Daniel Schiffner (Hrsg.): Proceedings of DeLFI Workshops 2018
co-located with 16th e-Learning Conference of the German Computer Society (DeLFI 2018)
Frankfurt, Germany, September 10, 2018

Twinning – Ein Ansatz zum veranstaltungsübergreifenden Sharing von Testitems

Daniel Schiffner[1], Florian Horn[1] und Ralf Dörner[2]

Abstract: The creation of high quality test items is a complex and time consuming task. Furthermore, the exchange of those items is not simple. We propose item twinning, where existing test items are used to create tailored variants. Along with exchangeability and similarity, these items become comparable. We have tested our method with a pool of test items stemming from a computer science lecture, which will be used at various universities.

Kurzfassung: Die Erstellung von hochwertigen Testitems bildet einen komplizierten und aufwändigen Prozess. Weiterhin ist der Austausch von Fragen nicht einfach möglich. Wir adressieren diese Herausforderungen durch die Methode des Testitem-Twinning. Dabei werden von existierenden Items Veranstaltungsspezifische Varianten der Fragen erstellt. Neben der Übertragbarkeit stehen insbesondere die Ähnlichkeit und dadurch die Vergleichbarkeit der Wissensüberprüfung im Fokus. Wir beschreiben, wie wir aus einem existierenden Pool für die Einführung in die Informatik Testitem-Twins erstellt haben, welche zukünftig in verschiedenen Hochschulen eingesetzt werden.

Keywords: Testitems, Hochschulübergreifender Einsatz, Ähnlichkeit

1 Einleitung

Sammlungen von Testitems zu Lehrinhalten haben unterschiedliche Einsatzgebiete in der Lehre an Hochschulen. Zum einen stellen Sie eine Entlastung für die Lehrenden dar, die auf Basis dieser Sammlung effizient neue Prüfungen zusammenstellen können. Zudem kann z.B. bei Multiple-Choice-Tests der Korrekturaufwand reduziert werden, da die Antworten zu Testitems zum Teil einfacher oder sogar vollständig automatisiert ausgewertet werden können. Bei wiederholtem Einsatz können Lehrende zudem Prüfungen standardisierter und Notengebung vergleichbarer machen, etwa in dem Meta-Informationen zum Schwierigkeitsgrad eines Testitems aufgezeichnet werden. Für Lernende erlauben Sammlungen von Testitems ein Selbst-Assessment und können zur Prüfungsvorbereitung eingesetzt werden. Bekannte Beispiele sind Sammlungen von Fragen zur Führerscheinprüfung oder Sammlungen von Original-Prüfungsfragen im Physicum der Medizinerausbildung.

[1] Goethe Universität, studiumdigitale, Frankfurt, <schiffner, horn>@studiumdigitale.uni-frankfurt.de
[2] Hochschule RheinMain, Wiesbaden, ralf.doerner@hs-rm.de

Im Bereich eLearning haben Sammlungen von Testitems eine besondere Bedeutung, da sie als Grundlage für die automatisierte Erstellung und Auswertung von Selbst-Assessments und Prüfungen dienen. Moderne Ansätze erlauben eine automatisierte Adaption des Schwierigkeitsgrades einer Prüfung durch Auswahl von Testitems basierend auf den bisherigen Antworten und können so z.B. die Prüfungszeit verkürzen und eine Unterforderung als auch Überforderung des Prüflings verhindern.

Unser Beitrag in diesem Artikel ist die Vorstellung eines Ansatzes, wie verschiedenen Hochschullehrer ein Sharing ihrer Sammlungen von Testitems ermöglicht werden kann. Dieses Sharing führt zu größeren Sammlungen, die auch für eLearning-Systeme relevant sind. Dazu identifizieren wir potenzielle Vorteile als auch Hindernisse, die einem Sharing momentan entgegenstehen im folgenden Abschnitt. Abschnitt 3 stellt unseren Lösungsansatz vor. Ein Fallbeispiel der Anwendung dieses Ansatzes im Bereich der Programmierausbildung in Informatikstudiengängen an Hochschulen folgt in Abschnitt 4. Dieser Abschnitt enthält auch eine Diskussion unserer bisherigen Erfahrungen. Der Artikel endet schließlich mit einer Zusammenfassung und einem Ausblick.

2 Sharing von Testitems – Potenziale und Hindernisse

Es existiert umfangreiche Literatur zur Erstellung von Testitems in der Hochschullehre [RA17]. Dabei werden auf Basis unterschiedlicher Theorien wie z.B. der Cognitive Load Theory Wege aufgezeigt, wie die Validität von Testitems gesteigert werden kann [GPE15] oder wie man durch Nutzung von verbalen oder visuellen Testitems dem Lernstil des Lerners entgegen kommen kann [BH15]. Die mit der Erstellung von Testitems einhergehende Standardisierung von Prüfungen erleichtert Learning Analytics [BI14]. So kann beispielsweise die Güte von Prüfungen substanzieller bewertet werden [Si18], das Verhalten von Lernern wird besser modellierbar [BKA15] oder neue Testitems lassen sich generieren [SW17].

Welche Vorteile lassen sich erwarten, wenn Sammlungen von Testitems unter Hochschullehrern ausgetauscht und gemeinsam genutzt werden? (1) Die Anzahl der Testitems vergrößert sich. Dies ist insbesondere für Learning Analytics und Machine Learning vorteilhaft, die von größeren Anzahlen profitieren, z.B. weil Aussagen mit höherer Wahrscheinlichkeit getroffen werden können. Außerdem wird das Problem umgangen, dass Tests bei kleiner Testitemzahl vorhersehbar werden und beim Lernen weniger die zu lernenden Inhalte als das Testverfahren im Zentrum der Bemühungen der Lerner stehen. (2) Die Arbeit bei der Erstellung von Testitems kann auf verschiedene Personen verteilt werden. (3) Es findet ein Austausch unter den Lehrenden statt, der zu inhaltlicher Aktualisierung von Lehrveranstaltungsinhalten führen kann. (4) Es können Bewertungsmaßstäbe und Niveaus gefördert werden, die Lehrveranstaltungs- und Lehrkraftübergreifend sind. (5) Ein Testitem wird von einer höheren Anzahl von Lernern bearbeitet, da es nicht nur in der Prüfung eines Hochschullehrers, sondern von mehreren Hochschullehrern genutzt werden kann. Dies hat auch wieder positive Auswirkungen auf

Learning Analytics. So kann z.B. der Schwierigkeitsgrad eines Testitems genauer kalibriert werden, wenn dies häufiger eingesetzt wird.

Wenn es derart umfassende Vorteile gibt, warum ist ein Sharing von Testitems unter Hochschullehrern bislang weiterverbreitet? Ein Grund besteht darin, dass es an Hochschulen keine festen Lehrpläne oder Curricula gibt, die übergreifend gelten. Selbst Lehrveranstaltungen, die der gleichen Modulbeschreibung für einen spezifischen Studiengang folgen, weisen oft eine hohe Varianz in den behandelten Inhalten und gesetzten Schwerpunkten auf. Dies ist nicht zuletzt mit Blick auf die Freiheit der Lehre gewünscht. Allerdings verhindert es ein Sharing, da bestimmte Testitems fremder Lehrkräfte nicht dem eigenen Prüfungsstoff entsprechen, Begriffe anders definiert oder Aufgabenstellungen für die Studierende ungewohnt, weil stark von den Aufgabenstellungen der Übungen abweichend, empfunden werden können. Ein weiterer Grund besteht darin, dass ein Sharing aufgrund nicht vorhandener Infrastruktur nicht unterstützt wird. Weiche Faktoren sind in der Kultur an vielen Hochschulen zu finden, die gegenseitiges Besuchen von Lehrveranstaltungen und Austausch nicht befördern.

3 Lösungsansatz Twinning

Unserem Lösungsansatz liegt die Erkenntnis zugrunde, dass man Testitems aufgrund der großen Variabilität von Lehrveranstaltungen in der Hochschullehre nicht ohne jegliche Bearbeitung für eine eigene Sammlung von Testitems übernehmen kann. Die Grundidee ist daher, dass beim Sharing von Testitems fremde Testitems zunächst überarbeitet werden, wodurch ein Testitem-Twin entsteht. Die Bearbeitung ist dabei explizit auf Begrifflichkeiten oder syntaktische Anpassungen beschränkt. Ziel ist es, Ähnlichkeit in Bezug auf den Schwierigkeitsgrad, das zu überprüfende Kompetenzniveau und den Inhalt sicherzustellen. Wird also ein von Lehrkraft D erstelltes Testitem T(L,D) aus einer Lehrveranstaltung L für eine Lehrveranstaltung L‘ bei Lehrkraft D‘ übernommen, dann erzeugt D‘ durch Überarbeitung ein möglichst zu T(L,D) ähnliches Testitem T‘(L‘,D‘).T(L,D) und T‘(L‘,D‘) nennen wir Testitem-Twins. Durch getroffenen Einschränkungen ist das Twinning transitiv: Erstellt eine Lehrkraft D‘‘ für Lehrveranstaltung L‘‘ ein Testitem T‘‘(L‘‘,D‘‘) aus T‘(L‘,D‘), dann sind auch T‘‘(L‘‘,D‘‘) und T(L,D) Testitem-Twins. Damit kann ein Testitem in mehreren Ausprägungen vorliegen.

Die Twin-Relation ist bei eLearning Systemen abzubilden. Dadurch können bei Learning Analytics alle Testitem-Twins hinsichtlich der über sie gesammelten Daten als Äquivalenzklassen aufgefasst werden. Es ist für das jeweilige Ziel im Bereich Learning Analytics zu entscheiden, ob man Daten aller Testitem-Twins zusammenlegen darf oder ob z.B. eine Normierung hinsichtlich Lehrkraft oder Lehrveranstaltung zu erfolgen hat. Ebenfalls empfehlen wir einen Reviewprozess, um die Testitem-Twins einer Qualitätsprüfung zu unterziehen. Dieser Prozess ist dabei nicht verpflichtend, aber durch

die offene Diskussion unter Lehrkräften sind unklare Formulierungen oder schlecht gewählte Distraktoren klarer zu erkennen.

Der Prozess des Sharings und dem dabei durchgeführten Twinnings kann durch Softwarewerkzeuge unterstützt werden. Diese können als Features die Suche nach existierenden Testitem-Twins oder fehlenden Testitem-Twins unterstützen. Für das Twinning ist die Sammlung von Meta-Daten essentiell [We97, IEEE02]. Neben dem intendierten Kompetenzniveau, sind auch Zielgruppe und eine Verortung in einer Taxonomie oder Ontologie hilfreich. Hierbei können graphische Darstellungen von Ontologien Lehrkräfte unterstützen, Testitems von Dritten für Bereiche zu finden, die eine Lehrkraft bislang selbst wenig abdeckt. Ebenso ist es so einfach möglich, neue Fragen in die Ontologien einzupflegen und den Reviewprozess zu starten. Durch eine dezentrale Infrastruktur kann eine Lehrkraft individuell entscheiden, ob ihre Fragen in der Gesamtontologie zu finden sind oder in den Reviewprozess eingehen. Um eine möglichst große Menge an Fragetypen zu unterstützen ist QTI als Austauschstandard sinnvoll.

4 Fallbeispiel aus der Programmierausbildung

Wir haben Twinning an einem Fallbeispiel entwickelt, bei dem vier Lehrkräfte aus zwei Universitäten und zwei Hochschulen für Angewandte Wissenschaften Testitems für die grundständige Lehrveranstaltung „Programmieren“ im ersten Semesters eines Informatikstudiums teilen. Hier wurde ein hohes Potenzial für Sharing vermutet, da die in einer derartigen Lehrveranstaltung behandelten Inhalte und angestrebten Kompetenzen hochschulübergreifend eine höhere Ähnlichkeit aufweisen als z.B. in Spezialveranstaltungen oder Wahlveranstaltungen. Außerdem hat die Lehrveranstaltung „Programmieren“ formale Sprachen zum Gegenstand, so dass nicht nur einfache Multiple-Choice Items verwendet werden können, sondern auch komplexere Freitextaufgaben, deren Lösung in einer formalen Sprache (d.h. durch Angabe eines Programms) notiert werden und damit auch einer automatisierten Bewertung zugänglich sind.

Insgesamt wurden 393 Items in einen Pool eingebracht und von den Lehrkräften bewertet. Dabei wurde explizit geprüft, ob die „fremden“ Testitems auch an die eigenen Studierenden geben werden kann. Die verwendeten Items sind größtenteils Programmiersprachenspezifische Fragen, welche in Python und Java vorliegen. Aber auch Programmiersprachenunabhängige Fragen finden sich im Pool, welche in einzelnen Begrifflichkeiten angepasst wurden. Alle Fragen in dem Pool wurden nach dem Prinzip der Testitem-Twins in die jeweilige Veranstaltung übertragen. Die resultierenden Items mussten anschließend in einem Reviewprozess bezüglich ihrer Ähnlichkeit bewertet werden.

Um diese Entscheidung zu vereinfachen, wurde zur Kategorisierung der Ähnlichkeit eine Skala entwickelt. Diese spiegelt auf 4 Ebenen wieder, wie einfach ein Item „übersetzt“ werden kann. Level 0 entspricht keinen bis trivialen Änderungen. Als Beispiel kann hier der Terminator eines Ausdrucks genannt werden. In Python ist dies durch einen einfachen Return kodiert, wohingegen bei Java ein Semikolon verwendet wird. Level 1 sind

beispielsweise Schleifen Konstrukte, welche bei Java und Python unterschiedlich sind, aber gleiche Kompetenzen erfordern. Level 2 schließt syntaktische und semantische Änderungen mit ein, wohingegen Level 3 starke inhaltliche Anpassungen erfordert, wie z.B. das Prinzip der abstrakten Klassen in Python.

In Level 0 fielen 50.13% der Items, und konnten damit direkt geteilt werden. Die komplexeren Anpassungen wurden durch die Lehrkräfte durchgeführt, mit dem Ziel den geprüften Stoff abzudecken. Die Verteilung der Fragen in die Kategorien ist in Tabelle 1 zusammengefasst.

Kategorie	**Anzahl Fragen**	**Prozent**
Level 0	197	50.13 %
Level 1	36	9.16 %
Level 2	35	8.91 %
Level 3	125	31.81 %

Tab. 1: Verteilung der Fragen in Kategorien

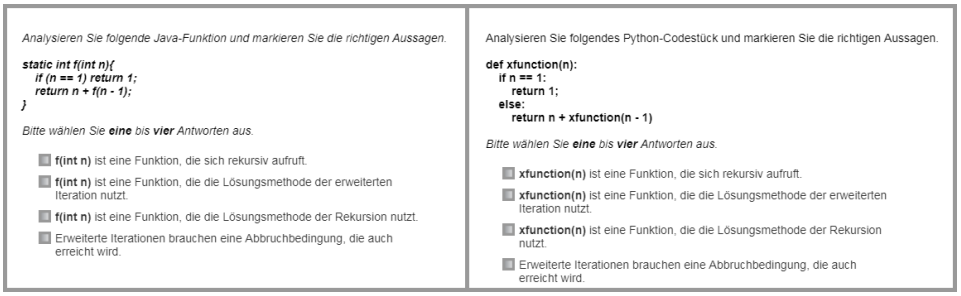
Auch wenn alle Fragen an die Standorte anpassbar sind, können diese dennoch nicht immer direkt genutzt werden. Der Grund liegt neben der Ähnlichkeit auch in der unterschiedlichen Verwendung oder der starken Verknüpfung zu einer Vorlesung. Teilweise werden auch Inhalte nicht gelehrt (Stichwort: Freiheit der Lehre) oder finden erst in höheren Semestern statt. Insgesamt konnten von den 268 Fragen 150 Fragen in den Gesamtpool aufgenommen werden.

Durch den direkten Austausch der Fragen sind schlechte oder unklare Formulierungen besser reflektiert worden, wodurch die Qualität der Testitem-Twins erhöht wurde. Herausforderungen ergaben sich in dem Reviewprozess selbst, sowie der dadurch entstandenen asynchronen Bearbeitung. Hier ist eine Unterstützung durch entsprechende eLearning Systeme sinnvoll, insbesondere bei der Bestimmung der Ähnlichkeit bzw. dem Auffinden von Fragen.

Die Fragen wurden mit dem Hintergrund als Testitem-Twins konvertiert, um in einer Kalibrierungsstudie hochschulübergreifend verwendet zu werden. Dafür war es notwendig, eine starke Formalisierung des Erscheinungsbildes zu garantieren. Das verwendete Testsystem unterstützte die Lehrkräfte dabei nicht sehr gut, wodurch der positive Gesamteindruck geschmälert wurde. In einem kurzen qualitativen Interview sprachen sich alle 4 Lehrkräfte dennoch für das Testitem-Twinning aus, und empfanden das Übertragen in einen anderen Kontext durch eine Vorlage als sehr angenehm bzw. als sehr unterstützend. Gerade bei der Wahl der Distraktoren für Single Choice oder Multiple Choice Fragen war das „übernehmen der Vorlage/Twins“ eine „Arbeitserleichterung“. Durch die Vorlage konnte „[…] mehr auf die Inhalte konzentriert werden“. Trotz des

vermehrten Austausches fanden sich viele formale Fehler, welche explizit korrigiert werden mussten. Dieser Punkt sollte ebenfalls durch ein entsprechendes System gesichert werden.

In Abbildung 1 und Abbildung 2 sind zwei Beispiele von Testitem-Twins zu sehen. Die Fragen wurden aus einer Vorlesung entnommen und entsprechend konvertiert. Das Beispiel in Abbildung 1 zeigt eine sehr gut übertragbare Variante, wohingegen das zweite Beispiel klar zeigt, dass die Komplexität der Frage nicht erhalten werden konnte. Diese wurde im Kategoriensystem auch als Level 3 eingestuft.



Analysieren Sie folgende Java-Funktion und markieren Sie die richtigen Aussagen.

```
static int f(int n){
   if (n == 1) return 1;
   return n + f(n - 1);
}
```

*Bitte wählen Sie **eine** bis **vier** Antworten aus.*

- **f(int n)** ist eine Funktion, die sich rekursiv aufruft.
- **f(int n)** ist eine Funktion, die die Lösungsmethode der erweiterten Iteration nutzt.
- **f(int n)** ist eine Funktion, die die Lösungsmethode der Rekursion nutzt.
- Erweiterte Iterationen brauchen eine Abbruchbedingung, die auch erreicht wird.

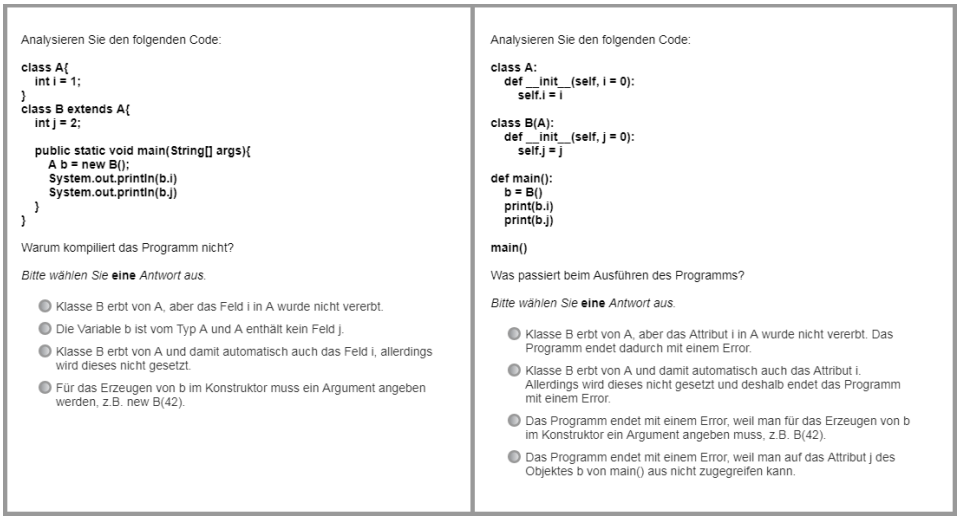
Analysieren Sie folgendes Python-Codestück und markieren Sie die richtigen Aussagen.

```
def xfunction(n):
   if n == 1:
      return 1;
   else:
      return n + xfunction(n - 1)
```

*Bitte wählen Sie **eine** bis **vier** Antworten aus.*

- **xfunction(n)** ist eine Funktion, die sich rekursiv aufruft.
- **xfunction(n)** ist eine Funktion, die die Lösungsmethode der erweiterten Iteration nutzt.
- **xfunction(n)** ist eine Funktion, die die Lösungsmethode der Rekursion nutzt.
- Erweiterte Iterationen brauchen eine Abbruchbedingung, die auch erreicht wird.

Abb. 1: Beispiel für eine gute Itemübertragung. Links ist die Java, rechts die Python Frage. Abgesehen von kleineren Semantischen und Syntaktischen Anpassungen stimmen die Fragen komplett überein.



Analysieren Sie den folgenden Code:

```
class A{
   int i = 1;
}
class B extends A{
   int j = 2;

   public static void main(String[] args){
      A b = new B();
      System.out.println(b.i)
      System.out.println(b.j)
   }
}
```

Warum kompiliert das Programm nicht?

*Bitte wählen Sie **eine** Antwort aus.*

- Klasse B erbt von A, aber das Feld i in A wurde nicht vererbt.
- Die Variable b ist vom Typ A und A enthält kein Feld j.
- Klasse B erbt von A und damit automatisch auch das Feld i, allerdings wird dieses nicht gesetzt.
- Für das Erzeugen von b im Konstruktor muss ein Argument angeben werden, z.B. new B(42).

Analysieren Sie den folgenden Code:

```
class A:
   def __init__(self, i = 0):
      self.i = i

class B(A):
   def __init__(self, j = 0):
      self.j = j

def main():
   b = B()
   print(b.i)
   print(b.j)

main()
```

Was passiert beim Ausführen des Programms?

*Bitte wählen Sie **eine** Antwort aus.*

- Klasse B erbt von A, aber das Attribut i in A wurde nicht vererbt. Das Programm endet dadurch mit einem Error.
- Klasse B erbt von A und damit automatisch auch das Attribut i. Allerdings wird dieses nicht gesetzt und deshalb endet das Programm mit einem Error.
- Das Programm endet mit einem Error, weil man für das Erzeugen von b im Konstruktor ein Argument angeben muss, z.B. B(42).
- Das Programm endet mit einem Error, weil man auf das Attribut j des Objektes b von main() aus nicht zugegreifen kann.

Abb. 2: Beispiel für schlechte Itemübertragung. Links ist die Java, rechts die Python Frage. Wie zu sehen ist stimmen selbst die Distraktoren nicht mehr überein.

5 Zusammenfassung und Ausblick

Die Erstellung eines Fragenpools und der Austausch zwischen Lehrkräften bietet ein interessantes Forschungsfeld. Neben der Erstellung der notwendigen Infrastruktur ist auch die Übertragbarkeit von Fragen wichtig. Durch das Twinning bietet sich die Möglichkeit, effizient hoch qualitative Fragen zu erstellen. Durch die starke Einschränkung auf die didaktischen Konzepte und minimale Änderungen ist eine Vergleichbarkeit der Fragen sichergestellt.

Für ein laufendes Forschungsprojekt haben wir dazu im Rahmen der Erstsemesterveranstaltungen von verschiedenen Informatikstudiengängen Fragen gesammelt und nach dem Twinning-Prinzip konvertiert. Danach wurde eine Kategorisierung verwendet, um die Ähnlichkeit der Items zu bestimmen. Weiterhin wurden die Testitems einem formalen Review unterzogen, um inhaltliche sowie formale Fehler zu identifizieren. Hierbei fanden sich wenige inhaltliche Probleme und die beteiligten Lehrkräfte empfanden die Idee als gute Unterstützung.

Weitere Untersuchungen werden sich auf die Ähnlichkeit der Testitem-Twins beziehen und versuchen zu identifizieren, welchen Einfluss die gemachten Veränderungen auf die Items haben. Ebenfalls wird die Kategorisierungsskala detaillierter untersucht und versucht eine formalisierte Form davon zu erstellen.

Im Hinblick auf Learning Analytics muss untersucht werden, wann Daten von Nutzern vereinheitlicht werden können bzw. wann eine Differenzierung notwendig ist. Die Kombination von verschiedenen Visualisierungen, z.B. mit der verwendeten Ontologie, bietet viele Möglichkeiten, den Wissenszustand der Lernenden effizient zu repräsentieren.

Literaturverzeichnis

[BH15] Bacon, Donald R., and Steven W. Hartley. "Exploring antecedents of performance differences on visual and verbal test items: Learning styles versus aptitude." *Marketing Education Review* 25.3 (2015): 205-214

[BI14] Baker, Ryan Shaun, and Paul Salvador Inventado. "Educational data mining and learning analytics." *Learning analytics*. Springer New York, 2014. 61-75

[BKA15] Bedek, M. A., Kickmeier-Rust, M. D., & Albert, D. (2015). Formal concept analysis for modelling students in a technology-enhanced learning setting. In ARTEL@ EC-TEL (pp. 27-33).

[GPE15] Gillmor, Susan C., John Poggio, and Susan Embretson. "Effects of reducing the cognitive load of mathematics test items on student performance." *Numeracy* 8.1 (2015): 4

[IEEE02] IEEE. "IEEE Standard for Learning Object Metadata." *IEEE Std 1484.12.1-2002. IEEE*, 2002. 1-40

[RA17] Rodriguez, Michael, and Anthony Albano. *The College Instructor's Guide to Writing Test Items: Measuring Student Learning*. Routledge, 2017.

[SW 17] Schweighart, M. "Using item response theory to generate an item pool for an e-learning-system." *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*. ACM, 2017.

[Si18] Singh, Neerja. "Impact of Learning Analytics on the Assessment of a Curriculum-Based Test." *Impact of Learning Analytics on Curriculum Design and Student Performance*. IGI Global, 2018. 56-70

[We97] Weibel, Stuart. "The Dublin Core: A Simple Content Description Model for Electronic Resources." Bulletin of the American Society for Information Science and Technology 24(1). Wiley Online Library, 1997. 9-11.