving semantic relationships such as the association between questions and their responses, the identification of correct answers, and the definition of scoring behavior. This step ensures that all subsequent transformations are deterministic, reproducible, and independent of the original XML structure.

The proposed metamodel is structured into three layers: (i) Assessment Organization (Figure 3), (ii) Content and Presentation (Figure 4), and (iii) Response and Evaluation Semantics (Figure 5). To reduce redundancy and improve coherence, a common $Identifiable$ base abstraction is used across metaclasses that require stable identification. The metamodel concentrates on core concepts to preserve essential semantics while minimizing complexity.

Assessment Organization. Assessments are modeled using the concepts *AssessmentDefinition*, *AssessmentPart*, *AssessmentSection*, and *Question*. These elements capture the logical composition and flow of an assessment. Navigation and submission behaviour are represented via dedicated enumerations (e.g., linear vs. nonlinear, individual vs. simultaneous), enabling platform-independent reasoning about assessment execution semantics. This layer avoids presentation and scoring details in order to maintain a clear, high-level abstraction of assessment structure.

Content and Presentation Abstraction. As shown in Figure 4, question content and learner interaction are modeled through the *Question* and *QuestionBody* abstractions. A *Question* represents a complete assessment item and aggregates content, response declarations, and feedback. The *QuestionBody* encapsulates the instructional and interactional elements of an item, including prompts, paragraph blocks, and selection constraints such as minimum and maximum allowed

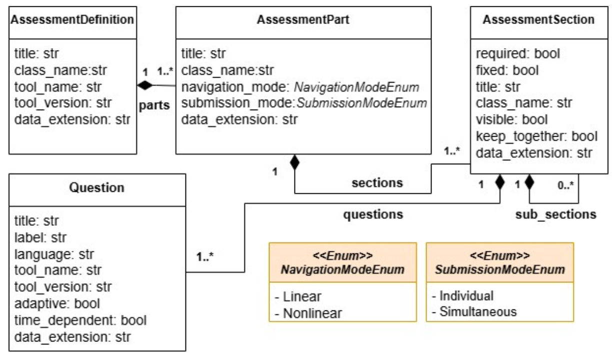


Fig. 3. Assessment Metamodel.

choices. Interaction options are represented as *Choice* elements. This layered design preserves essential assessment semantics while maintaining a level of abstraction suitable for model-driven transformation and reuse.

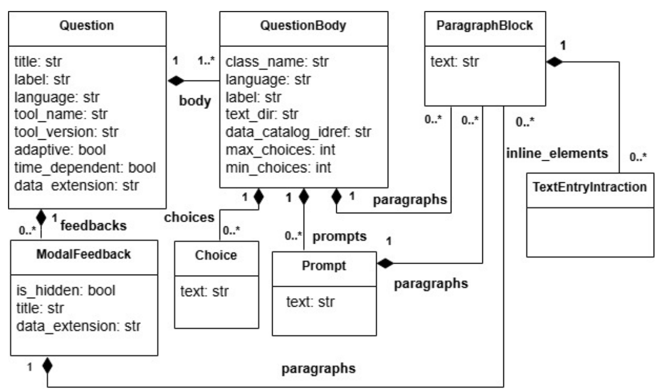


Fig. 4. Content and Presentation Metamodel.

Response and Evaluation Semantics. As shown in Figure 5, a *Question* may be associated with the *ResponseDeclaration* metaclass. The *ResponseDeclaration* abstraction is employed to model response semantics independently from interaction content. A *ResponseDeclaration* specifies the expected structure of learner input by defining its cardinality and base type, while delegating evaluation semantics to explicit representations of correct choices and acceptable answers. In particular, for essay-type questions, correctness is captured through the *Answer* abstraction, which associates permissible responses with corresponding scores. This design enables uniform representation of correct and alternative answers, while maintaining a clear separation between response structure and evaluation logic. In this version of the metamodel, we do not address response processing, as the focus is on semantic preservation and deterministic transformation rather than execution logic. A full specification and representative examples are available in our repository.[6]

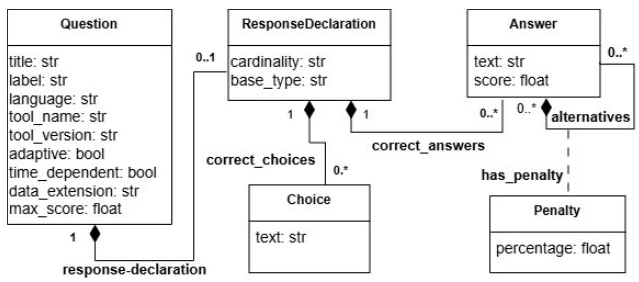


Fig. 5. Response and Evaluation Semantics Metamodel.

Model-to-Text Transformation The final step generates executable LMS-compatible artifacts from the model-based representation using a model-to-text transformation. To demonstrate the feasibility of this approach, Moodle, as a widely used LMS platforms, is considered as a concrete target LMS in the current implementation. Transformation rules are encoded using Jinja[7] templates. For each question type, explicit transformation rules define how content, interactions, scoring information, and feedback are translated into Moodle's XML. The current implementation supports multiple-choice (single- and multiple - answer), true/false, short-answer, and essay question types, representing the bulk of typical assessment content. The generated XML is syntactically valid and LMS-ready without manual adaptation.

As illustrated in Listing 3.3, the Jinja template generates LMS-specific answer definitions by iterating over the modeled choice set and deriving fractional scores from the number of correct responses. The template supports both single- and multiple-correct configurations, normalizes scoring values accordingly, and conditionally adapts textual rendering based on the question type. Each answer is rendered as an <*answer*> element that combines scoring information with associated feedback. The template embeds feedback logic directly within the generated LMS artifact, producing feedback for correct and incorrect responses. This exemplifies how scoring and feedback semantics captured at the model level are deterministically translated into executable LMS representations.

```
{%- for choice in question.body.choices | sort(attribute='identifier') %}
 {%- set frac = 100 / num_correct %}
 <answer fraction="{% if choice.identifier in correct_ids %}{{ frac if frac % 1 else
      frac | int }}{% else %}0{% endif %}">
```

[6] https://github.com/AtefehNirumandJazi/PDF-to-LMS-Converter

[7] https://jinja.palletsprojects.com/en/stable/

```
  {%- if ns.question_type == 'truefalse' %}
   <text>{{choice.text | lower}}</text>
  {%-else%}
   <text><![CDATA[{{choice.text}}]]></text>
  {%- endif %}
   <feedback format="html">
     <text><![CDATA[
     {%- if choice.identifier in correct_ids %}
     Correct!
     {%- else %}
     Incorrect. The correct answer is {{ question.body.choices |
          selectattr('identifier', 'in', correct_ids) | map(attribute='text') |
          join(', ') }}
     {%- endif %}]]></text></feedback></answer>{%- endfor %}
```

Listing 3.3. Excerpt of a Jinja template for LMS-compatible XML answer generation.

Figure 6 shows the successful rendering of an item generated by the pipeline within the LMS environment, based on the PDF example shown in Figure 2. This result confirms both the correctness and practical applicability of the proposed approach.

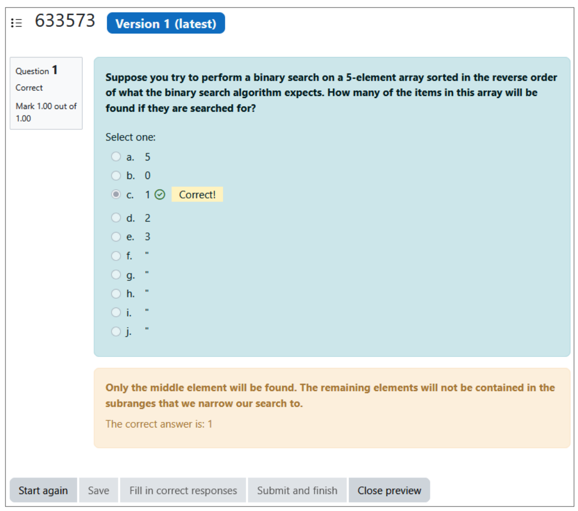


Fig. 6. LMS Rendering of an Assessment Item Generated by the Pipeline.

4 Tool Support

The entire pipeline is implemented on top of the BESSER [1] model-driven framework, which provides the execution environment for the proposed metamodel and the associated transformations. BESSER facilitates the integration of parsing, modeling, and code generation within a unified framework, thereby enabling extensibility and systematic evolution of the approach. As explained earlier, the PDF-to-QTI transformation phase relies on an LLM. According to the technical capabilities described by OpenAI [14], only multimodal models that support both textual and visual inputs are suitable for processing PDF documents directly. Based on this constraint, we evaluated the LLM-based component of the pipeline using several candidate models: GPT-4o, GPT-4o-mini, GPT-5-mini, and GPT-5.1.

A preliminary evaluation was conducted on a subset of the dataset described in Section 5, to select the most promising LLM. The results indicate that GPT-4o and GPT-5.1 consistently achieve superior performance in terms of syntactic correctness of the generated QTI XML, structural consistency with the expected QTI schema, and preservation of the semantic content of the original assessment items. In addition, timing measurements show that GPT-5.1 provides noticeably faster response times compared to GPT-4o. Based on these findings, GPT-5.1 was selected as the default model for the LLM-based transformation phase.

5 Evaluation

The proposed pipeline is evaluated on the following research questions:

RQ1 (Correctness): To what extent does the transformation pipeline preserve the semantic correctness of QTI-based assessments when generating LMS-compatible representations?
RQ2 (Flexibility): How well does the pipeline support the structural and behavioural constructs defined in QTI 3.0 across real-world assessments?
RQ3 (Run-time Performance): How does the pipeline perform and scale when processing assessments of increasing size and complexity?

Datasets and Experimental Setup. The evaluation covers both LLM-based and deterministic stages of the proposed pipeline and is based on two publicly available and widely adopted repositories: (i) The *Canterbury Question Bank* [5], which provides a large collection of assessment items originally distributed as PDF and QTI files. (ii) The *official IMS QTI* examples repository [4], which contains reference QTI examples covering a range of interaction types and specification features. We consider 120 case studies from the Canterbury Question Bank dataset that only focus on textual assessment materials, and discard the ones containing images, not supported yet. The QTI examples repository contains a set of examples; however, for the purposes of this evaluation, we selected five representative cases corresponding to different question types, namely multiple-choice, true/false, short-answer, and essay questions. The experimental phase of this study was conducted on a device with Windows 11 Enterprise, an Intel(R) Core(TM) i7-1280P 1.80 GHz CPU, and 32.0 GB RAM.

5.1 Correctness (RQ1)

Correctness is evaluated at two key phases of the pipeline: (i) the PDF-to-QTI transformation (LLM-based) and (ii) the QTI-to-LMS transformation.

Table 1. Correctness Evaluation Results of the PDF-to-QTI Transformation.

Metric	Result
Question text similarity	100%
Feedback text similarity	100%
Correct answer matching	100%
Choice count mismatch	0
Total answer choices (ground truth)	1154
Total answer choices (generated QTI)	1154
Total assessment items evaluated	120
Precision (item-level) = 100%, **Recall (item-level)** = 100% , **F-measure** = 100%	

PDF-to-QTI Correctness. We use as reference the QTI files provided by the Canterbury QuestionBank (originally in QTI 1.2 format), with their corresponding PDF documents, as ground truth in our evaluation. As our pipeline produces QTI 3.0 output, the reference QTI 1.2 files were migrated to QTI 3.0 using an existing tool[8], and subsequently refined using a Python-based post-processing script to ensure full QTI 3.0 compliance. The resulting files serve as ground truth for an automated evaluation framework, which compares generated and reference QTI files at the item level, measuring question and feedback text similarity, correctness preservation, and structural alignment of answer choices.

Textual similarity is computed using normalized string similarity metrics based on the *SequenceMatcher* algorithm from Python's *difflib* library, applied to normalized strings. Answer choices are aligned using a best-fit matching strategy with a high-confidence threshold. Based on these item-level comparisons, precision, recall, and F1-score are computed to quantify structural and semantic preservation. As summarized in Table 1, the results show full preservation of the question text, feedback, correct answers, and choice structures. All generated items exhibit identical choice cardinality to the ground truth, with no structural discrepancies observed, yielding precision, recall, and F1-scores of 100% at the item level.

QTI-to-LMS Correctness. The reference QTI files serve as ground truth and are compared against the LMS artifacts generated by the pipeline. The automated evaluation is employed to compare QTI and LMS representations. This is achieved by computing multiple metrics, including question text similarity, feedback similarity, correct answer preservation, and answer choice overlap. Special care is taken to normalize HTML, CDATA sections, and formatting differences introduced by LMS-specific XML syntax. As demonstrated in Table 2, the deterministic transformation has been shown to preserve assessment semantics with a high degree of fidelity, whilst introducing no information loss. To support replicability, we provide the evaluation scripts, datasets used in the evaluation, and the corresponding generated QTI and LMS XML artifacts[9].

[8] https://github.com/sonyccd/qti-migrator/
[9] https://github.com/AtefehNirumandJazi/PDF-to-LMS-Converter

Table 2. Correctness Evaluation of the QTI-to-LMS Transformation.

Metric	**Result**
Question text similarity	100%
Feedback text similarity	100%
Correct answer preservation accuracy	100%
Items with choice count mismatch	0
Total answer choices (reference QTI)	1154
Total answer choices (LMS XML)	1154
Total assessment items evaluated	120
Precision (item-level) = 100%, **Recall (item-level)** = 100% , **F-measure** = 100%	

5.2 Flexibility (RQ2)

Flexibility assesses the ability of the proposed pipeline to effectively handle assessment material exhibiting structural and linguistic variability while preserving semantic correctness during transformations. In our evaluation, flexibility is analyzed from two complementary perspectives: (i) coverage of QTI 3.0 constructs and question types, and (ii) robustness to varying linguistic complexity of the input assessment documents.

Support for QTI Constructs and Question Types. The pipeline supports a representative subset of QTI 3.0 interactions including multiple-choice, true/false, short-answer, and essay questions, which account for the majority of assessments in practice and are directly supported by mainstream LMSs. The evaluation covers 120 real-world assessments from the Canterbury QuestionBank and representative examples from the official QTI repository. Across all cases, the pipeline successfully generated LMS-compatible assessments that were imported without manual intervention, demonstrating robustness across different QTI organizations, item structures, and feedback configurations.

Robustness to Linguistic Complexity. To assess robustness at the input level, the linguistic complexity of the PDF-based questions was analyzed using standard readability and lexical metrics (Table 3). The results show substantial variation in question length, lexical diversity, and readability, ranging from short factual items to long, multi-sentence descriptions and from simple to academically dense text. Despite this variability, the pipeline consistently extracted structured QTI representations and successfully generated LMS-compatible assessments, demonstrating that the LLM-based extraction is resilient to differences in wording and complexity, while the subsequent model-driven transformations ensure deterministic and stable behavior.

5.3 Run-time Performance (RQ3)

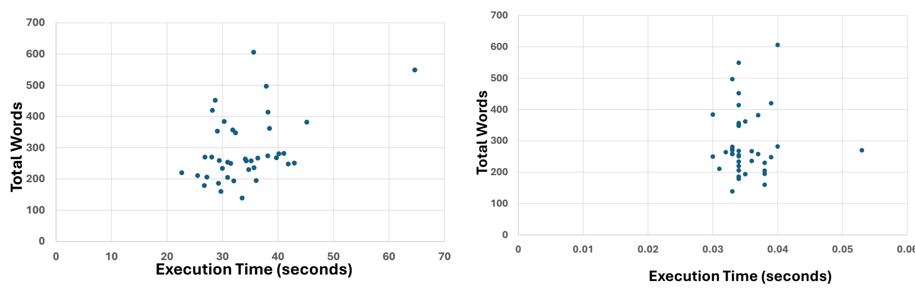
Although the run-time performance is not a primary concern of this work, execution-time measurements are reported to confirm its practical feasibility.

Table 3. Descriptive Statistics of Textual Complexity Metrics.

Statistic	Total Words	Avg. Word Length	Type-Token Ratio	Flesch Reading Ease
Min	6	2.62	0.45	30.36
Max	383	5.52	1.00	102.21
AVG	98.6	4.257	0.74	75.34

The evaluation was conducted on 40 PDF documents (three items each), resulting 40 independent executions. The total execution time was divided into two phases: (i) PDF-to-QTI transformation and (ii) QTI-to-LMS transformation. The document size was approximated by the total number of words per PDF.

As shown in Figure 7, the PDF-to-QTI phase dominates total time due to LLM-based interpretation, ranging from about 22.7 to 64.6 seconds and growing with document length. The QTI-to-LMS phase consistently finishes in 0.03–0.05 seconds, driven by deterministic transformations. Although not a scalability study, the results indicate the pipeline adds no significant overhead beyond the initial interpretation step and is suitable for batch-oriented assessment authoring in practical educational settings.



(a) Processing Time for PDF-to-QTI vs. total words

(b) Processing time for QTI-to-LMS vs. total words

Fig. 7. Scatter plots of processing time versus document size (total words) for the (a) PDF-to-QTI and (b) QTI-to-LMS transformation phases.

6 Related Work

Related work can be grouped into two areas: (i) model-based formalization of the IMS QTI specification, and (ii) approaches for transforming document-based assessment content into LMS-compatible formats.

6.1 QTI Formalization and Model-Based Approaches

Radenković et al. [15] propose a UML-based QTI metamodel using the Model-Driven Architecture (MDA) paradigm to enable interoperability among QTI

2.1–compliant systems, emphasizing response processing and semantic consistency, including Description Logic–based extensions. Their approach assumes assessment content is already available as structured QTI artifacts. In contrast, our QTI 3.0 metamodel focuses on high-level assessment organization, employs type-safe enumerations for navigation and submission modes, and leverages Identifiable inheritance for unified identification, supporting modern assessment workflows while reducing response-processing complexity. Early systems such as R2Q2 [18] similarly operate directly on QTI XML for rendering and response processing and are no longer maintained. Recent platforms, such as CleverTesting [2], improve standards compliance and reusability for QTI 2.2 but remain largely syntax-driven and do not address the transformation of unstructured assessment sources. Overall, despite highlighting the importance of QTI for interoperability, these approaches are hindered by XML-centric processing and the lack of lightweight, high-level metamodels suitable for systematic transformation.

6.2 From Document-Based Assessments to LMS-Compatible Formats

A second line of work addresses the transformation of assessment content from document-oriented formats into LMS environments. Most existing approaches assume structured inputs (e.g., databases or JSON) and focus on course content rather than assessment logic. For example, Kandagor [6] present a migration framework for structured course materials but explicitly exclude assessment interoperability standards and unstructured documents. Other approaches, such as the integration of LaTeX with Moodle proposed by Gallego et al. [17], improve authoring efficiency but rely on manual authoring and target a specific LMS without providing an intermediate, LMS-independent representation. Existing LMS plugins for QTI import are often limited to obsolete QTI versions and suffer from robustness issues [7,12,13,10]. The proposed approach addresses these limitations by combining LLM-based document interpretation with QTI-centered, model-driven transformations, enabling systematic validation, deterministic processing, and reuse across LMS platforms.

7 Generalization, Limitations, and Threats to Validity

Generalization to other languages. The proposed approach is extensible across languages and document sources. Although the evaluation focuses on English-base assessments, a small set of French-language assessments[10] was successfully processed without modifying the LLM prompt or the QTI metamodel. Language-specific adaptations were handled through lightweight preprocessing and XML normalization, demonstrating a clear separation between language-dependent content interpretation and the underlying semantic and transformation logic.

[10] Examples selected from: https://eduscol.education.fr/4157/la-bibliotheque-d-outils-de-positionnement-un-ensemble-de-ressources-au-service-des-enseignants

Further details, including representative assessment examples and corresponding output artifacts, are available in the project repository[11].

Limitations. While the proposed approach exhibits strong accuracy and semantic preservation, the outputs of LLM-based document interpretation phase may vary depending on document layout, linguistic clarity, and prompt configuration. Although deterministic transformations mitigate this variability, incorrect interpretations at the extraction stage may still affect the generated assessment models. Moreover, the metamodel intentionally focuses on core QTI constructs, improving robustness and analyzability but currently limiting support for advanced interactions, adaptive testing, and complex logic.

Threats to Validity. The reliability of the evaluation is contingent upon the quality and representativeness of the document-based datasets used in the experiments. To mitigate this threat, the evaluation draws on the Canterbury QuestionBank, which contains diverse and realistic assessment item structures. Nevertheless, generalization threats remain, as the results may not fully extend to other PDF-to-LMS scenarios beyond the evaluated cases, potentially affecting external validity. This risk is partially addressed by incorporating the official IMS QTI examples to cover multiple question types and by targeting Moodle as a widely used LMS, thereby increasing the relevance of the findings. However, broader empirical validation remains a future objective.

8 Conclusion

We presented a transformation pipeline that generates LMS-ready assessment content from document-based sources by combining LLM-based semantic extraction with deterministic, model-driven transformations. Using QTI as a pivot and a streamlined, domain-oriented metamodel, the approach reliably recovers assessment structure and semantics from unstructured documents and supports reproducible LMS deployment. Evaluation on real-world repositories shows semantic preservation, stable transformation, and successful LMS import for common question types.

Future work will focus on extending metamodel coverage to additional QTI interaction types and response-processing constructs, improving robustness of semantic extraction for complex document layouts, and supporting multiple LMS targets. We also plan to enable bidirectional pipeline execution, allowing transformations not only from document-based sources to LMS-ready formats but also in reverse, such as from Moodle to QTI or PDF. Further empirical studies involving instructors and large assessment corpora are also planned to evaluate usability, correctness, and impact on assessment generating efficiency. Comparative experiments using open-source AI models will also be conducted to assess their effectiveness in semantic extraction and transformation accuracy.

Acknowledgments. This work has been partially funded by the RDI Law project "Innovations for 21st Century Assessment Authoring" financed by the Luxembourg

[11] https://github.com/AtefehNirumandJazi/PDF-to-LMS-Converter

Ministry of the Economy, and the Luxembourg National Research Fund (FNR) PEARL program (grant agreement 16544475).

References

1. Alfonso, I., Conrardy, A., Sulejmani, A., Nirumand, A., Ul Haq, F., Gomez-Vazquez, M., Sottet, J.S., Cabot, J.: Building besser: an open-source low-code platform. In: International Conference on Business Process Modeling, Development and Support. pp. 203–212. Springer (2024)
2. Boussakuk, M., Bouchboua, A., El Ghazi, M., El Bekkali, M., Fattah, M.: Designing and developing e-assessment delivery system under ims qti ver. 2.2 specification. International Journal of Emerging Technologies in Learning (iJET) **16**(1), 219–233 (2021)
3. Brambilla, M., Cabot, J., Wimmer, M.: Model-driven software engineering in practice. Morgan & Claypool Publishers (2017)
4. Consortium, I.G.L.: 1edtech/qti-examples (2025), `https://github.com/1EdTech/qti-examples`
5. Group, C.W.: Canterbury questionbank (2025), `https://web-cat.org/questionbank/`
6. Kandagor, S.: Practical framework for migrating json format course content from wordpress cms to moodle lms (2025)
7. Kutzner, T., Meißner, M., Nesterow, I., Rojas, P., Freytag, A.: Concept of a cross-university question exchange platform with moodle lms (2020)
8. Kutzner, T., Nesterow, I., Freytag, A.: New moodle plugin for ims question and test interoperability specification (qti) (2021)
9. Lang, S.: Learning Management Systems (LMSs), pp. 173–182. Springer International Publishing, Cham (2023)
10. moodle.org: Github: qformat_imsqti (2025), `https://github.com/jmvedrine/moodle-qformat_imsqti21`
11. Moodle.org: Moodle (2025), `https://moodle.org/`
12. moodle.org: Moodle: Plugins. (2025), `https://moodle.org/local/plugins/`
13. moodle.org: Question formats: Questionmark qml importer (2025), `https://moodle.org/plugins/qformat_qml`
14. openai: Openai platform (2025), `https://platform.openai.com/docs/guides/pdf-files?api-mode=chat`
15. Radenković, S., Krdžavac, N., Devedžić, V.: A qti metamodel. In: Proceedings of International Multiconference on Computer Science and Information Technology, T ISSN. vol. 7094, p. 2007 (1896)
16. Sanchez, L., Penarreta, J., Soria Poma, X.: Learning management systems for higher education: a brief comparison. Discover Education **3**(1), 58 (2024)
17. Terris-Gallego, R., Fabra, F., López-Salcedo, J.A., Seco-Granados, G.: Designing parametric multiple-choice tests with moodle and latex for engineering education. In: 2025 34th Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE). pp. 1–9. IEEE (2025)
18. Wills, G., Davis, H., Chennupati, S., Gilbert, L., Howard, Y., Jam, E.R., Jeyes, S., Millard, D., Sherratt, R., Willingham, G.: R2q2: rendering and reponses processing for qtiv2 question types (2006)