A Web-Based Recommendation System for Higher Education: SIDDATA

History, Architecture, and Future of a Digital Data-Driven Study Assistant

https://doi.org/10.3991/ijet.v17i22.31887

Felix Weber*(✉), Johannes Schrumpf*, Niklas Dettmer, Tobias Thelen
Institute of Cognitive Science, University of Osnabrück, Germany
fweber@uni-osnabrueck.de

Abstract—The SIDDATA data-driven digital study assistant offers students various services that help them identify and achieve their personal study goals. The software's features and infrastructure have evolved to become a universal platform for interactive self-regulated learning and digital study planning throughout three annual software development cycle iterations. The software is fully integrated into an existing learning management system (Stud.IP) and has been tested by more than 3000 students from three German universities during the last three years. This paper presents the SIDDATA software architecture, design philosophy, and modular, feature-centered application logic. Developed during a third-party-funded research project with limited temporal scope, the web-based software is publicly available under an MIT license. We conclude with application opportunities for researchers, developers, educators, and higher education institutions.

Keywords—digital student assistant, e-learning, artificial intelligence, innovative learning technologies

1 Introduction

German higher education at the beginning of the 21st century is characterized by its heterogeneity in terms of learning environment embeddedness: While university courses pre-covid took place in-person and followed classical teaching paradigms such as lectures, seminars, and practice sessions, a vast amount of self-organization and learning content engagement is taking place via online platforms such as on-campus learning management systems (LMS) or off-site learning platforms that provide MOOCs, OERs or other material. Simultaneously, German higher education offers the opportunity for students to follow secondary personal and educational goals, such as

* Authors contributed equally.

spending a semester abroad or connecting with other students to form friendships or networks for their professional life.

The number and variety of AI-based digital learning assistants constantly increase. The following list of examples introduces some systems representative of the field:

- The tech4comp project at the University of Dresden is developing an AI-based adaptive mentoring system, which is intended to augment human tutors. In contrast to our learner-centered approach, tech4comp provides teachers and mentors with information about individual learning progress.
- Commercial MOOC platforms provide more or less intelligent search functions to find resources matching the learners' needs.
- For linguistic tasks, there are AI-based assistants available already. For instance, Grammarly, an AI-based commercial writing assistant, can correct spelling and grammar and provide feedback about word variety, sentence structure, plagiarism, and the mood of texts.
- Institutions in higher education, such as, for instance, the University Bamberg, or the Insitute of Cognitive Science at Osnabrück University, provide study planning assistants that allow scheduling courses in semesters according to curricular study regulations.

The field of AI and data-driven learning assistants is relatively young, and systems using data and AI for educational purposes occur under variable terms, making it challenging to oversee the whole field.

Project SIDDATA aims to develop a study assistant software (DSA) that galvanizes and supports students in identifying, organizing, and achieving their personal educational goals [3,6]. This is achieved through a rich selection of recommendations and support features that are presented to users based on their individual needs and interests. Over three years, the SIDDATA DSA has evolved to become a universal platform for hosting features to achieve these goals while simultaneously providing AI researchers the basis for developing data-driven personalized recommendation algorithms.

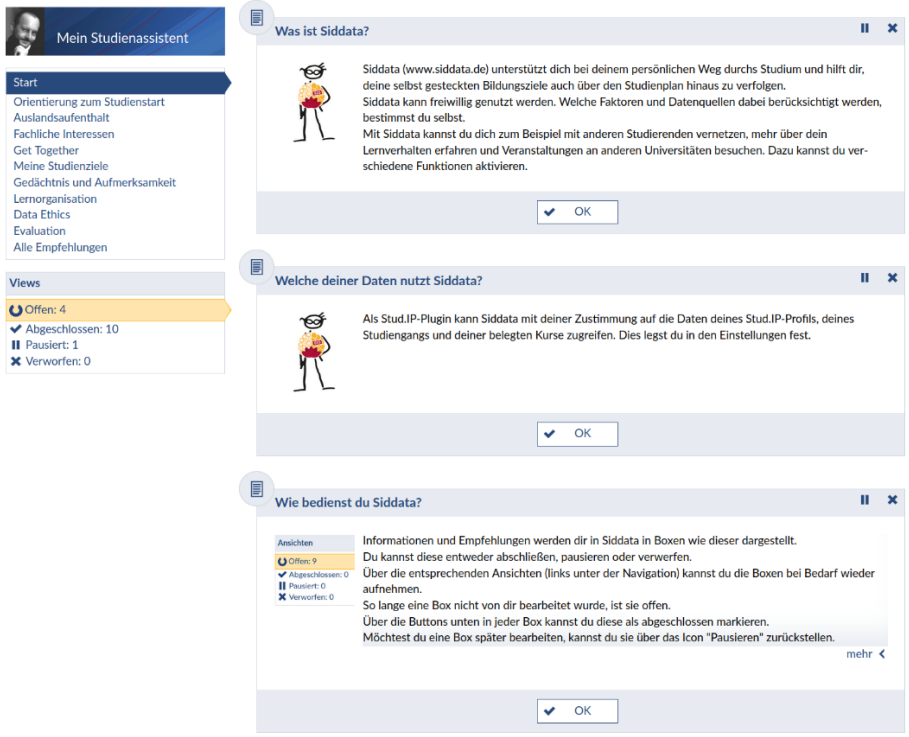

Fig. 1. The graphical user interface (GUI) of the Stud.IP plugin with available recommenders in the navigation bar at the left and the news-feed-like recommendations at the right

In this paper, we present the final SIDDATA DSA software architecture that resulted from three annual development cycles during the project lifetime, extensive testing and refinement in field experiments with more than 3000 students, its core design philosophies and features, and existing data processing pipelines. We provide an overview of existing out-of-the-box study assistant features, outlooks on possible future feature expansions, and reflect on lessons learned from the development process.

2 The data-driven study assistant

Among the challenges in the development of digital study assistants are: realizing technical accessibility with low costs and burdens for users, gathering and utilizing data from heterogeneous sources in concordance with data regulations, transparency, protecting the data from IT security threats, and applying AI algorithms to the data. The following sections show how the SIDDATA study assistant solves these technical challenges. We introduce the SIDDATA software architecture, a set of central database models, and subsequently give an overview of major features.

2.1 System architecture

The SIDDATA DSA is a web application and consists of a backend server based on the Django web framework [1] and a frontend implemented as a PHP-based plugin for the Learning Management System Stud.IP [4]. Table 1 informs about both components while Figure 2 illustrates the software architecture.

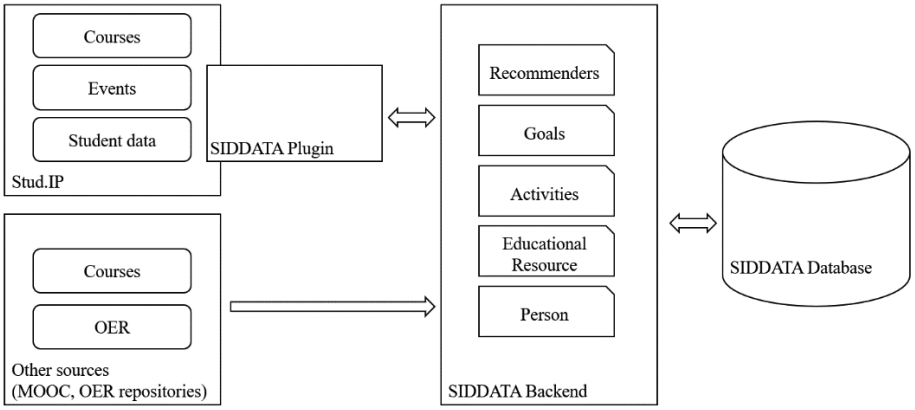

Fig. 2. Software architecture

The Stud.IP Plugin provides a graphical user interface (see Figure 1). It has access to learner-related data, such as the study program, and public data, such as courses available at the university. Users can decide which information to share with SIDDATA and other users in a data settings view. The data transfer to the backend server happens via RESTful API, and data is stored in a pseudonymized format in the backend's PostgreSQL database [5]. A cronjob in the Stud.IP plugin collects public data at night when the system usage is low and sends it to the backend database. Another cronjob in the backend server collects educational resources data from public repositories for Open Educational Ressources (OER) and Massive Open Online Courses (MOOCs).

Table 1. Software components

	Backend Server	**Stud.IP Plugin**
GitHub URL	https://github.com/virtUOS/siddata_backend	https://github.com/virtUOS/siddata_studip_plugin
License	MIT License	MIT License
Technologies	Django, PostgreSQL, Python, TensorFlow	PHP, Html, javascript

2.2 Database models

Django's Object-Relational Mapping (ORM) mechanism allows developers to implement Python classes and instantiate objects, while in the background, SQL relations represent classes and database entries represent objects. The data types listed in

Table 2 play a central role in the backend's logic and are characterized in the following sections.

Table 2. Central ORM classes

Model class	Function
SiddataUser	learners using the DSA, possibly information about the study program
Recommender	functional modules, encapsulating separate features
Goal	educational goals entail corresponding activities and recommendations
Activity	single recommendations / user interaction boxes
EducationalResource	courses, events, books, OERs, MOOCs

System users are modeled as *SiddataUsers* with possible information about their study program. We systematically avoid information that could lead to students' identification, such as names or student IDs. Instead, the frontend uses hashed student ids with a secret salt to calculate ids on which further data processing is based.

The backend supports the efficient prototypical implementation of assistive functionalities, encapsulated in *Recommender* class objects, which appear in the GUIs navigation bar and are represented as a relation in the DB. New recommenders are added as a single file containing a class definition inheriting from a recommender base class.

Personal and educational goals play a central role in the theoretical foundation of the SIDDATA DSA. *Goals* are hierarchically nested in recommender objects, and all recommendations, represented as *Activity* type objects generated by the DSA, are related to exactly one goal. Activities type objects represent recommendations or ask for user input and occur in variations such as todo-items, questions, courses, resources, events, or persons. *EducationalResource* objects represent various learning materials or learning occasions and entail metadata according to the Dublin Core metadata standard and partially to the Learning Object Metadata (LOM) standard. As a result, *EducationalResource* objects can represent all kinds of educational resources in a standardized data format.

2.3 Features

The SIDDATA DSA provides the following features for learners:

- easy access through integration into local LMS
- transparent data usage and fine-grained data-sharing settings
- data protection by design by hashed pseudonymization of user-related data in the backend DB
- intuitive usability, realized by a newsfeed-like general user-interaction
- reminders and nudging functions with push notifications by e-mails
- a growing set of recommendation functions, for instance, personalized recommendations of learning materials [2], support and nudging for a semester abroad, and a social matching recommender for academic contacts

The following aspects are relevant for developers and researchers to reuse the code:

- open-source publication under MIT license
- documentation in the code and automatically generated documentation
- cronjobs allow for database functions, such as OER and MOOC synchronization with online repositories
- cronjobs allow to train and update machine learning models
- admin views allow to inspect data and generate reports and statistics
- alternative frontends can use the RESTful API of the backend
- low effort for building new recommenders in one Python file

3 Discussion & reflection

Aside from general software tools such as LMS, digital solutions for higher education have primarily resided in the context of chatbot-driven interactive systems for question-answering [7], intelligent tutoring systems [8,9], or learning analytics methods [10]. The DSA proposed in this work differentiates itself from these established tools in two major categories: On the one hand, the SIDDATA DSA aims to support students throughout their entire studies. This stands in contrast to intelligent tutoring systems and, to some extent, learning analytics, which usually only operate within the scope of one specific course or teaching unit. In parallel, the DSA is non-domain specific in that it covers various aspects of student interests in the form of individual recommender modules. A chatbot-driven question answering system thus may be integrated into the DSA as a new recommender feature. Therefore, the DSA proposed in this work is best characterized as a universal platform for supporting students over a time frame beyond a single semester by providing an LMS-integrated hub for multiple features that differ in scope and interaction paradigm.

It follows a brief reflection on the SIDDATA DSA software design and secondary factors we have encountered during development. With its reliance on established web-development frameworks and database frameworks, SIDDATA ensures a high level of software readiness in the future. During SIDDATA development, the three participating universities fielded the Stud.IP LMS in different versions, making a successful deployment of the DSA at multiple universities challenging at first glance. However, with its tandem design, the independence between the LMS plugin and SIDDATA backend enabled the system's successful deployment under different LMS modalities. These factors make the SIDDATA DSA highly flexible in terms of LMS compatibility and, given further development of plugin software, may enable SIDDATA to interface with other LMS software in the future.

Another advantage of the SIDDATA software architecture is the modularity of recommendation features: Because recommendation features are self-contained and individually controllable, developers can prototype them rapidly. New features can be tested in local skeleton systems to be integrated into the live system later. This allowed for easy onboarding of new developers during the project's run-time.

One conceptual challenge we encountered during the development of the SIDDATA DSA was to agree upon an exact definition of the term "Digital Study Assistant". Existing literature on digital assistant systems [11,12] focuses on user interfaces or

conversation agent systems. However, these aspects do not paint a complete picture of digital assistants as a tool integrated into the normal study process and neglect necessary groundwork, such as processing raw data from LMS systems into a form usable for generating recommendations. It was possible to create a system beyond classical digital assistants' scope to emerge as a universal platform through agile development. We conclude that agile development should be retained as a development philosophy for the future development of digital study assistant systems as it allows for technological groundwork and user needs to be integrated gradually.

The difficulty of balancing personal data protection and the usage of this data for scientific purposes has been covered in recent literature, such as [13]. Further, data protection policies and their implication for rapidly evolving study assistant systems must be considered early in development. However, in the case of DSAs, residing at the intersection between software services and tools for scientific research in digital education, the policies may not be as clear-cut as in classical cases, such as companies using person-related data. To avoid data protection concerns, the current version of the SIDDATA DSA implements an extensive consent form and user control over which data they share with the system.

4 Future directions

Extending the features and a growing body of educational data can lead to synergetic effects for all parties involved. The current software architecture is optimized for the environment of the joint research project at the universities of Bremen, Hannover, and Osnabrück, all using Stud.IP as Learning Management System. The REST API of the backend server allows for the implementation of other Frontends, for instance, for other LMS, such as ILIAS or MOODLE. The Django web framework, building the foundation of the backend server, supports Html views that, in principle, allow the implementation of a stand-alone web-server version by the developer community and Higher Education Institutions.

The SIDDATA platform presented in this work offers a universal tool for implementing automated recommendation features for higher education and tools to measure and reflect upon their effectiveness. Hence, the system's usefulness at large is ultimately determined by the features realized on the platform. We, therefore, call for an active exchange between learners, teachers, and programming personnel to design these features. In the future, the SIDDATA DSA may facilitate a more hands-on collaboration between multiple disciplines by providing a technical basis for recommendation feature implementation, thereby making a directly experienceable method for rapid feature development possible.

5 Acknowledgment

The authors acknowledge the financial support from the Federal Ministry of Education and Research of Germany for Siddata (project number 16DHB2124). We also acknowledge support by Deutsche Forschungsgemeinschaft (DFG) and Open Access Publishing Fund of Osnabrück University.

6 References

[1] Django Software Foundation, "Django: The web framework for perfectionists with deadlines," *Djangoproject.Com*, 2013.

[2] J. Schrumpf, F. Weber, and T. Thelen, "A neural natural language processing system for educational resource knowledge domain classification," in *DeLFI 2021 – Die 19. Fachtagung Bildungstechnologien*, 2021, pp. 283–288.

[3] K. Schurz, J. Schrumpf, F. Weber, F. Seyfeli, and K. Wannemacher, "Towards a user focused development of a digital study assistant through a mixed methods design," in *18th International Conference on Cognition and Exploratory Learning in the Digital Age, CELDA 2021*, 2021, pp. 45–52.

[4] R. Stockmann and A. Berg, "Stud.IP," *Eleed*, vol. 1, no. 1, 2005.

[5] M. Stonebraker and G. Kemnitz, "The POSTGRES next generation database management system," *Commun. ACM*, vol. 34, no. 10, pp. 78–92, 1991. https://doi.org/10.1145/125223.125262

[6] F. Weber, J. Kernos, M. Grenz, and J. Lee, "Towards a web-based hierarchical goal setting intervention for higher education," in *18th International Conference on Cognition and Exploratory Learning in the Digital Age, CELDA 2021*, 2021.

[7] H. T. Hien, P. N. Cuong, L. N. H. Nam, H. L. T. K. Nhung, and L. D. Thang, "Intelligent assistants in higher-education environments: The FIT-EBOt, a chatbot for administrative and learning support," *ACM Int. Conf. Proceeding Ser.*, pp. 69–76, 2018, https://doi.org/10.1145/3287921.3287937

[8] A. Yuce, A. M. Abubakar, and M. Ilkan, "Intelligent tutoring systems and learning performance: Applying task-technology fit and IS success model," *Online Inf. Rev.*, vol. 43, no. 4, pp. 600–616, 2019, https://doi.org/10.1108/OIR-11-2017-0340

[9] J. Sjöström and M. Dahlin, "Tutorbot: A chatbot for higher education practice," in *Designing for Digital Transformation. Co-Creating Services with Citizens and Industry*, 2020, pp. 93–98. https://doi.org/10.1007/978-3-030-64823-7_10

[10] O. Viberg, M. Khalil, and M. Baars, "Self-regulated learning and learning analytics in online learning environments: A review of empirical research," *ACM Int. Conf. Proceeding Ser.*, pp. 524–533, 2020, https://doi.org/10.1145/3375462.3375483

[11] A. Maedche, C. Legner, A. Benlian, et al., "AI-Based Digital Assistants," *Bus. Inf. Syst. Eng.*, vol. 61, pp. 535–544, 2019, https://doi.org/10.1007/s12599-019-00600-8

[12] C. C. Aguirre, C. D. Kloos, C. Alario-Hoyos, and P. J. Muñoz-Merino, "Supporting a MOOC through a conversational agent. design of a first prototype," *2018 International Symposium on Computers in Education (SIIE)*, 2018, pp. 1–6, https://doi.org/10.1109/SIIE.2018.8586694

[13] F. Molnár-Gábor, "Germany: A fair balance between scientific freedom and data subjects' rights?" *Hum. Genet.*, vol. 137, pp. 619–626, 2018, https://doi.org/10.1007/s00439-018-1912-1

7 Authors

Felix Weber (fweber@uos.de) is a research fellow and Ph.d. candidate at the Institute of Cognitive Science at Osnabrueck University. He is interested in building research cooperations on topics related to digital assistants for the individualization of studies in higher education.

Johannes Schrumpf (jschrumpf@uos.de) is a researcher at the Institute of Cognitive Science at Osnabrueck University and a Ph.d. candidate for Cognitive Science in the domain of Artificial Intelligence. In his research, he explores the capabilities of natural language processing systems to extract relevant features from educational resource descriptions for recommendation systems.

Niklas Dettmer (ndettmer@uni-osnabrueck.de) is a master's student of Cognitive Science at Osnabrueck University with a heavy focus on the entire development stack of AI-assisted software applications.

Tobias Thelen (tobias.thelen@uni-osnabrueck.de) is deputy managing director of the Center for Digital Teaching, Campus Management and Higher Education Didactics at Osnabrueck University and member of the Artificial Intelligence group at the university's Institute of Cognitive Science. He is working on the joint development of Open Source software for higher education and is interested in fostering development cooperation between higher education institutions. Tobias' research focuses on recommender systems for learning and analyzer components for adaptive learning systems.

Article submitted 2022-04-22. Resubmitted 2022-09-16. Final acceptance 2022-09-29. Final version published as submitted by the authors.