УДК 378.16
doi:10.18720/SPBPU/2/id24-520

Nikolai V. Scerbakov [1],
Associate Professor, Candidate of Technical Sciences;
Vadim G. Pak [2],
Associate Professor, Candidate of Physical and Mathematical Sciences

JUST-IN-TIME AND INDIVIDUALIZED COURSEWARE DELIVERY

[1] Graz University of Technology, Graz, Austria, wbtmaster2@gmail.com;
[2] Peter the Great St.Petersburg Polytechnic University, St. Petersburg, Russia, pak_vg@spbstu.ru

Abstract. Learning Management Systems (LMS) play a critical role in modern e-learning environments by providing structured repositories of courseware materials. E-Books, a key component of these materials, are typically organized as collections of HTML documents interlinked in a manner that can resemble “spaghetti” link structures. These structures, while functional, pose challenges for flexible segmentation and just-in-time delivery of educational content. In this paper, we analyze these challenges and propose new models and methods for structuring E-Books that facilitate improved segmentation and timely content delivery. We also describe an online channel for the just-in-time courseware delivery based on the Telegram bots. Our approach aims to enhance the efficacy and user experience within LMS platforms, making learning more adaptive and responsive to individual needs.

Keywords: E-Learning, Courseware, Just-in-Time delivery.

Щербаков Николай Васильевич [1],
доцент, канд. техн. наук;
Пак Вадим Геннадьевич [2],
доцент, канд. физ.-мат. наук

СВОЕВРЕМЕННАЯ И ИНДИВИДУАЛЬНАЯ ДОСТАВКА УЧЕБНЫХ ПОСОБИЙ

[1] Австрия, Грац, Технический университет города Грац, wbtmaster2@gmail.com;
[2] Россия, Санкт-Петербург, Санкт-Петербургский политехнический университет Петра Великого, pak_vg@spbstu.ru

Аннотация. Системы управления обучением (LMS) играют важнейшую роль в современных средах электронного обучения, предоставляя структурированные хранилища учебных материалов. Электронные книги, являющиеся ключевым компонентом этих материалов, обычно организованы в виде наборов HTML-документов, связанных между собой таким образом, что они могут напоминать структуры ссылок типа «спагетти». Эти структуры, хотя и функциональны, создают проблемы для гибкой сегментации и своевременной доставки образовательного контента. В этой статье мы анализируем эти проблемы и предлагаем новые модели и методы структурирования электронных книг, которые способствуют улучшению сегментации и своевременной доставке контента. Мы также описываем онлайн-канал для доставки учебных пособий «точно в срок», основанный на ботах Telegram. Наш подход направлен на повышение эффективности и удобства использования платформ LMS, что делает обучение более адаптивным и отвечающим индивидуальным потребностям.

Ключевые слова: электронное обучение, учебные программы, доставка точно в срок.

Introduction

Learning Management Systems (LMS) have revolutionized the educational landscape by providing a digital platform for delivering course materials, managing educational content, and tracking student progress [1, 2, 4, 5]. A fundamental component of LMS is the E-Book, a digital counterpart to traditional textbooks, which offers interactive and multimedia-rich content [6, 7].

Just-in-time delivery [9, 11, 12] of educational content refers to the timely provision of learning materials to students, aligned with their current needs and learning progress. This method contrasts with traditional models where students receive a bulk of information at once, often leading to cognitive overload or disengagement. Just-in-time delivery ensures that learners receive relevant information at the right moment, enhancing retention and application.

Two crucial components for effective Just-in-time delivery are:

– Online Notification Channel: a system to guarantee the notification of users about new content;

– Flexible Courseware Structure: a system that ensures the possibility to split the courseware into self-contained modules (segmentation) that can be delivered and used independently.

However, the conventional structuring of E-Books in LMS often relies on a web of hyperlinks. This “spaghetti” structure impedes the efficient segmentation of content and hampers just-in-time delivery.

This paper aims to address these issues by proposing a more structured and flexible approach to organizing E-Books within LMS. This approach was implemented in LMS called WBT-Master [5, 12].

1. Courseware Developing

WBT-Master supports the new data model (HM Data Model) for E-Books [10, 12]. The Data Model focuses on the organization and navigational structures of multimedia data, treating HTML documents as atomic units. This model provides a robust framework for flexible segmentation and structured delivery of E-Books.

In the HM Data Model, courseware consists of addressable composites called Structured Collections (S-collections). An S-collection encapsulates members together with an internal navigational topology, which is essentially a link structure expressing the relationships between members. Each member can be either a document or another S-collection, with one member designated as the head.

– S-collection: an opaque container with a head and optional label document;

– Members: can be documents or nested S-collections.

Browsing within an S-collection involves navigating its internal link topology. Users can view and explore the contents by entering and exiting collections using “Zoom-In” and “Zoom-Out” operations.

– Zoom-In: makes an S-collection the current container and its head the current member;

– Zoom-Out: restores the previous current container and member.

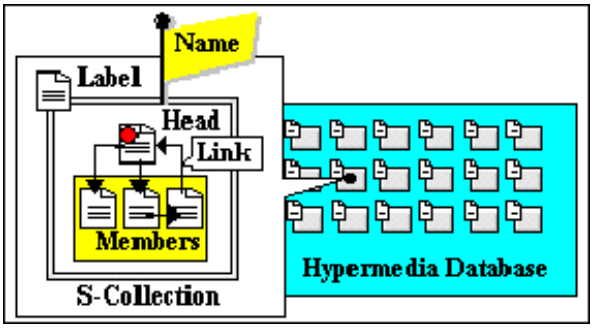

Fig. 1. Internal Structure of an S-Collection

S-collections are instances of predefined types, each with specific link topologies:

– Folder: bi-directional circular list linking each member;

– Envelope: links each member to every other member;

– Menu: connects the collection head to every other member bi-directionally;

– Freelinks: allows arbitrary linking, ensuring a path from the head to each member.

The HM Data Model offers several advantages over traditional hypermedia data models, including:

– Encapsulation of Links: links are encapsulated within S-collections, preventing external links and ensuring that each collection can be used independently without superfluous connections;

– Courseware Segmentation: building courseware as a group of independent, self-contained and uniquely addressed modules facilitate segmentation of the courseware without inflicting dependency confusion;

– Re-use of Complex Collections: collections can contain other collections, supporting the reuse of resources in different contexts and maintaining the integrity of complex hypermedia databases.

The courseware can be accessed and browsed using a normal tree-like structure where the S-Collections are visualized as folders containing plain documents or other S-Collections.

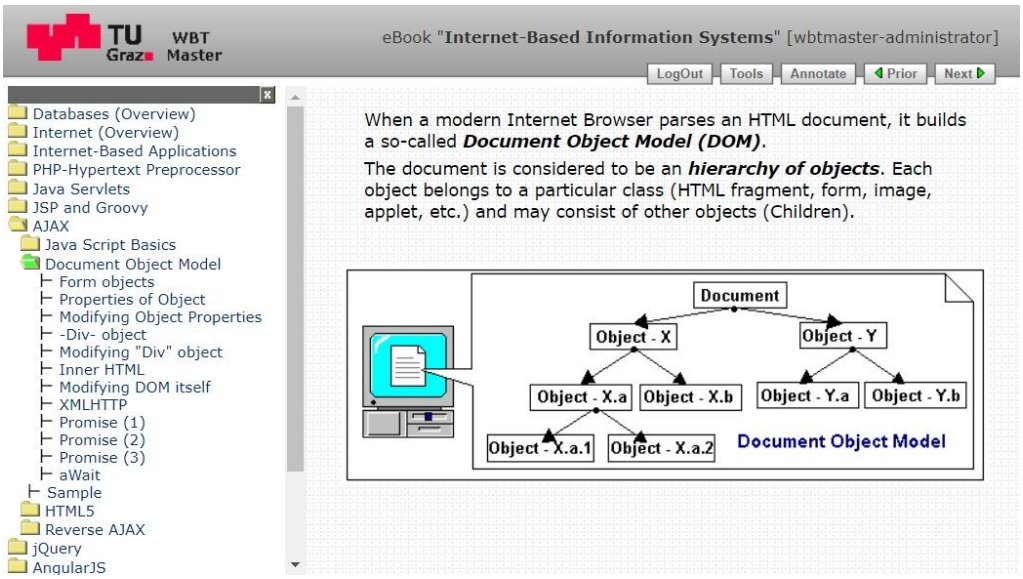

Fig. 2. Browsing courseware

2. Courseware Segmentation

The courseware can be segmented efficiently, utilizing unique URLs to access different S-Collections. Each S-Collection acts as a self-contained module that learners can easily access and browse. For example, the URL

http://scerbakov.com:8080/wbtmaster/courseViewer.htm?cif=allcoursescontent/lpi/lpi6_1_2.cif

directs to the S-Collection “lpi6_1”, which covers security aspects. Learners can use any HTTP client to reach and explore these modules.

This approach highlights the ease of segmentation and modular nature of the courseware, ensuring straightforward exchange with URLs as dissemination means.

Fig. 3. Browsing S-Collection

3. Courseware Delivery

To efficiently deliver messages containing references to specific courseware fragments that need to be learned at certain times, various technological solutions can be employed, such as: SMS messages, Push messages, Email messages, Online chats, Social systems.

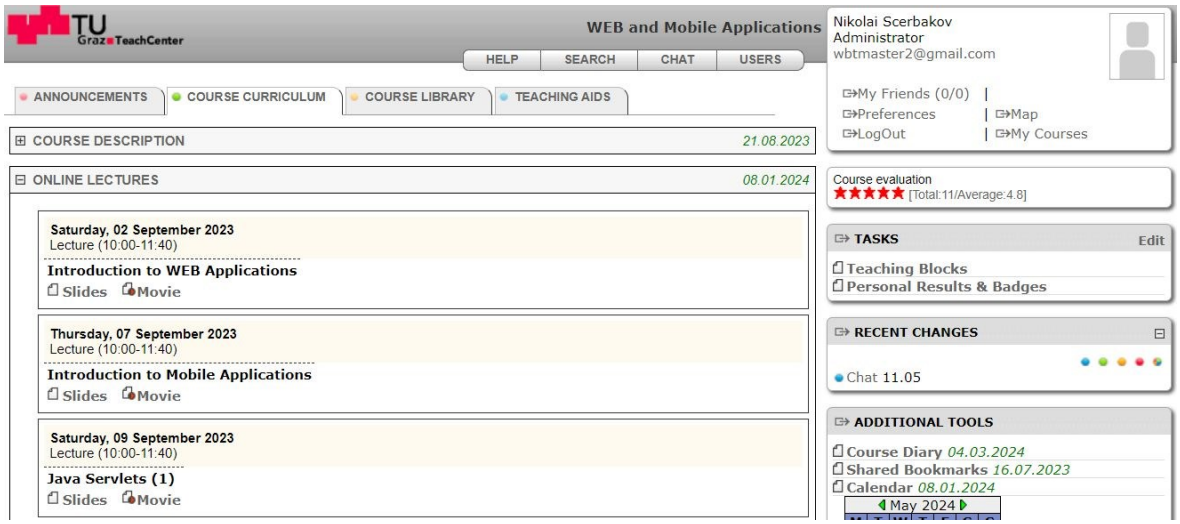

Fig. 4. Course Calendar

After thorough analysis, we have chosen to develop the Just-in-Time Delivery component using Telegram Bots [7, 10]. Telegram Bot solutions are available for various programming platforms, all following a similar structure:

Import the TelegramBot class from the library:

```
var TelegramBot = require('telegrambot');
```

Create an instance of the TelegramBot class:

```
var api = new TelegramBot("...");
```

Utilize the public interface to send messages:

```
api.sendMessage({chat_id: ..., text: ...});
```

To determine the exact timing for content delivery, the Learning Management System (LMS) supports a course calendar. This calendar lists course-related events, each linked to the relevant courseware fragment.

Fig. 5. User Notification via Telegram Bot

To facilitate the Just-in-Time delivery of courseware, the following process is implemented:

The teacher creates a special Telegram group where all students and the Telegram Bot are subscribed as members.

The Telegram Bot is given access to the course calendar, which lists course-related events and their corresponding courseware fragments.

The bot operates like a cronjob, periodically checking the course calendar.

On the stipulated date and time, the bot retrieves the relevant calendar entry.

The bot publishes a message to the Telegram group with a reference to the recommended courseware fragment.

All students are notified on the new message in the group and Students can access the courseware fragment with a single click.

4. Individualized Delivery

Obviously, the Just-in-Time Delivery process can be extended to deliver individualized content to individual learners. In this case,

– Personal Telegram chats with each student are used. The telegram bot is subscribed for each chat.

– The Telegram Bot is given access to the list of events carefully crafted for each individual student. We call this list individual calendar.

– The bot operates like before, periodically checking the individual calendars.

On the stipulated date and time, the bot retrieves the relevant calendar entry.

– The bot publishes a message to the individual Telegram chat with a reference to the recommended courseware fragment.

– The student is notified on the new message in the chat and can access the courseware fragment with a single click.

Thus, the main component of the individualized content delivery is a personal calendar for each student. Individual training calendars can be automatically generated on the base of personalized learning preferences.

To generate the personal calendar, each module in the course is tagged with specific keywords that encapsulate its core content and skills. These keywords are collated into a comprehensive list representing the entire course's content. Students review the list of keywords and select a subset that aligns with their learning preferences and goals. This subset reflects the student's areas of interest or topics they need to focus on. A special metric (e.g., cosine similarity, Jaccard index) [3] is used to calculate the relevance of each module based on the

selected keywords. Modules with the highest relevance scores are prioritized and delivered to the students.

An alternative method can be implemented as follows. At the beginning of the course, students fill out a detailed questionnaire designed to capture their learning preferences, needs, and goals. Questions could cover various aspects such as preferred learning style, prior knowledge, specific interests, and learning pace.

As modules are delivered using, for example, the usual course calendar, students provide feedback on the relevance and usefulness of each module. This feedback helps in understanding how well the content meets their educational needs. Pairs of questionnaire responses and the corresponding sets of highly relevant modules as identified by the students' feedback are collected to accumulate a substantial dataset of these pairs.

The accumulated dataset is used to train a neural network that can predict the most relevant courseware modules based on questionnaire responses. The model must be continuously refined with new data to improve accuracy and relevance.

Conclusion

The proposed Just-in-Time Delivery system offers a compelling alternative to traditional free-hand browsing of courseware by streamlining the process of locating necessary content. This approach aims to enhance efficiency and adaptability in Learning Management Systems (LMS) by eliminating the time-consuming task of searching for relevant fragments, thereby making the system more maintainable and scalable.

However, the effectiveness of the system depends on the method used to select courseware fragments, which can be challenging and context dependent. Below are the primary methods considered, along with their advantages and limitations:

Course Calendar-Based Solution Advantages:

– Ease of Implementation: simple to set up and manage;

– Predictability: provides a structured timeline for content delivery.

Limitations:

– Lack of Flexibility: may not accommodate individual learning paces or unexpected changes in the course schedule;

– Dependence on Quality: the system's effectiveness relies heavily on the accuracy and comprehensiveness of the predefined course calendar; any discrepancies in the calendar can lead to inefficiencies or gaps in content delivery.

Keyword-Based Calendar Generation Advantages:

– Relevance: offers more targeted results based on specific keywords, which can improve the alignment of content with user needs;

– Customization: allows for some degree of customization through manual keyword assignment.

Limitations:

– Manual Effort: requires ongoing effort to assign and manage keywords;

– Distance Metric Optimization: the effectiveness of this method depends on the precise selection and continuous optimization of the distance metric used to match user queries with courseware units; regular fine-tuning is necessary to maintain and improve performance.

Neural Networks Approach Advantages:

– User-Friendliness: provides a highly intuitive and contextually accurate content delivery experience;

– Adaptability: can learn and adapt over time to better meet user needs, offering personalized content recommendations.

Limitations:

– Complex Implementation: requires significant expertise in machine learning for initial setup and ongoing training of the neural network;

– Resource Intensive: demands considerable computational resources and continuous maintenance to ensure optimal performance.

In conclusion, the selection of the appropriate method for delivering courseware fragments through the Just-in-Time Delivery system depends on the specific requirements and constraints of the educational environment. A course calendar-based solution may be sufficient for straightforward, time-bound courses, while a keyword-based calendar generation offers more flexibility and relevance for dynamic content needs. For the most sophisticated and personalized experience, a neural networks approach is ideal, provided that the necessary resources and expertise are available to support its implementation and maintenance. Balancing these factors will be key to maximizing the efficiency and effectiveness of the Just-in-Time Delivery system in enhancing the LMS.

References

1. Dietinger T., Maurer H. GENTLE – General Network Training and Learning Environment // Proc. of ED-MEDIA98 / ED-TELECOM 98, Freiburg, 1998. – Pp. 274–280.

2. Ebner M., Scerbakov N., Maurer H. New features for eLearning in higher education for civil engineering // Journal of Universal Science and Technology of Learning. – 2006. – Vol. 1, No. 1. – Pp. 93–106.

3. Myers E. W. An O(ND) difference algorithm and its variations // Algorithmica. – 1986. – Vol. 1. – Pp. 251–266.

4. Scerbakov A., Ebner M., Scerbakov N. Using cloud services in a modern learning management system // Journal of Computing and Information Technology. – 2015. – Vol. 23(1). – Pp. 75–86.

5. Scerbakov N. TU Graz Teach-Center [Electronic resource]. – URL: http://coronet.iicm.tugraz.at/wbtmaster/welcome.html (access date: 01.05.2024).

6. Scerbakov A., Kappe F., Scerbakov N., Pak V. Intelligent curriculum assistant // In: Proceedings of 11th International Conference on Education and New Learning Technologies, Palma, Mallorca, Spain, July 1–3, 2019. – Pp. 7964–7967. – ISBN: 978-84-09-12031-4, ISSN: 2340-1117.

7. Scerbakov N., Schukin A., Rezedinova E. Architecture of modern e-learning management system // 2023 4th International Conference on Communications, Information, Electronic and Energy Systems (CIEES), Plovdiv, Bulgaria, 2023. – Pp. 1–5. – DOI:10.1109/CIEES58940.2023.10378809.

8. Telegram Bot API [Electronic resource]. – URL: https://core.telegram.org/bots/api (access date: 01.05.2024).

9. Davey K. Just in time training [Electronic resource]. – URL: https://www.docebo.com/glossary/just-in-time-training (access date: 01.05.2024).

10. Kappe F., Scerbakov N. Object-oriented architecture of a modern learning management system // In: Proceedings of 9th International Conference on Education and New Learning Technologies, Barcelona, Spain, July 3–5, 2017. – Pp. 4910–4916.

11. Kappe F., Scerbakov N. Quantitative evaluation of university lecturing // In: Proceedings of 20th International Conference on Interactive Collaborative Learning, Budapest, Hungary, September 27–29, 2017. – Pp. 335–340.

12. Scerbakov N., Kappe F. WBT-Master – Modern learning management system // In: 24th International Conference, ICIST 2018, Vilnius, Lithuania, October 4–6, 2018. – Springer "Communications in Computer and Information Science". – 2018. –Vol. 920. – Pp. 461–476.