An Investigation into Smart Textbooks

Kyle Robbertze
RBBKYL001@myuct.ac.za
University of Cape Town
Cape Town, South Africa

ABSTRACT

This paper investigates the digital smart textbook platforms currently available and generalises these into a set of features that are required for a digital smart textbook platform to be successful. These features include visual metaphors, such as highlighting and making notes in the margins, that users expect from using hard-copy textbooks, and smart features, such as question and answer generation, concept summaries and in-line definitions. A digital smart textbook is backed by a knowledge graph and this paper investigates ontologies as the representation used for such a knowledge graph, guided by existing implementations.

CCS CONCEPTS

• **Information systems → Enterprise applications**; *Digital libraries and archives.*

KEYWORDS

smart textbooks, ontologies, knowledge representation, textbook use, digital textbooks

1 INTRODUCTION

Textbooks have long been used in the classroom to convey knowledge. With the progression of the digital age, textbooks frequently have also become digital. However, these digital textbooks often take the form of a scanned or typed copy of the physical copy and thus, do not take advantage of new techniques that digital books bring. These techniques include context-based navigation and automatic question and answer generation. Including new digital techniques has been shown to improve students' performance in homework tasks by approximately 10% [5]. "Inquire Biology" is the only example of such a knowledge-driven textbook. It is proprietary and is very specific to a single edition of a single textbook.

Thus, there is a lack of a generalised smart textbook platform that can be used to interact with a wide variety of textbooks. Creating such a platform has two distinct sections: the textbook representation and the generation of questions and answers as required by the user of the platform. This is a literature review into the existing solutions and research around digital smart textbook platforms and related fields.

A knowledge-driven smart textbook platform can be further broken up into several components. Section 2 addresses displaying the textbook's contents such as text and diagrams. In Section 3, the report addresses representing and storing the knowledge contained in the textbook, while Section 4 investigates how students interact with smart textbooks. The results of these components and what they mean for knowledge-driven smart textbooks are discussed in Section 5. Finally, Section 6 concludes what features a smart textbook platform must have in order to be successful.

2 DIGITAL TEXTBOOKS

Textbooks have long been used by teachers and students alike to cover material and plan lessons [13, 25]. They are considered as an essential learning tool, no matter the manner in which it is used in the classroom [30]. They have been shown to directly contribute to students' academic performance in developing countries. This is because a textbook provides a more comprehensive coverage of the content than a teacher has time to do in the classroom [22].

In recent years, universities are increasingly using digital textbooks to offer cost-effective, efficient and accessible resources to students [31]. There have been many studies into how digital textbooks have changed education. South African students read digital textbooks at the same speed as they read hard-copies, while maintaining the same comprehension levels as when there were using hard-copies [32]. However, Sackstein et al. (2015), hypothesized that reading speed could be due to students' prior exposure to the medium that they were reading on [32].

Digital textbooks are easier to update and distribute via the internet [7, 10]. Frequent updates are required for subjects such as taxation tables and case-law where new standards are often being set. This easy updating also reduces the number of different editions of textbooks students' need to purchase, reducing the cost incurred while studying, potentially changing their preferences [11]. A web-based platform is able to provide updates quickly to all its users because all textbooks and associated data are located in one location controlled by the maintainer of the platform.

2.1 Student Preference

Weisberg (2011) investigated the question of preference between digital textbooks and hard-copies and found that students at Suffolk University in Boston preferred the use of digital textbooks over hard-copies [40].

However, Jeong (2012) found that high school students in South Korea preferred hard-copies of textbooks over their digital representations [18]. This difference suggests that preference may be dependent on a variety of factors such as familiarity with the medium, students' cultural attitudes and how extensively the knowledge is studied [31]. While both Weisberg and Jeong contrasted digital textbooks and hard-copies, they excluded the possibility that students may use both types simultaneously. This gap was covered by Du Plessis and Wiese (2014) in a study at the University of Pretoria. They found that while only 15% of students preferred to use the digital textbook exclusively, 43% preferred to use both the hard-copy and digital textbook concurrently [11].

While these preferences are important and should be accounted for, the move to digital textbooks is inevitable with the increasingly digital world. Thus, educators should account for and seek to incorporate digital textbooks [9].

2.2 "Inquire Biology"

While there is a growing trend of using structured knowledge to drive learning [1], the majority of these efforts are focused on the personalisation of course content to match students' learning speed, such as ONTODAPS [27], and the search and retrieval of content, such as the Curriculum Builder in the Federated Virtual University of the Europe of Regions (CUBER) [29].

Chaudhri et al. (2013) investigated the use of a knowledge-driven textbook "Inquire Biology" in answering homework questions. In the study of learners answering homework and test questions, they found that the use of the knowledge-driven textbook helped learners perform 10% better than those using a standard digital textbook or the hard-copy. This knowledge-driven textbook includes many smart features that aided students learning through active reading. These features include concept summaries, navigation history, asking and answering students' questions, in-line definitions of biological terms, and extended note-taking and suggested questions when highlighting text [5].

Active reading is the practice of summarizing, questioning, clarifying and predicting when learning a concept. Palinscar and Brown (1984) found that active learning significantly improved students' comprehension of the content being studied. Students were also able to generalize these gains to class tests, thus improving their academic results [28].

Another aspect of studying is active leaning, which Bonwell and Eison (1991) defined as "anything that involves students in doing things and thinking about the things they are doing" [4]. By this definition, active reading forms a part of active learning. Active learning is important as it enables students to develop higher order thinking skills and engage in activities. It involves 4 steps: experiencing, reflecting, generalizing and applying [34]. When studying a textbook, active reading enables students to perform reflection (through summarizing), generalizing (through predicting) and applying (through questioning).

In "Inquire Biology", a concept summary is provided for each biology term. This starts with a definition of the term with key facts and relationships extracted from the knowledge representation of the digital textbook. Concept summaries are independent of chapters and pull knowledge from different chapters into a single place, helping students to read actively. Active reading occurs through enabling students to clarify and summarize the content around a concept. These concept summaries are important as they clearly summarize the content without requiring the student to wade through a great deal of irrelevant information as they would if they were searching through other resources such as Wikipedia [5].

A potential limitation of a generated concept summary is that students lose any advantages gained by writing the summary themselves. Summarizing a piece of text is the first step of active reading [28]. A generated concept summary does not prevent students from creating a summary of the content themselves, although it may make students less likely to do so.

While students type in questions, "Inquire Biology" attempts to predict and suggest completed questions that they may want to ask. This reduces the chance that the question cannot be parsed by the system and reduces the variety of questions the system has to manage [5]. The answers to these questions are formatted based on the question's classification. For example comparison questions are formatted into a table with similarities and differences highlighted for various aspects of the two concepts [5]. This helps students discriminate between two concepts more easily, leading to better retention of both concepts [2]. The act of asking questions is a part of active reading and further helps students improve their understanding and retention of the textbook's subject domain [28]. It also saves students time, thus potentially improving students' ability to concentrate on the content as they can go through the process of active learning while reading the textbook [4].

In-line definitions of biological terms reduces students' cognitive load when working through the textbook by providing easy access to basic terminology [5]. Students experience high cognitive load when learning new concepts through problem-solving. Reducing this load improves concept acquisition [37]. This is a form of active reading and thus improves academic results [28]. "Inquire Biology" provides these in-line definitions by underlining the term and providing a pop-up dialogue containing the definition when the student mouses over the term. This pop-up is useful as it allows students to skip over the definition if they already know it, while giving them the opportunity to test their knowledge of the definitions if they so choose [5].

2.3 Interactive Textbooks

Miller and Ranum (2012) developed a different interactive textbook to use in teaching a first-year undergraduate Python programming course. This textbook took the form of a website with text interspersed with videos, executable code snippets and an interactive code visualizer [23]. Code visualizers and snippets are alternative methods of clarifying meaning from a digital textbook for the programming subject domain. These types of questions could be generated from the knowledge backing the smart digital textbook and displayed for certain classifications of questions. This shows that subject domains can have specific question classification that are unique.

The video snippets were produced by the authors and all had a duration of between 5 and 10 minutes. They were constrained in scope and did not necessarily cover everything that was covered in text portions of the digital textbook. These constraints ensured that students were more likely to watch the videos, because they were short, and engage with the text for the content lacking in the videos [23].

Code snippets allowed students to run and edit embedded code that was relevant to the section they were working though. This enabled them to engage actively with the programming language and thus exercise the questioning, clarifying and predicting portions of active learning [28]. It also enables students to apply the Experiencing and Applying activities of active learning [34].

The code visualizer enabled students to step through code examples, viewing the output and values of variables as the program ran. When surveyed at the end of the study, students found this code visualiser to be very useful as it allowed them to visualise the steps the program executed to arrive at the final result [23].

When surveyed at the end of the semester, students reported that they enjoyed using the interactive textbook and ranked the interactive elements provided such as running and editing the code

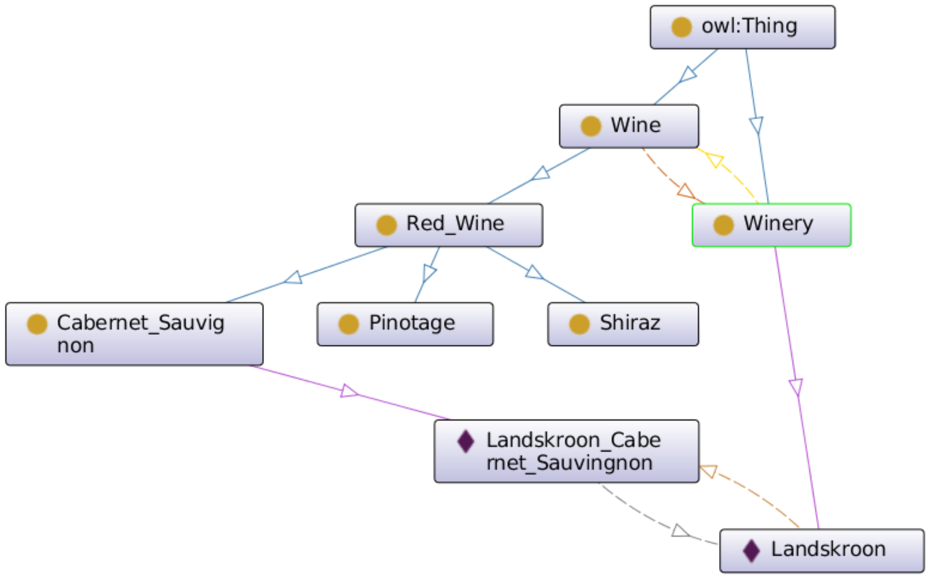

Figure 1: A simple graphical representation of an ontology created in WebProtégé [39]. The yellow dots are classes (concepts) with the solid blue arrows indicating inheritance. Dotted lines indicate relationships and purple diamonds indicate instances of classes. The purple lines indicate instances of a class.

as very important elements to their success in the course [23]. This engagement is an important part of active learning, which is shown to improve academic performance [28].

3 KNOWLEDGE REPRESENTATION

The backbone of any knowledge-driven smart textbook, is the representation and storage of the knowledge encapsulated in the textbook. This lends well to the use of an ontology which is what "Inquire Biology" has done [5].

An ontology is a formal representation of concepts and their relationships [15]. In the context of smart textbooks, the ontology is a graph with nodes representing concepts and edges representing the relationships between concepts. Figure 1 shows the basic idea of an ontology represented as a directed graph.

A graph in this context is a set of points connected with lines. These points are called nodes and the lines are edges [3]. In a smart textbook ontology, the nodes are concepts with a set of slots containing features of the concept. Concept relationships are defined by edges with pre-determined relationship identifiers that describe how the concepts are related [15].

3.1 Existing Ontology Creation Tools

There are several existing ontology creation tools that can be adapted to fit the requirements for creating textbook-knowledge graphs. These include the Automated User-Centred Reasoning and Acquisition System (AURA) [16], which is used by "Inquire Biology" and Protégé [26].

As the goal is to develop a platform for many smart textbooks, it would be useful to be able to integrate existing ontologies around the textbook's subject domain into the ontology developed for the textbook. However, this can prove to be difficult as there are often differences in schematics, semantics and scope. Differences in schematics include how data are represented, labelled and generalized. Differences in semantics involve differences between how concepts are named and units of measure used. Finally, differences in scope include how much of the domain is included and to what level of detail. Software exists to combine different ontologies using a variety of methods, however these usually require user screening before they can be used [12]. Furthermore, including additional information, like that contained in a generalised ontology, would result in concept summaries and generated questions which contained information that the student would find irrelevant [5].

Ontology creation is inherently skill intensive and thus requires a subject matter expert (SME) to construct the knowledge representation [16]. These SMEs are experts in their own fields and are not necessarily very computer-literate. Thus, the creation tool's interface must be easy to use and not require much prior knowledge about ontologies or, alternatively, be easy to learn quickly.

Protégé has a key advantage in its support for plugins. These allow users to extend Protégé to perform any needed or wanted functions that are missing from the core design [26]. Some of these plugins allow ontologies created in Protégé to be exported as a flat text file that can then be used in ontology-based applications [24]. This interoperable file format allows ontology-based applications to be independent of the creation tool and allow SMEs to create ontologies in any creation environment they are comfortable with as long as the environment can export Web Ontology Language (OWL) 2 files.

3.2 Ontology Reasoning

"Inquire Biology" interacted with the knowledge ontology using an inference system (SILK) developed specifically for the task of working with AURA. SILK contains a knowledge reasoning language and reasoner. It communicates with AURA as the back-end and allows programs to query the ontology for forward and backward inferencing [14].

The Knowledge Machine (KM) is the core reasoning engine used by AURA [16]. KM is a knowledge representation language implemented in the LISP programming language that is capable of inference, is expressive and can include English-like justifications of its conclusions [8].

Two other independent reasoners are the ELK Reasoner and FaCT++ which both offer high performance reasoning with easy use and extensibility. ELK is implemented in Java, while FaCT++ is written in C++. As they are independent reasoners, they both support being used as a reasoning library in ontology-based applications [19, 38].

3.3 Text Annotation

Ontology-based text annotation is the process of detecting ontology elements within the text, based on the provided subject domain ontology. This is done through detecting metadata and structuring the data for improved processing later [20]. It is important, as this is how the textbook and ontology are combined to be presented to the user in a smart digital textbook platform. The ontology provides semantic interpretation of the textbook [36].

The main link between an ontology and its text is a terminology, which maps concepts in the ontology to terms in the text. However, usually there is no simple one-to-one mapping between terms and concepts. According to Spasic et al. (2005), the probability of two experts to refer to the same concept with the same term is less than 20%. Another potential ambiguity is when the same term refers to multiple concepts depending on the context [36].

Laclavik et al. (2012) developed Ontea to link terms and concepts using regular expressions. These regular expressions created ontology elements that are then assigned as properties of the applied class. In their approach, each domain ontology required its own patterns in order to achieve good results. These patterns were incorporated into the ontology through a special extension class Pattern with defined properties. This approach was effective, with an average success of over 79% on recall (the ratio of correct positive predictions to total positive examples) and 53% on precision (the ratio of correct positive predictions to all predictions made by the system). The low precision result came from the simplicity of the regular expressions and thus depends heavily on how the regular expressions are defined [20].

This terminology link makes the text annotation very sensitive to the source textbook's language. If experts using the same language use different terms [36], then experts using two different languages are very likely to so too. This is another reason for the ontology to be specific to the textbook as argued by Chaudhri et al. (2013) [5].

4 USER INTERACTION

Digital textbooks need to retain the visual metaphors already familiar to users of hard-copies [21]. These metaphors include breaking information up into pages and retaining the table of contents and indexes. This is due to the fact that users approach digital textbooks with the expectations formed from their experiences with hard-copies [41].

Simpson and Nist (1990) found that students who commented text through highlighting and making notes in the margin performed 6% better on average than students who just reread the textbook. However, students required training to comment text correctly in an efficient and productive manner. This training included the background behind text notation, active practice, as well as peer-evaluation and guidance using a checklist. After training, students were given a 3 000 word excerpt and were tested on it a week later. The students who used text notation while studying reported that most of their study time was spent testing themselves on their notations instead of trying to decide what was important and what was superfluous [33]. This training would be made redundant through the use of a smart digital textbook, as it can be done through suggesting questions and definitions when the students begin to comment sections of the text.

Landoni and Gibb (2000) found that digital textbooks need to implement these notation features for them to be considered successful. This is due to the fact that students expect the notation visual metaphor from hard-copies [21] and this is a part of active learning [33].

Henke (2002) corroborated these findings in his survey of 163 people which looked at their experiences with digital textbooks. The results ranked opening the digital textbook to the last opened page as the most important feature, with text search, bookmarks and personalisation ranking close behind. Furthermore, easy access to user defined notes (comments, margin notes and the like) and highlighting text ranked 12 and 13 respectively [17].

5 DISCUSSION

Digital smart textbooks are changing the paradigm of textbook use through the promotion of active reading [31] and their use is only set to grow [9]. While there is no definitive study on the preferences of South Africa university students, most have worked with digital textbooks [11] and universities continue to promote their use [31]. The addition of interactive and other smart features improves student engagement and comprehension of the subject domain [5, 23].

Access to digital resources in developing countries cannot be assumed to be fair across the population. South Africa has a very large gap between the urban areas with easy access to digital resources and the rural areas where there is often none at all [10]. This poses a challenge to the adoption of digital smart textbooks, as they can be inaccessible for those in rural areas. Universities provide internet access to their students in order for them to access resources critical to their success at the institution [6]. This can aid their access to a web-based digital smart textbook platform. An alternative option would be to distribute the textbook as an offline application on a flash drive or mobile phone. However, both remove the advantage of easy updating of content and prevent the large processing requirements to be offloaded to more powerful servers.

The price of digital textbooks may also be enticing to students. The majority of students would rather buy a digital textbook if it was cheaper than a second-hand hard-copy. This could be the deciding factor for students to select using a digital textbook, as opposed to its feature set [11]. This should be taken into account when designing a digital smart textbook platform, as the per user hosting and management cost should be less than that of a second-hand hard-copy textbook.

Distribution of digital textbooks is also logistically easier than hard-copies. Universities can provide online copies to students which they can then download and access offline [10]. Online digital smart textbook access also provides the possibility to use more powerful remote servers to process the resource-intensive parts such as ontology interaction and question and answer generation, which allows users of less-powerful computing devices such as mobile phones to access the same content making it more accessible to the masses [35]. Thus, a web-based smart digital textbook would need a mobile-friendly application as well as ubiquitous access from other devices the student may have.

6 CONCLUSIONS

A generalised digital smart textbook platform would provide students with a way of interacting with course subject domains in a new way. These knowledge-driven features would ensure students' active reading of the subject domain with the subsequent improvement in academic performance.

These features must include the standard metaphors associated with hard-copy textbooks such as retaining the table of contents and indexes.

The platform must include the following features to ensure that students can achieve active reading and interact fully with the subject domain:

- allow students to highlight text,
- allow students to make notes around specific sections of text,
- provide generated answers to student asked questions,
- generate questions for the student to practise with,
- produce concept summaries,
- define terms in-line of text, and
- suggest questions around concepts when they are highlighted

In order for students to easily access the potentially computationally expensive processing behind a smart textbook, the platform must be online and web-based. It must also support importing of ontologies in the OWL format to support a variety of ontology creation environments. This would hopefully ensure SMEs can create ontologies in whatever environment they are comfortable with.

REFERENCES

[1] Maha Al-Yahya, Remya George, and Auhood Alfaries. 2015. Ontologies in E-learning: review of the literature. *International Journal of Software Engineering and Its Applications* 9, 2 (2015), 67–84.

[2] David P Ausubel and Mohamed Youssef. 1963. Role of discriminability in meaningful paralleled learning. *Journal of Educational Psychology* 54, 6 (1963), 331–336.

[3] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. 1976. *Graph theory with applications*. Vol. 290. Elsevier.

[4] Charles C Bonwell and James A Eison. 1991. *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports.* ERIC Clearinghouse on Higher Education, The George Washington University, One Dupont Circle, Suite 630, Washington, DC 20036-1183, Chapter 1, 19.

[5] Vinay K Chaudhri, Britte Cheng, Adam Overtholtzer, Jeremy Roschelle, Aaron Spaulding, Peter Clark, Mark Greaves, and Dave Gunning. 2013. Inquire Biology: A textbook that answers questions. *AI Magazine* 34, 3 (2013), 55–72.

[6] Jennifer Chipps, Jane Kerr, Petra Brysiewicz, and Fiona Walters. 2015. A survey of University students' perceptions of learning management systems in a low-resource setting using a technology acceptance model. *CIN: Computers, Informatics, Nursing* 33, 2 (2015), 71–77.

[7] Gobinda Chowdhury. 2012. How digital information services can reduce greenhouse gas emissions. *Online Information Review* 36, 4 (2012), 489–506.

[8] Peter Clark, Bruce Porter, and Boeing Phantom Works. 2004. Km–the knowledge machine 2.0: Users manual. *Department of Computer Science, University of Texas at Austin* 2, 5 (2004).

[9] Alan R Dennis, Kelly O McNamara, Anastasia S Morrone, and Joshua Plaskoff. 2015. Improving learning with etextbooks. In *2015 48th Hawaii International Conference on System Sciences*. IEEE, 5253–5259.

[10] Nomusa Dlodlo and JP Tolmay. 2012. An architecture for eBook provision to South African schools. In *14th Annual Conference on World Wide Web Applications, Durban, 7-9 November 2012*. ZA-WWW, 4–15.

[11] Giselle Du Plessis and Melanie Wiese. 2014. The battle of the e-textbook: libraries' role in facilitating student acceptance and use of e-textbooks. *South African Journal of Libraries and Information Science* 80, 2 (2014), 17–26.

[12] Jérôme Euzenat, Pavel Shvaiko, et al. 2007. *Ontology matching*. Vol. 18. Springer.

[13] Bruce Fuller, Donald B Holsinger, David Baker, Rosemary Bellew, Richard Bennett, Prema Clarke, Brunhilda Forlemu, Ben H Fred-Mensah, Rosalind Michahelles, Pablo Stansbery, et al. 1993. *Secondary education in developing countries*. Education and Social Policy Department, World Bank.

[14] Benjamin Grosof, Mike Dean, and Michael Kifer. 2009. The SILK system: Scalable and expressive semantic rules. In *8th International Semantic Web Conference*. Springer, 34.

[15] Thomas R Gruber. 1993. A translation approach to portable ontology specifications. *Knowledge acquisition* 5, 2 (1993), 199–220.

[16] David Gunning, Vinay K Chaudhri, Peter E Clark, Ken Barker, Shaw-Yi Chaw, Mark Greaves, Benjamin Grosof, Alice Leung, David D McDonald, Sunil Mishra, et al. 2010. Project Halo update—progress toward digital Aristotle. *AI Magazine* 31, 3 (2010), 33–58.

[17] Harold Henke. 2002. Survey on electronic book features. In *Open eBook Forum, online*. 1–14.

[18] Hanho Jeong. 2012. A comparison of the influence of electronic books and paper books on reading comprehension, eye fatigue, and perception. *The Electronic Library* 30, 3 (2012), 390–408.

[19] Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik. 2012. ELK Reasoner: Architecture and Evaluation.. In *ORE*.

[20] Michal Laclavik, Martin Šeleng, Marek Ciglan, and Ladislav Hluchỳ. 2012. Ontea: Platform for pattern based automated semantic annotation. *Computing and informatics* 28, 4 (2012), 555–579.

[21] Monica Landoni and Forbes Gibb. 2000. The role of visual rhetoric in the design and production of electronic books: the visual book. *The electronic library* 18, 3 (2000), 190–201.

[22] Marlaine E Lockheed, Stephen C Vail, and Bruce Fuller. 1986. How textbooks affect achievement in developing countries: Evidence from Thailand. *Educational Evaluation and Policy Analysis* 8, 4 (1986), 379–392.

[23] Bradley N Miller and David L Ranum. 2012. Beyond PDF and ePub: toward an interactive textbook. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education*. ACM, 150–155.

[24] Boris Motik, Bijan Parsia, and Peter F Patel-Schneider. 2009. OWL 2 web ontology language XML serialization. *World Wide Web Consortium* (2009).

[25] Jeanne Moulton. 1994. How do teachers use textbooks and other print materials: A review of the literature. *The Improving Educational Quality Project, South Africa* (1994).

[26] Mark A Musen et al. 2015. The protégé project: a look back and a look forward. *AI matters* 1, 4 (2015), 4–12.

[27] Julius T. Nganji, Mike Brayshaw, and Brian Tompsett. 2012. Ontology-driven disability-aware e-learning personalisation with ONTODAPS. *Campus-Wide Information Systems* 30, 1 (2012), 17–34.

[28] Aannemarie Sullivan Palinscar and Ann L Brown. 1984. Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and instruction* 1, 2 (1984), 117–175.

[29] P Poyry and J Puustjarvi. 2003. CUBER: A personalised curriculum builder. In *Proceedings 3rd IEEE International Conference on Advanced Technologies*. IEEE, 326–327.

[30] Amanda J Rockinson-Szapkiw, Jennifer Courduff, Kimberly Carter, and David Bennett. 2013. Electronic versus traditional print textbooks: A comparison study on the influence of university students' learning. *Computers & Education* 63 (2013), 259–266.

[31] Bella Ross, Ekaterina Pechenkina, Carol Aeschliman, and Anne-Marie Chase. 2017. Print versus digital texts: understanding the experimental research and challenging the dichotomies. *Research in Learning Technology* 25 (2017).

[32] Suzanne Sackstein, Linda Spark, and Amy Jenkins. 2015. Are e-books effective tools for learning? Reading speed and comprehension: iPad® i vs. paper. *South African Journal of Education* 35, 4 (2015).

[33] Michele L Simpson and Sherrie L Nist. 1990. Textbook annotation: An effective and efficient study strategy for college students. *Journal of Reading* 34, 2 (1990), 122–129.

[34] Karl L Smart and Nancy Csapo. 2007. Learning by doing: Engaging students through learner-centered activities. *Business Communication Quarterly* 70, 4 (2007), 451–457.

[35] Tolga Soyata, He Ba, Wendi Heinzelman, Minseok Kwon, and Jiye Shi. 2015. Accelerating mobile-cloud computing: A survey. In *Cloud Technology: Concepts, Methodologies, Tools, and Applications*. IGI Global, 1933–1955.

[36] Irena Spasic, Sophia Ananiadou, John McNaught, and Anand Kumar. 2005. Text mining and ontologies in biomedicine: making sense of raw text. *Briefings in bioinformatics* 6, 3 (2005), 239–251.

[37] John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive science* 12, 2 (1988), 257–285.

[38] Dmitry Tsarkov and Ian Horrocks. 2006. FaCT++ description logic reasoner: System description. In *International joint conference on automated reasoning*. Springer, 292–297.

[39] Tania Tudorache, Csongor Nyulas, Natalya F Noy, and Mark A Musen. 2013. WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the web. *Semantic web* 4, 1 (2013), 89–99.

[40] Mitchell Weisberg. 2011-6. Student Attitudes and Behaviors Towards Digital Textbooks. *Publishing research quarterly* 27, 2 (2011-6), 188,196.

[41] Ruth Wilson. 2002. The look and feel of an ebook: considerations in interface design. In *Proceedings of the 2002 ACM symposium on Applied computing*. ACM, 530–534.