Upcycling *UnivIS*: Discovering Study Planning 2.0

Michaela Ochs[1], Tobias Hirmer[1], and Andreas Henrich[1]

Abstract: Technical innovation in university settings often progresses slowly, with legacy systems frequently remaining in use for years. This can lead to outdated technologies that no longer meet user needs. In this demonstration, we explore the revival of such legacy systems through the example of *UnivIS*, a course administration system used at the University of Bamberg for over two decades. We introduce *Baula*, a digital study assistant that builds on *UnivIS* data and functionality, integrating them into a modern, user-friendly interface. By upcycling existing infrastructure, *Baula* addresses significant student concerns regarding usability and functionality. This demonstration illustrates how such legacy systems can be modernized to better align with contemporary user requirements, showcasing a model for innovation within a traditionally slow-moving environment.

Keywords: Legacy Systems, Digital Transformation, Higher Education, Study Planning

1 Introduction

In the rapidly evolving landscape of technology, many universities continue to rely on legacy systems, which seems to be a common issue in highly administrative environments [Ir23]. These outdated technologies can persist for years, creating challenges in adapting to new educational demands and hindering the overall user experience. One such example is *UnivIS* [eG24], a course management and information system that has been in use at the University of Bamberg for more than two decades. Over this long period, technological advancements and user requirements have evolved significantly. A recent university-wide survey revealed strong student dissatisfaction with the system [HOH23]. Key criticisms included searching for courses, editing, updating, and saving the semester plan as well as the system's overall lack of intuitive design, with one student even describing the system as "user-hostile". In this demonstration, we present *Baula*, a digital study assistant (DSA) that, among other features, integrates *UnivIS* data and functionality into a more modern and intuitive semester planning interface, and is therefore more tailored to students' present-day needs.

The challenge of modernizing legacy systems is a common technical barrier at universities [GP22]. Our case demonstrates how, rather than waiting for slow replacements, these systems can be "upcycled" – utilizing existing (parts of) legacy systems to create something new [Di24]. Accordingly, we will first provide an introduction into *Baula* to give an impression of the system. Second, we will explore the technical background before focusing on the semester planning functionality in the demonstration. Specifically, we will showcase how

[1] University of Bamberg, Chair of Media Informatics, An der Weberei 5, 96047 Bamberg, Germany, {michaela.ochs, tobias.hirmer, andreas.henrich}@uni-bamberg.de, https://orcid.org/0000-0002-3850-8585, https://orcid.org/0000-0002-5281-0342, https://orcid.org/0000-0002-5074-3254

doi:10.18420/BTW2025-52

Baula addresses the concerns raised by students and how the system can be extended. Last, we address limitations and conclude the paper. While *UnivIS* is also used by instructors who manage courses, we will focus on the student perspective.[2]

2 Digital Study Assistant *Baula*

Baula is a digital study assistant with a strong focus on the process of study planning, aiming to support students in deciding which module or course to choose when [OHH23]. Therefore, it allows students to create their semester plan (= timetable for current semester) as well as their long-term study plan (= list of semesters) containing modules they want to attend the following semesters. Additional features are a dashboard and an interactive module catalog. Due to the availability of machine-readable data, *Baula* is currently restricted to study programs of the faculties "information systems and applied computer sciences" and "social sciences, economics and business administration". *Baula*'s functionality[3] can be accessed via a demo account that is provided for participants in the demonstration session.

3 Data Acquisition and Technical Implementation

Due to the focus of this demonstration, the description of the technical backend of *Baula* is limited to the data acquisition from *UnivIS* as well as the data needed for the semester plan (Fig. 1). First, all courses of the current semester are crawled daily via the PRG-API of *UnivIS*. This open accessible API delivers an XML-file containing all courses of a requested semester as well as all linked data such as persons or rooms.[4] This data is then transformed into JSON using the npm-package Camaro [Ca24].[5] Afterwards, this JSON is used in our transformation service to create new courses, update existing ones and delete courses, that have been deleted in *UnivIS* since the last crawl. Therefore, existing courses are queried from the database and new courses are transformed into the correct data format. As this may create computational overhead, we crawl the system in off-peak times. To further simplify the database queries, we are using the ORM Prisma [Pr24]. Since *Baula* should run as independently of *UnivIS* as possible, we store the transformed courses in our database, making *UnivIS* necessary only for crawling.

To store our structural data (modules and courses), which is stable and mainly accessed READ-only, we chose MariaDB, a suitable relational database for our prototyping purposes. Since we also needed a solution for storing dynamic and evolving user data, which requires

[2] Please note that this article is not a criticism of *UnivIS* or its developers. The fact that it has been in use for so long and is still an important component of the IT landscape of many universities shows its quality.

[3] https://baula.minf.uni-bamberg.de/welcome

[4] E.g. the following link shows all lectures of the winter term 2024/25: https://univis.uni-bamberg.de/prg?search=lectures&show=xml&sem=2024w (last accessed 2024-10-24)

[5] Camaro enables self-defined templates using xPath to select specific parts of the XML. For our template we mainly rely on the existing *UnivIS* data scheme, with only small adjustments in combining some attributes.

flexible schemata, we selected MongoDB, aligning with the MEAN stack of *Baula*. We intentionally separated stable structural data from user data, which has to be defined dynamically in an ongoing process of evolutionary prototyping. Furthermore, we wanted to make the user data less dependent on changes in the structural data.[6] This separation will have to be re-evaluated when the system is in a more stable state.

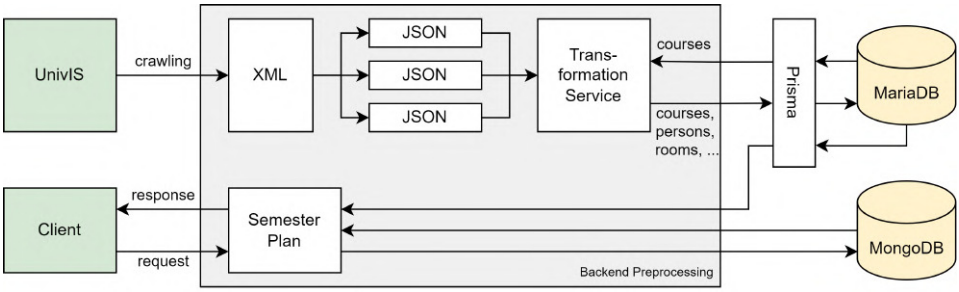

Fig. 1: Data Flow and Transformation from *UnivIS* to the Client

4 Demonstration

Guided by the concerns raised by students in a survey on the challenges and perspectives of a DSA [HOH23], we derived four major questions based on which the *UnivIS* functionality was extended in our "upcycling" process as outlined in this section: (1) How can we enable a refined, effective, and intuitive search for courses? (2) How can we create a flexible solution for editing and adjusting the semester plan? (3) What options can we create for persisting the semester plan? (4) How can we make semester planning helpful to the individual student?

Searching Lectures and Courses To enhance the search capabilities provided by *UnivIS*, a central objective was to refine the existing search functionality by addressing key limitations that impacted the user experience. Specifically, this refers to the downsides of the exact strict-matching approach, i.e. sensitivity to minor user input errors such as trailing whitespaces or misspellings. These obstacles affected the usability and effectiveness of the search process in *UnivIS*. In order to circumvent such limitations, we implemented a simple fuzzy search mechanism in *Baula* using Fuse.js [Fu24b]. Fuse.js uses approximate string matching and is an initial attempt to improve user experience and mitigate user frustration when searching in a simple manner. Moreover, we integrated a live search feature which provides immediate feedback as it adapts to the user's search input dynamically, increasing the responsiveness when searching. We thus decided to perform the search on the client side and query all courses of the current semester from the MariaDB initially, whereas optional filter settings (cf. Fig. 2 – A-D) are applied on the server side when the data is queried from the database.

[6] E.g. if an *UnivIS* course is deleted, referential integrity could lead either to inconsistency between our database and *UnivIS* or to a deleted course within the user's semester plan without any notice.

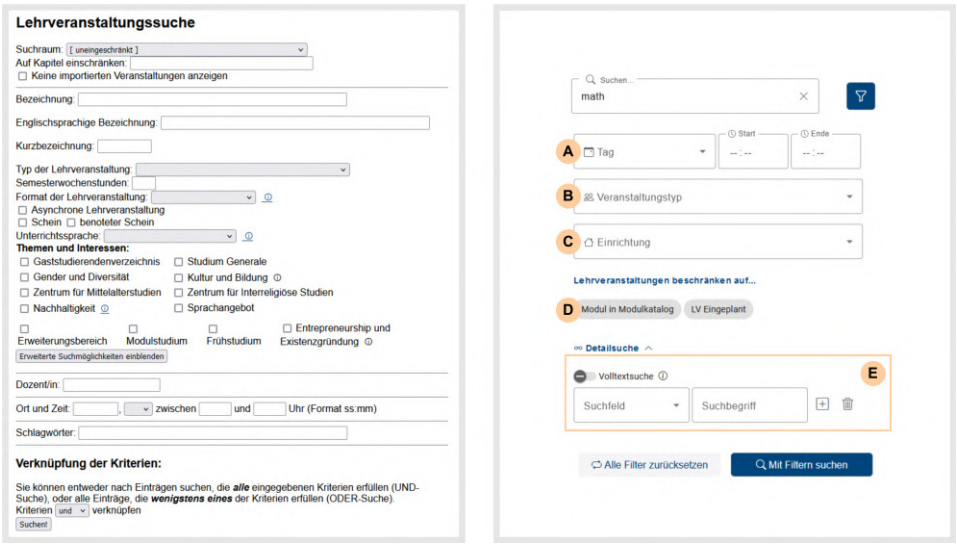

Fig. 2: Comparison of *UnivIS* Advanced Search (left) and *Baula* Advanced Search (right)

This solution works for our current needs[7], but we will have to rethink this distribution between client and server with changing requirements.

Furthermore, we wanted to enhance the search options provided by *UnivIS*. *UnivIS* provides a simple search input and a detailed advanced search accessible via the main menu, which includes a multitude of options (Fig. 2 – left). As default search, we also provide a simple input field, but make the advanced options more easily accessible via a filter button next to the input. In these filters (Fig. 2 – right), we only include the most important options belonging to a course in prominent position, i.e. time and day of a course (Fig. 2 – A), type of course (Fig. 2 – B) as well as institution (Fig. 2 – C). We added two selected options (Fig. 2 – D) below which can be used to limit the results further such as only displaying courses students have already planned. We further included more advanced options (Fig. 2 – E) like a full-text and detailed search for more advanced users, but they are collapsed by default, so the major options are clearly visible and the options remain comprehensive.

Editing & Adjusting Planned Courses Another major critical aspect of semester planning with *UnivIS* is the inconvenience of editing and adjusting planned courses in the semester plan by navigating back and forth via the menu. In the first place, we address this issue by allowing for searching and managing courses and viewing the semester plan side by side via a course searching expandable and collapsible side navigation (cf. Fig. 3).

[7] Currently, with a moderate number of users on the system the initial search including querying the necessary data (approx. 8MB without filtering) takes about 1-2 seconds depending on the internet connection and system load.

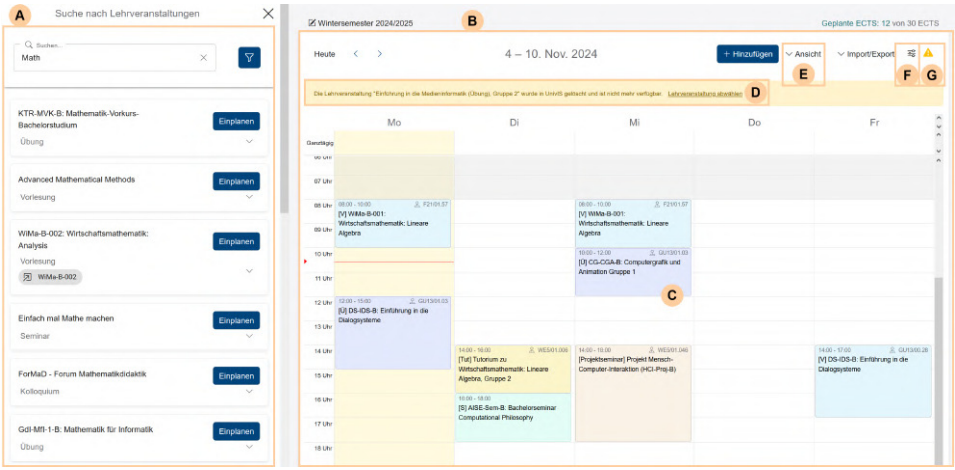

Fig. 3: Searching for Courses (left) and Semester Plan (right) Side by Side

Accordingly, courses can be searched and (de-)selected on the left-hand side (Fig. 3 – A) whereas the semester plan can be managed on the right-hand side (Fig. 3 – B). Course details can be viewed by clicking on a semester plan event (Fig. 3 – C). The semester plan is currently based on the FullCalendar standard version [Fu24a], which we found to provide the relevant functionality. The semester plan displays all relevant courses that a student has selected for each individual week of the semester, considering the date, schedule, and time the course manager has defined in *UnivIS*. Since the course data is synchronized daily, we can react to updates like time changes or course deletions so that if a course is deleted in UnivIS, it is also deleted in our database. In contrast to UnivIS, *Baula* informs the user when a planned course was deleted (Fig. 3 – D). Also, FullCalendar comes with the functionality of further semester plan customization, which allows students to choose a display option (week, day, month or list view) (Fig. 3 – E) or by making other adjustments (Fig. 3 – F) such as hiding weekends according to personal preference.

Persisting the Semester Plan Another central aspect for study planning is the availability of options of persisting the semester plan. In *UnivIS*, a collection of selected courses can be stored and loaded, but only within the current session. In *Baula*, selected courses are saved in the MongoDB to store the plan persistently (cf. Fig. 1 – bottom part) and retrieve it when needed. To do this, we store the planned courses together with a unique user ID, which is provided with the login via the university AAI. Besides PDF export, we provide the import or export of plans in JSON format, which not only allows students to store a local copy, but also to share their plan with other students, which was not possible before.

(Individual) Planning Support What is also not covered by the previous system is the support of more individual planning and error prevention. A common planning error that can occur with simple course selection via *UnivIS* is that students do not know or forget that they need to select all courses that belong to a module (abstract thematic unit), i.e. not only the lecture item, but also the matching tutorial. We therefore use an unobtrusive warning icon (Fig. 3 – G), which students can click on to inspect which modules this applies to.

Our focus for further development is on extending the current functionality towards more individual needs such as integrating personal events into the semester plan or supporting students in their study planning by recommending relevant courses [Oc23]. A further idea that would be helpful for students is integrating a notification mechanism if course details were changed within *UnivIS*. Finally, we recognize the importance of linking this semester plan with *Baula's* long-term planning functionality as central to further extension of the system, so students can create a semester plan directly from their long-term plan.

5 Conclusion

As we have demonstrated, developing a more modern, user-friendly system based on a legacy system is basically feasible. Issues previously raised by students – such as the search functionality, difficulties in editing or in persisting the semester plan – can be effectively addressed through a system like *Baula*. However, certain limitations remain.

A key concern is that *Baula*, drawing on *UnivIS* data, relies on user-generated data (information on courses provided by the lecturers) that often lack quality and consistency with other systems. For instance, features like course recommendations and warning alerts require clear knowledge of how courses are affiliated with modules. While courses are managed in *UnivIS*, modules and their course references are managed in a separated system with no automated system-side mapping, which leads to inconsistent references which must be manually integrated into *Baula*. These are organizational rather than technical obstacles, which are challenging to address. Additionally, while we conducted an initial evaluation of the semester planning functionality [Ba24], further assessments are necessary to refine the system and explore future adjustments for personalization. We have noted positive initial adoption among students and continue to receive valuable feedback. These user insights are crucial in shaping such a tool and should be considered both in the upcycling process and more generally in the gradual replacement of legacy systems.

Moreover, this article shows how upcycling legacy systems can enable institutions such as universities to innovate without complete, immediate replacements of outdated technologies. It emphasizes that modern solutions can arise from existing infrastructures, particularly with ongoing user involvement, which enhances the relevance and usability of these systems. However, while developing new solutions is crucial, it cannot replace the need to phase out legacy systems. Therefore, both efforts – upcycling existing technologies and strategically replacing outdated systems – must occur simultaneously as complementary initiatives.

Bibliography

[Ba24] Bartel, Lena; Ochs, Michaela; Hirmer, Tobias; Henrich, Andreas: Design Principles for a Study Planning Assistant in Higher Education. In: Proceedings of the 2024 Conference on Human Information Interaction and Retrieval. CHIIR '24, Association for Computing Machinery, New York, NY, USA, pp. 243–253, 2024.

[Ca24] Camaro: Camaro - npm, 2024. https://www.npmjs.com/package/camaro [Accessed: 2024-10-24].

[Di24] Dictionary, Cambridge: Upcycling, October 2024. https://dictionary.cambridge.org/dictionary/english/upcycling [Accessed: 2024-10-18].

[eG24] eG, Config: UnivIS, 2024. http://www.univis.de/ [Accessed: 2024-10-18].

[Fu24a] FullCalendar: FullCalendar - JavaScript Event Calendar, 2024. https://fullcalendar.io/ [Accessed: 2024-10-24].

[Fu24b] Fuse.js: Fuse.js, 2024. https://www.fusejs.io/ [Accessed: 2024-10-24].

[GP22] Gkrimpizi, Thomais; Peristeras, Vassilios: Barriers to Digital Transformation in Higher Education Institutions. In: Proceedings of the 15th International Conference on Theory and Practice of Electronic Governance. ICEGOV '22, Association for Computing Machinery, New York, NY, USA, pp. 154–160, 2022.

[HOH23] Hirmer, Tobias; Ochs, Michaela; Henrich, Andreas: Studierende und die Studienplanung: Untersuchung von Herausforderungen und Entwicklungsperspektiven eines digitalen Studienplanungsassistenten. In: 21. Fachtagung Bildungstechnologien (DELFI), pp. 291–292. Gesellschaft für Informatik e.V., Bonn, 2023.

[Ir23] Irani, Zahir; Abril, Raul M.; Weerakkody, Vishanth; Omar, Amizan; Sivarajah, Uthayasankar: The impact of legacy systems on digital transformation in European public administration: Lesson learned from a multi case analysis. Government Information Quarterly, 40(1):101784, 2023.

[Oc23] Ochs, Michaela; Hirmer, Tobias; Past, Katherina; Henrich, Andreas: Design-Focused Development of a Course Recommender System for Digital Study Planning. In (Abelló, Alberto; Vassiliadis, Panos; Romero, Oscar; Wrembel, Robert; Bugiotti, Francesca; Gamper, Johann; Vargas Solar, Genoveva; Zumpano, Ester, eds): New Trends in Database and Information Systems. Communications in Computer and Information Science, Springer Nature Switzerland, Cham, pp. 575–582, 2023.

[OHH23] Ochs, Michaela; Hirmer, Tobias; Henrich, Andreas: Concept and Possible Impacts of a Study Planning Assistant in Higher Education. In: 2023 International Symposium on Educational Technology (ISET). pp. 161–165, 2023.

[Pr24] Prisma: Prisma | Simplify Working and Interacting with Databases, 2024. https://www.prisma.io/ [Accessed: 2024-10-24].