

## Kopplung von Jupyter Notebooks mit externen E-Assessment-Systemen am Beispiel des Data Management Testers

Martin Petersohn<sup>1</sup>, Konrad Schöbel<sup>2</sup> und Andreas Thor <sup>3</sup>

**Abstract:** Wir präsentieren einen generischen Ansatz für interaktives E-Assessment mit Jupyter Notebooks, bei dem ein externes E-Assessment-System in ein Notebook integriert wird. Im Unterschied zu bisherigen Arbeiten ist das Notebook nicht selbst das Artefakt, welches vom System prozessiert wird, sondern dient vielmehr als User Interface für das E-Assessment-System. Unsere Integration ermöglicht dadurch ein interaktives Assessment während des Bearbeitungszyklus des Notebooks, d.h. Lernende können während der Bearbeitung des Notebooks ohne Unterbrechung bzw. Systemwechsel ihre Lösungen live bewerten lassen. Wir stellen eine prototypische Implementation für die Anbindung des E-Assessment-Tools DMT (Data Management Tester) vor und zeigen dabei auf, welche Schnittstelle ein beliebiges E-Assessment-System implementieren muss, um durch unseren Ansatz in Jupyter Notebooks integriert werden zu können.

**Keywords:** Jupyter Notebook, E-Assessment, Data Management Tester

### 1 Einleitung

Jupyter Notebooks [K116] sind in den letzten Jahren zu einem populären Element der Hochschullehre in MINT-Fächern geworden und werden in verschiedenen Lernszenarien eingesetzt [Jo20]. Lehrende nutzen Jupyter Notebooks z. B. für interaktive Vorlesungen, bei denen sie (vorbereiteten) Quellcode inklusive Beschreibungen im Markdown-Format präsentieren, live entwickeln, verändern und ausführen können. Zusätzlich können vorbereitete Jupyter Notebooks auch an die Lernenden verteilt werden, um ihnen praktische Übungsmöglichkeiten zu ermöglichen.

Damit spielt auch das E-Assessment für Jupyter Notebooks zunehmend eine Rolle, da die von Lernenden innerhalb eines Notebooks erstellten digitalen Artefakte auch Grundlage einer manuellen oder automatischen Bewertung sein können. Werkzeuge wie z. B. nbgrader [Ju19] ermöglichen es Lernenden, Lösungen zu einzelnen Aufgaben in den Zellen entsprechend vorbereiteter Notebooks einzutragen und anschließend zur Bewertung einreichen. Im Unterschied zu bisherigen Systemen, die Jupyter Notebooks mit E-Assessment-Systemen kombinieren, ist in unserem Ansatz das Notebook *nicht* das Artefakt, welches vom System prozessiert wird. Das Notebook dient vielmehr als *User Interface* für

---

<sup>1</sup> Hochschule für Technik, Wirtschaft und Kultur Leipzig, martin.petersohn@stud.htwk-leipzig.de

<sup>2</sup> Hochschule für Technik, Wirtschaft und Kultur Leipzig, konrad.schoebel@htwk-leipzig.de

<sup>3</sup> Hochschule für Technik, Wirtschaft und Kultur Leipzig, andreas.thor@htwk-leipzig.de

das E-Assessment-System. Dabei nutzen wir die Möglichkeit, dass der Output von Notebook-Zellen auch typische User-Interface-Elemente wie Buttons oder Textboxen enthalten kann. Dies ermöglicht die Integration eines *interaktiven Assessments während des Bearbeitungszyklus*, d.h. Lernende können das Notebook bearbeiten, z. B. in einer Zelle Code entwickeln, und ohne Unterbrechung bzw. Systemwechsel ihre Lösung in einer anderen Zelle bewerten lassen. Diese Form ermöglicht es auch, E-Assessment in einer interaktiven Vorlesung direkt in der Lehrveranstaltung einzusetzen.

Wir illustrieren unseren Ansatz prototypisch mit dem E-Assessment-Tool DMT (Data Management Tester) [TK21], welches für typische Aufgaben in der Datenbank-Lehre eingesetzt werden kann. Dabei zeigen wir auf, dass unser Vorgehen generisch ist und beschreiben, welche Schnittstelle (API) ein *beliebiges* E-Assessment-System implementieren muss, um durch unseren Ansatz in Jupyter Notebooks integriert werden zu können. Der Rest des Beitrags ist wie folgt strukturiert: Wir fassen in Kapitel 2 verwandte Arbeiten kurz zusammen und stellen in Kapitel 3 zunächst das anzubindende E-Assessment-Tool DMT näher vor. Unseren Integrationsansatz beschreiben wir in Kapitel 4 und gehen dabei auch auf Details der Implementation ein.

## 2 Related Work

Bei der Kopplung von Jupyter Notebooks mit E-Assessment-Systemen gibt es grundsätzlich zwei Richtungen. Werkzeuge wie z. B. nbgrader [Ju19] führen das Jupyter Notebook als digitales Artefakt dem Bewertungssystem zu, das in der Regel sowohl automatische als auch manuelle Bewertungen zulässt. Hierbei wird auf die Struktur eines Notebooks inkl. Metadaten der Zellen zurückgegriffen, um eine zuverlässige und effiziente Bewertung zu ermöglichen. Die automatische Bewertung (Auto-Grading) kann dabei durch externe E-Assessment-Systeme erfolgen, wie z. B. UNCode [Go21] für die Bewertung von Programmcode, oder durch selbst definierte Testszenarien wie z. B. beim Unit Testing Framework [THS23]. Für einen Vergleich verschiedener Auto-Grader sei auf [Go21] verwiesen.

Die andere Richtung ist die Integration auf Ebene des User Interface. Während unser Ansatz das User Interface nativ als Output der Zelle generiert, nutzt der in [PGA21] vorgestellte Ansatz Webbrowser-Technologien wie IFrames. Damit lassen sich sowohl die Aufgabenstellung und etwaiges Feedback (vom E-Assessment-System) als auch das Jupyter Notebook, auf das dann dynamisch zur Bewertung zugegriffen wird, gleichzeitig ("nebeneinander") darstellen, obwohl sie auf getrennten Systemen liegen. Einen hybriden Ansatz präsentiert [Ma20]. Hier wird das E-Assessment-System Web-CAT, welches automatisch Code verschiedener Programmiersprachen bewerten kann, mittels einer Jupyter-Front-End-Erweiterung angeschlossen. Dabei können Lernende ihr speziell aufbereitetes Notebook direkt an das Web-CAT-System senden und erhalten Bewertung und Feedback als Popup-Fenster von Web-CAT. Im Gegensatz dazu integriert unser Ansatz das Feedback direkt in den Output der jeweiligen Notebook-Zelle.

Welche Bücher wurden vor dem Jahr 1980 herausgegeben?

Geben Sie im **SELECT** nur den Buchtitel aus. Die korrekte Lösung hat 98 Datensätze.

Geben Sie Ihren SQL-Code hier ein ...

Database ready (6 tables [AUTOR, BUCH, BUCH\_AUT, BUC

ABGABE ÜBERPRÜFEN

Wie lautet das Ergebnis der SQL-Query

`SELECT A FROM R WHERE B<4` für die Relation  $R(a,b) = \{(1,6), (2,2), (3,2), (4,5)\}$ ?

A		
1		
2		
3		

ABGABE ÜBERPRÜFEN

Points: 0 of 1  
Unterschiedliche Anzahl an Zeilen (3; erwartet: 2).

Abb. 1: Screenshot des Data Management Testers für zwei verschiedene Aufgaben vom Typ SELECT (links) und TABLE (rechts, mit eingetragener, falscher Lösung und Feedback)

### 3 Data Management Tester

Das E-Assessment-Tool DMT (Data Management Tester) [TK21] erlaubt die automatische Bewertung strukturierter Ergebnisse, die häufig als Lösung von Aufgaben in der Datenbank-Lehre entstehen. Die Ergebnisformate umfassen u.a. SQL-Anfragen, die Spezifikation von Schemata und Relationen sowie View- und Constraint-Definitionen. Abb. 1 zeigt zwei verschiedene Aufgabentypen, die im Folgenden kurz charakterisiert werden.

Bei Aufgaben des Typs SELECT geben Lernende als Lösung eine SQL-Anfrage an (siehe Abb. 1, links). Zur automatischen Bewertung spezifiziert die Lehrperson eine SQL-Anfrage als Musterlösung und gibt zusätzlich an, ob die Reihenfolge der Datensätze relevant ist. Beim Aufgabentyp TABLE geben Lernende die Attributnamen und Tupel einer Relation in Tabellenform an (Abb. 1, rechts). Zum Vergleich mit der korrekten Lösung gibt die Lehrperson entweder eine SQL-Query oder das tabellarische Ergebnis im CSV-Format an. Zusätzlich kann die Lehrperson auch hier spezifizieren, ob die Reihenfolge der Datensätze relevant ist, z. B. wenn das Ergebnis einer ORDER BY-Query gesucht ist.

Wesentlicher Bestandteil von DMT ist die automatische Bewertung sowie die automatische Generierung von Feedback bzgl. der eingereichten Lösung der Studierenden. Für Aufgaben der Formate SELECT und TABLE wird z. B. das Anfrage-Ergebnis bzw. die angegebene Tupel-Menge mit einer Musterlösung verglichen, die ggf. selbst durch eine SQL-Anfrage spezifiziert ist. DMT nimmt anhand der Prüfung nicht nur eine Bewertung vor, ob die Lösung *korrekt* oder *nicht korrekt* ist, sondern generiert auch hinweisgebendes Feedback bei fehlerhaften Lösungen. Dieses bezieht sich auf die Syntax der SQL-Anfrage,

Grad und Kardinalität der Tupel-Mengen (wie im rechten Beispiel von Abb. 1), die Korrektheit des Schemas (Attributnamen) sowie der Tupel und ggf. deren korrekte Reihenfolge. Gemäß der Klassifikation in [Na08] fällt DMTs Feedback in die Bereiche KM (Knowledge about mistakes) sowie AUC (Answer-until-correct). Für weitere Details zur automatischen Bewertung und Generierung von Feedback (auch für alle anderen Aufgabentypen) sei auf [TK21] verwiesen.

Die aktuelle DMT-Implementation ist eine Webservice-basierte Webanwendung unter Verwendung von Java-Servlets, HTML und Javascript. Die Funktionalität wird durch zwei REST-basierte Webservices realisiert: `/getTaskInfo?taskId=...` sowie `/getTaskResult?taskId=...&answer=...`. Die Task-Id ist dabei ein String, der eine Aufgabe innerhalb von DMT eindeutig referenziert. Die Lösung der Lernenden wird im Parameter `answer` als String übergeben. Dieser ist je nach Aufgabentyp speziell strukturiert, z. B. im CSV-Format für den Aufgabentyp TABLE.

## 4 Integration von DMT in Jupyter Notebooks

Abb. 2 zeigt die schematische Architektur und den Workflow zur Kopplung von Jupyter Notebooks mit einem externen E-Assessment-System am Beispiel von DMT. Mit dem vorgestellten Ansatz kann prinzipiell jedes System angebunden werden, welches Informationen zur Aufgabenstellung (`getTaskInfo`) sowie zur Bewertung (`getTaskResult`) programmatisch per API bereitstellt. Die Kopplung wird durch eine Jupyter Extension realisiert. Für die prototypische Integration von DMT erfolgt dies durch DMT-Magic, welches durch das Jupyter Magic Command `dmf` innerhalb des Notebooks nutzbar wird.

Der linke Teil von Abb. 2 zeigt einen Ausschnitt eines Jupyter Notebooks bestehend aus zwei Zellen. Zelle [1] initialisiert die Nutzung von DMT für das Notebook (Schritt ❶). Dies geschieht durch das Laden der Extension DMT-Magic<sup>4</sup> und Konfigurieren der URL, unter der DMT erreichbar ist. Die Ausführung (Schritt ❶) von Zelle [2] generiert die zu bearbeitende Aufgabe in der darunter befindlichen Zellausgabe (beginnend mit „Welche Bücher wurden ...“). Die Zelleingabe besteht dabei – neben dem optionalen Kommentar – lediglich aus einem Magic Command (`%dmf`, siehe weiter unten für technische Details) und der Angabe einer Task-Id (`bibliothek:5`), die eine im E-Assessment-System hinterlegte Aufgabe referenziert<sup>5</sup>. Durch das Ausführen von Zelle [2] wird letztendlich die Zellausgabe produziert (Schritt ❷), die das eigentliche User Interface zur Eingabe der Lösung durch die Lernenden bereitstellt. Im Beispiel von Abb. 2 ist die Aufgabe vom Typ SELECT, so dass – analog zu Abb. 1 links – die Aufgabenstellung, eine Textbox, eine Statusleiste sowie ein Button zur Abgabe generiert werden.

---

<sup>4</sup> Die Extension muss vorher einmalig mittels `pip install dmf-magic` installiert werden, wobei auch abhängige Pakete, wie z. B. `ipywidgets` für Jupyter Widgets, automatisch bei Bedarf installiert werden.

<sup>5</sup> Die per Task-Id referenzierte Aufgabe kann sowohl statisch sein als auch dynamisch durch das E-Assessment-System zur Laufzeit generiert werden, z. B. mittels zufälliger Parameterwerte.

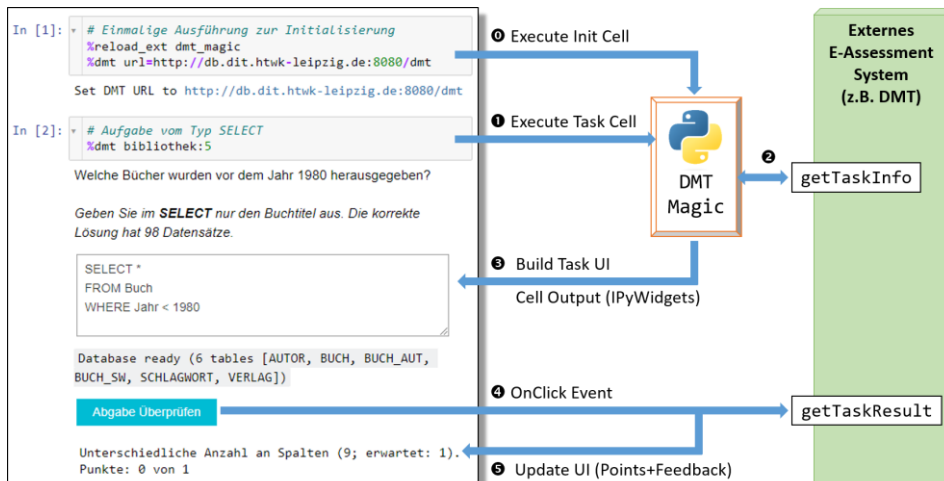


Abb. 2: Schematische Architektur und Workflow zur Kopplung eines Jupyter Notebooks (links) mit E-Assessment-System (rechts) am Beispiel von DMT mit Hilfe von DMT-Magic (mittig)

Zur dynamischen Generierung des User Interface sendet DMT-Magic einen `getTaskInfo`-Request an das E-Assessment-Tool DMT (Schritt 2), welches mit der übergebenen Task-Id alle zum Anzeigen der Aufgabe notwendigen Informationen, u.a. Aufgabenstellung und -typ, zurückliefert. Daraufhin kann das User Interface unter Verwendung von Jupyter Widgets, einer Sammlung interaktiver Steuerelemente zur Erstellung grafischer Benutzeroberflächen, gerendert werden. Dabei wird insbesondere dem Button „Abgabe überprüfen“ dynamisch eine `OnClick`-Methode zugewiesen, welche die eingegebene Lösung (im Beispiel die in der Textbox eingegebene SQL-Query) wiederum in einem `getResult`-Request an DMT sendet (Schritt 4), welches Bewertung und Feedback zurückliefert. Dieses wird dann unterhalb des Buttons angezeigt (Schritt 5).

Wie in Abb. 2 ersichtlich, muss das externe E-Assessment-System – in unserem Beispiel der Data Management Tester – lediglich die zwei Funktionen `getTaskInfo` und `getResult` bereitstellen, um in das Jupyter Notebook integriert werden zu können. Die Implementation der Anbindung an Jupyter Notebooks erfolgt mit Hilfe eines *Jupyter Magic Commands* oder kurz einer *Magic*. Solche Magics bieten eine einfache Möglichkeit, die Funktionalität des Notebook Kernels zu erweitern. Sie werden durch das Präfix `%` (siehe Zelle [2] in Abb. 2) bzw. `%%` gekennzeichnet und leiten den Inhalt der Zeile bzw. Zelle an die zugehörige Magic-Klasse weiter. Zur Implementation der DMT-Magic wurde eine Unterklasse von Magics gebildet, welche entsprechende Funktionen implementiert, u.a. zur Kommunikation mit DMT via HTTP-Rest-Requests, wobei die Daten im JSON-Format ausgetauscht werden. Um eine externe Magic innerhalb eines Notebooks zu verwenden, muss sie zunächst geladen werden (`%load_ext dmt_magic` bzw. `%reload_ext dmt_magic`). Darüber hinaus muss einmalig im Notebook die URL des DMT-Servers spezifiziert werden (`%dmt url=...`).

## 5 Zusammenfassung und Ausblick

Dieser Beitrag präsentierte einen generischen Ansatz für interaktives E-Assessment mit Jupyter Notebooks. Durch die Verwendung von Jupyter Magic Commands ist es möglich, externe E-Assessment-Systeme in das User Interface der Notebooks zu integrieren, sofern die Systeme entsprechende Schnittstellen bereitstellen. Die Integration ermöglicht ein interaktives E-Assessment während der Bearbeitung des Notebooks, so dass flexible Einsatzmöglichkeiten für Notebooks resultieren, u.a. für interaktive Vorlesungen oder als digitales Übungsblatt. Der vorgestellte Ansatz wurde prototypisch für das E-Assessment-System DMT implementiert. Zukünftige Arbeiten umfassen die Anbindung weiterer Systeme, auch unter Berücksichtigung von Sicherheitsaspekten durch Authentifizierungsverfahren, sowie die Evaluation des Ansatzes im Rahmen der Lehrveranstaltungen.

### Literaturverzeichnis

- [Go21] González-Carrillo, C. D.; Restrepo-Calle, F.; Ramírez-Echeverry, J. J.; González, F. A.: Automatic Grading Tool for Jupyter Notebooks in Artificial Intelligence Courses. *Sustainability* 13/21, 2021.
- [Jo20] Johnson, J. W.: Benefits and Pitfalls of Jupyter Notebooks in the Classroom. In: *Proceedings of the 21st Annual Conference on Information Technology Education. SIGITE '20*, 2020.
- [Ju19] Jupyter et al.: nbgrader: A Tool for Creating and Grading Assignments in the Jupyter Notebook. *Journal of Open Source Education* 2/16, 2019.
- [Kl16] Kluyver, T. et al.: Jupyter Notebooks – a publishing format for reproducible computational workflows. In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. 2016.
- [Ma20] Manzoor, H.; Naik, A.; Shaffer, C. A.; North, C.; Edwards, S. H.: Auto-Grading Jupyter Notebooks. In: *Proceedings of the 51st ACM Technical Symposium on Computer Science Education. SIGCSE '20*, 2020.
- [Na08] Narciss, S.: Feedback Strategies for Interactive Learning Tasks. In: *Handbook of Research on Educational Communications and Technology*, 2008.
- [PGA21] Panyahuti; Ganefri; Ambiyar: Jupyter Notebook-Based Diagnostic E-Assessment Model for Novice Programmer. *Turkish Journal of Computer and Mathematics Education* 12/10, 2021.
- [THS23] Tröbs, E.; Hagedorn, S.; Sattler, K.-U.: JPTest - Grading Data Science Exercises in Jupyter Made Short, Fast and Scalable. In: *BTW*. 2023.
- [TK21] Thor, A.; Kirsten, T.: Das E-Assessment-Tool DMT. *Datenbank-Spektrum* 21/1, 2021.