EPiC Series in Computing
Volume 95, 2023, Pages 332–344
Proceedings of European University Information Systems Congress 2023

Integrating AI Tools with Campus Infrastructure to Support the Life Cycle of Study Regulations

Markus von der Heyde[1*], Matthias Goebel[2*], Dietmar Zoerner[4+], Ulrike Lucke[3+]
[1,2] SemaLogic UG, Germany
[3,4] University of Potsdam, Germany
*firstname.lastname@semalogic.de, +firstname.lastname@uni-potsdam.de

Abstract

When implementing study regulations in student information systems, teachers' intuitive ideas often differ from the later IT mapping. In this article, a rationalisation (both economic and psychological) of such processes is being pursued. The ultimate goal is an AI-based assistance system which provides support for the generation, validation, accreditation and use of study regulations. Advanced AI tools could support a wide range of business capabilities within curriculum design as identified in the Higher Education Reference Model (HERM). Further objectives of this approach are to increase the consistency of individual study planning with study regulations and to support study guidance. Our symbolic approach applies common semantics of natural language and abstract logic which acts as a bridge between the legal norm and the course offerings. The article explains the general concept, introduces the developed technological architecture and presents single tools as well as their integration with the existing IT infrastructure on campus. The curriculum design use case involving extraction and interpretation of the coded knowledge, collaborative editing of study regulations, and fine-grained versioning of study regulations within the creation process is demonstrated and discussed. Finally, the resulting benefits, remaining challenges and future directions are discussed.

Keywords: microservice architecture, semantic modelling, higher education, artificial intelligence, curriculum life cycle, student information system

[1] https://orcid.org/0000-0002-6026-082X
[2] https://orcid.org/0000-0002-4671-0900
[3] https://orcid.org/0000-0003-4049-8088

1 Challenges in the validation of study regulations

Many processes which implement the general capabilities of higher education institutions (HEIs) are very specific to the institution, but are rather similar when viewed across different institutions. German HEIs stated that in areas where a high degree of digital tools was established, less than 40% of the solutions offered data interfaces, persistently connected their data to other services or offered quality measures (von der Heyde, 2022). Areas which were expected to change within the upcoming five years make up about 70% of all areas, and are also currently not yet connected to others due to a lack of interfaces, are not persistent and do not provide quality measures (ibid, p. 30/31). In sum, many digital tools are isolated solutions without any state-of-the-art exchange of data. As data transfer and data quality is a crucial point for any institution, the application of enterprise service bus technologies, an overarching data and application architecture and automated quality control measures appear of high importance. The digital processes along the life cycle of student curricula are especially underdeveloped in comparison to other parts of the student life cycle (ibid, p. 25). It is to be expected and confirmed in conversations with selected teams, that most of the processes in the curriculum design rely on standard office applications.

This situation is investigated here using the University of Potsdam as an example. It has a distinctive profile in the areas of digitalization and quality management in higher education (Hafer et al., 2021). In 2012, it was one of the first universities in Germany to receive system accreditation - i.e., it has acquired the right to self-accredit its study programmes through structured measures for quality assurance - and is one of four partners in the Quality Audit network. The understanding of good teaching is laid down in a participatory mission statement, which is underpinned by strategy papers for IT, e-learning, and internationalisation, to name just a few. A broad and efficient quality management system is anchored in the university via the Center for Quality Development in Teaching and Studies and with decentralised representatives at the faculties. Among other things, the development of study and examination regulations (based on a central regulatory framework for bachelor and master programs) and the evaluation of courses are supervised here. The planning of the courses offered is also carried out in conjunction with central administration and decentralised teaching units. Particularly noteworthy is the use of the Potsdam time slot model, which, despite multi-subject and interdisciplinary courses, provides over 85% freedom from overlap and thus makes a major contribution to the ease with which a program can be taken (referred to as “study-ability”).

Despite the advanced status of the overall quality management at the University of Potsdam, the supporting tools are often not integrated or do not provide any of the required data. Quality measures are manually performed, often supported by simple spreadsheet overviews. The integration of resulting data into the following process is performed manually, thus is prone to errors, slow, and demanding high effort.

The integrated student information system (based on the HIS Gx software) has been in use for a long time and supports the implementation of the more than 100 degree programmes offered. A divergence can be observed between the intuitive ideas of the teachers, expressed in the verbal presentations of the subject-specific regulations, and their representation as a logical set of rules in that system, which causes frustration for all parties involved and, above all, impairs the study-ability.

This problem is addressed with the solution presented here. The use of AI aims to improve processes around the description, validation and interpretation of study regulations. It follows a logic-based approach that combines natural language and symbolic rules to make these study regulations understandable to humans and machines at the same time. In order to enable subject representatives to design, test and use this approach and at the same time to increase AI competence in general, an offer of advanced training is being developed to accompany the technical developments. Thus, the following overarching goals are being pursued:

- Automated testing of study regulations will increase their structural validity as well as their consistency with individual study plans.

- Through a deeper understanding of AI, university stakeholders will be empowered to use and participate in the future development of AI systems.
- Formally audited study programmes will enable study guidance that satisfies individual requirements and legal frameworks alike.

Thus, the quality of higher education will increase through the practical implementation of AI in the organisation of studies. By following a generic microservice approach, all components are robust, scalable, and also easily transferable to other HEIs.

The remainder of this article is structured as follows. In section 2, a closer look at the processes related to study regulations is provided, and entry points for further assistance are identified. Section 3 introduces the main technological results of our work, i.e., the AI-based approach we follow, the architectural concept to integrate it with the existing IT infrastructure, as well as selected components and tools within this architecture. Finally, a summary of current results and an outlook to future work are presented in section 4.

2 Analysing the life cycle of study regulations

The general process model can be described using the Business Capability Model (BCM) of the Higher Education Reference Model (HERM). The model is published and maintained by CAUDIT in collaboration with EDUCAUSE and EUNIS. Figure 1 focuses on a portion of what is called Level 1 of the model (BC001) and arranges the included capabilities along a generic curriculum life cycle. Around the capabilities are named typical stakeholders that shape the processes underlying those capabilities. The individual processes of a specific institution were not originally designed by the application of this model. However, it illustrates a general approach that universities can use to compare their organisational structures without over-emphasising differences at the process level.

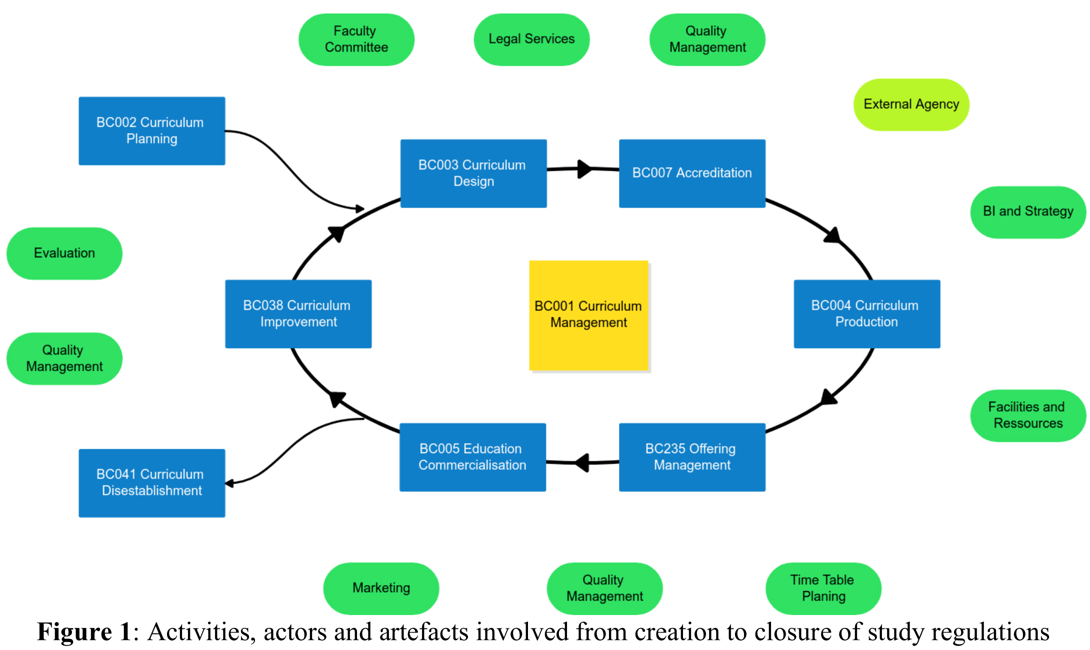

Figure 1: Activities, actors and artefacts involved from creation to closure of study regulations

Following the HERM definitions[4], *Curriculum Management (BC001) overall designs and produces, or sources, structured learning activities such as courses, subjects, and units, and ensures the institution is able to deliver them.* It contains all other capabilities described in Figure 1. The life cycle entry point is *Curriculum Planning (BC002) which researches the need, demand, and opportunities for educational products, such as courses and subjects, and decides which will be developed and offered.* New and existing curricula are formally defined in study regulations. *Curriculum Design (BC003) produces complete specifications of structured learning activities such as courses, subjects, and units.* Typically, those steps involve the local faculty as well as the legal and administrative services of the university. *Accreditation (BC007) is the assessment of the institution's curricula against the standards set by an accrediting authority.* Beside this external body, internal teams ensure and support this quality management in a structured way. *Curriculum Production (BC004) develops and provides resources needed to implement curriculum designs as deliverable educational products.* Therefore, facility management and other resource planning capacities are needed for the implementation. *Offering Management (BC235) plans and determines which of the institution's educational products will be made available for students in any particular teaching session.* This should include the requirements derived from the cohorts being enrolled in the various study programmes. In general, *Education Commercialisation (BC005) redesigns or redevelops products for sale in markets for which the current designs are unsuited.* In the German market, at least for publicly funded universities, this capability is probably implemented differently from private universities in Germany and the EU. *Curriculum Improvement (BC038) adjusts the curriculum in response to past or predicted curriculum performance.* Here, the results of the regular evaluation should be used to ensure a valuable delivery of the curriculum in the learning processes. To complete the life cycle, *Curriculum Disestablishment (BC041) manages the discontinuation of curriculum components* and manages the fade out of single study regulations.

In general, the HERM-based system covers most aspects of this curriculum life cycle which are directly or indirectly connected to the quality of study regulations. To apply AI-based assistance systems, the main process steps were identified, which are performed within the institution and typically with a low degree of digital assistance systems. The number and depth of necessary interfaces towards existing systems was therefore limited to a minimum. In this article, we focus on the Curriculum Design (BC003) activities as examples for the AI-based assistance system.

3 An AI-based assistance system to support the generation, validation and use of study regulations

In this section, the overall approach and architecture of the developed system along with its integration into the existing IT infrastructure is described. Then, selected parts in this architecture are presented. Our current focus is on the accreditation workflow, including basic modelling and editing of a study regulation, versioning and quality control. Special attention is paid to seamless integration with established processes and tools.

3.1 AI-based approach

We use a human-centric AI approach that complements, but does not replace, human capabilities. Our AI-powered systems should be open, transparent, and understandable so that they can work with humans and be compatible with their values and goals. We deliberately use symbolic rather than stochastic methods. One might argue that past course enrollments and grades provide information about good academic progress. However, we assume that the currently available data potentially do not

[4] which are quoted from https://www.caudit.edu.au/EA-Framework and set in *italics.*

represent desirable student behaviour, but only that which has been practised so far. In addition, recommendations based on previous course progressions lead to mainstreaming, which is not in line with our understanding of academic education and culture. Therefore, although we supplement the symbolic AI approach with concrete data at selected points, we do so not for statistical analysis and pattern recognition, but rather for adaptation to actual needs.

The general approach is that rule texts like a study regulation should be understood equally by humans and should be evaluated by automatic algorithms without room for interpretation. It is essential that such a formulation, which is close to logical programming, is nevertheless understood by experts in the professional context of the modelled object and is not only accessible to computer science experts or programmers. In general, the modelling of rule texts allows algorithms to follow the contained rules not only in the sense of a program, but to check them by automatic validation or to determine the solution space by combinatorics.

	Natural Language	Formal Specification Language	Technical Language
brief definition	... is the normal language people use to communicate. The focus here is on legal language in the context of study regulations, so the use of figurative and artistic descriptions is not expected.	... is a simplified language style that follows a standardised grammatical structure. A set of keywords start rules that define the intended logical structures.	... is a context-based programming language in which a predefined set of keywords allows rules to be formalised.

example	`The master’s degree program in [] consists of the following components:`
	• `Master Thesis`
	• `Mandatory Modules`
	• `Optional Modules`
		`The name of master’s degree program is called []. This master’s degree program consists of the Master Thesis, the mandatory modules and the optional modules.`	`name:master's degree program := Cognitive Systems;`
			
			`master's degree program [Master Thesis, Mandatory Modules, Optional Modules];`

typical use case	Natural usage: • Written statements in a study regulation Technical usage: • Static form e.g. PDFs • Natural language processing and translations	Natural usage: • Formal definition as a future form of study regulations • Collaborative editing Technical usage: • Version-controlled representation of rules • Baseline for dynamic feedback	Natural usage: • Programming language for experts Technical usage: • Validation of rules • Symbolic handling of contained rules

Figure 2: Interplay of natural language, formal specification language and technical language in the description of a study programme.

The modelling of the structured logic of study regulations is performed with SemaLogic as consistent mapping between technical language and the human-readable version (von der Heyde & Goebel, 2020). A consistent set of rules is derived from the formal specification text (see Figure 2). This set of rules, including constraints, is provided automatically and unambiguously from the textual or graphical SemaLogic representation as the basis for applying Answer Set Programming (ASP; Gebser et al., 2012). ASP has already proven its suitability for solving Timetable Management (BC027) problems (Banbara et al., 2019). In general, rule-based approaches are ideally suited for modelling legal text like study regulations which describe ordered processes. Once modelled, it can be used to answer a wide variety of queries in addition to consistency tests, such as for possible incompletions or incompatibilities, and also for optimal progressions using quantitative or qualitative criteria. This is far more powerful than, for example, the approach of Catala (Merigoux et al., 2021), where the program code for the representation of legal texts is created directly in a technical language. In the future, additional constraints will be considered, such as:

- logistical constraints due to multiple campus locations,
- limitations due to available teaching capacity,
- overlap-free study planning for different subject combinations,
- alternative forms of teaching or examination in the case of certain impairments.

Taking those constraints, one would be able to match additional personal preferences of the users and adapt the output of the AI-based assistant system. Thus, features to be preferred or avoided can be modelled, such as:

- as many elective options in the second subject as possible,
- as many projects or seminars as possible,
- as few module exams per semester as possible,
- as few changes of location within a day as possible.

The conditions and options of study-ability to be considered are first collected in interviews with those responsible or affected, and afterwards formally modelled. By means of an ASP solver from Potassco[5], solution sets (e.g., as study-ability in a given period of time or with individual constraints) can be tested efficiently. In addition, automated indications of conflicting rules or requirements can be generated by iterative optimizations. Thus, direct hints for order generation and validation are derived from solvability in ASP and modelling in SemaLogic (von der Heyde et al., 2023).

Currently, only a limited set of study regulations was translated from the original natural language into formal language by the application of SemaLogic. While translating existing study regulations, any practical hurdles and the future potential of the approach will be evaluated. Only afterwards is it planned to establish a new procedure for the formulation of new regulations and the internal review loops. For this purpose, legal, statistical and expert knowledge will be surveyed and systematised among those responsible in the departments.

3.2 Overall architecture

The architecture follows the principles of microservices, so that certain services each perform a dedicated and manageable task. The following architecture describes a template for the different support systems. Each of those support systems consists of a specific user interface and use case controller. The strategy for dividing the problem into the respective services applies principles of domain-driven design. The use case controller contains the logic to fulfil the operations with suitable underlying microservices. Its API is the only visible interface to the outside world.

The Semantic Language Service handles transformations of formal language texts. Its output can be transformed to Answer Set representations and processed by the ASP Service to reply to complex queries of the particular use case. The Versioning Service persists across different working stages of formal study regulations. The Personal Data Service manages access rules of the different users and saves personal preferences. The SIS Service retrieves data on personal student information, student achievements and course offers.

[5] freely available under https://potassco.org, based on Gebser et al. (2012)

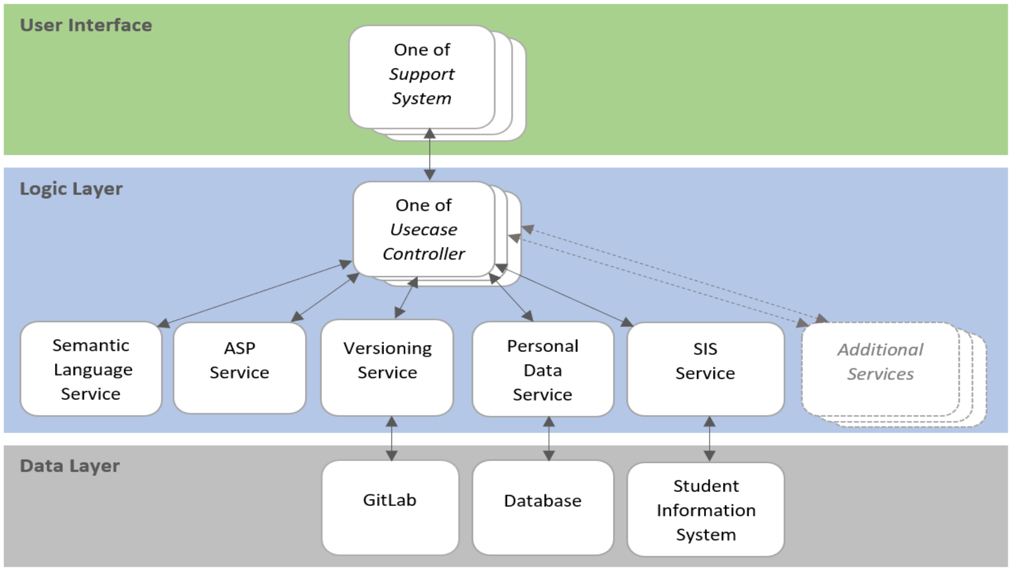

Figure 3: Integration of the developed support system in the existing infrastructure.

This type of architecture ensures these advantages in particular:

- flexibility and maintainability
- reusability of parts of the system
- high potential for integration of standard tools

The deployment of our system uses standard tools where applicable to support sustainable development. The Versioning Service is implemented by using the open API of the local GitLab installation. The database is implemented by a standard PostgreSQL installation. The SIS is encapsulated via a specific API and thus prepared for the use of other SISs.

In addition, all services are open to additional requirements such as other data formats. Each component is designed to be robust against misuse and missing data. Stability and open interfaces are essential for the integration with a system, where each component can be easily replaced at any time. Robustness, maintainability and flexibility in terms of load capacity and extensibility of the services used is achieved by using Docker in swarm mode. TCP/UDP load balancing is handled by nginx, and Traefik is used for access control and routing between the services.

3.3 Supporting the process of creating study regulations

As mentioned above, the capability *Curriculum Design (BC003)* requires the definition of study regulations as a series of structured learning activities. Already in this creation process, the use of a common language should not only enable faculty members and the central services to have the same understanding of the formal regulations, but also make the technical interpretation of the text transparent and show the resulting effects. The direct transformability of the technically generated text into logical models also makes it possible to provide feedback through direct rule validations and checking of the solution spaces while the text is still being created. Systematic errors as well as accidental misunderstandings become immediately visible. The studyability of the resulting curriculum can thus be validated during the creation process.

Study regulations are inherently subject to changes over time. Each change should be visible by itself, but also has an effect on the remainder of the document. Changes concerning logical rules may be spread across the entire document. Therefore, the ability to review several versions in parallel and to compare the resulting logical structure of a ruleset at a glance is crucial to an effective collaboration of those designing a curriculum. On an abstract level, there are specific requirements for technical tools to support this creation process of study regulations:

a) Extraction and interpretation of the coded knowledge
b) Collaborative editing of study regulations by faculty staff and central services
c) Fine grained versioning of study regulations within the creation process

In the following, each of these requirements and reasons for the chosen solution are described in more detail, and the specific use case and implementation is introduced.

The *rules or knowledge encoded in the study regulation can be automatically extracted and interpreted (a)* to allow for feedback and further use. Graphical representation of the rules provide a unique overview and support the easy recognition of nested constellations. To provide real-time feedback during the editing, the capability for the formal language interpretation needs to be integrated into the editor. Following the microservice approach, this is implemented using a plug-in concept which enables a wide range of editors to integrate a common feature. Web services like Grammarly[6] follow a similar approach, as they offer a service ready to be integrated into any text editing in a browser. The SemaLogic service was therefore extended by an OpenAPI interface offering stateless and stateful communication (von der Heyde et al., 2023). Any editor (web-based or desktop application) can be extended by its own plugin to integrate the text analysis and feedback provided by the Semantic Language Service, which could be consumed from the logic layer. At the same time, this service can be used to formalise and interpret the study regulations for further usage in other use cases. As the interpretation of the formal language is identical in editing and further use, maintenance of the same source of information for the derived logical representation is ensured.

For *collaborative editing of study regulations (b)*, any editor with open interfaces or a plugin concept can be extended to use the formal language service provided. Unlike established stand-alone desktop applications, modern web-based applications often offer collaborative elements and use open or standardised formats. In contrast, office applications are often used to customise the design of documents to match corporate design. In view of the necessary long-term availability of study regulations and their simultaneous representation as a formatted document, the Markdown markup language was chosen as the central element for the editing process. There are a number of suitable Markdown editors available, some of them free of charge. For simplicity, it was decided to use Obsidian[7], a Markdown editor that offers an extensive community-created plugin collection. The wide variety of examples enabled rapid prototyping. Obsidian also offers a suitable user interface, fast inline display of Markdown formatting, and a transparent directory-based vault file storage.

When collaborating on texts, a sophisticated *management of versions (c)* is necessary to trace changes. The ability to manage, display, and store consistent versions and the changes between these versions has existed for a long time in the context of programming and is realised there by systems such as RCS, CVS, or Git. The ability to manage alternative development branches and merge them later is a valuable feature of Git. At the same time, Git has established itself as a backend service for storing system configurations to enable a dynamic CI/CD process in agile software development. Git also works very well on flat, Markdown formatted texts and was therefore an ideal choice. GitLab provides a REST API to its underlying Git repository and is able to provide any required operation. To reduce the complexity of interacting with the GitLab API, a wrapper service has been developed. This service transfers the appropriate access tokens and simplifies the operations required. GitLab is only used for version management in the background, so the authors of the study regulations are unaware of Git and its technical concepts.

[6] see: https://grammarly.com
[7] see: https://obsidian.md/

We will now describe the interaction of these components on the basis of a use case: the combination of intuitive editing of a study regulation (in Obsidian), real-time feedback of coded knowledge (through SemaLogic), and a shared file storage structure including version management (using Git). The SemaLogic API calls were fitted into an Obsidian plugin, distributed via the Obsidian community platform or the local versioning service. User preferences and technical settings are implemented in standard configuration dialogs. The study regulation as a result of the collaborative editing can be checked into any Git structure. As the collection of study regulations across the university would require a structured approach, the Version Service representing the GitLab API will be connected to stage the major version of the study regulations in the long run.

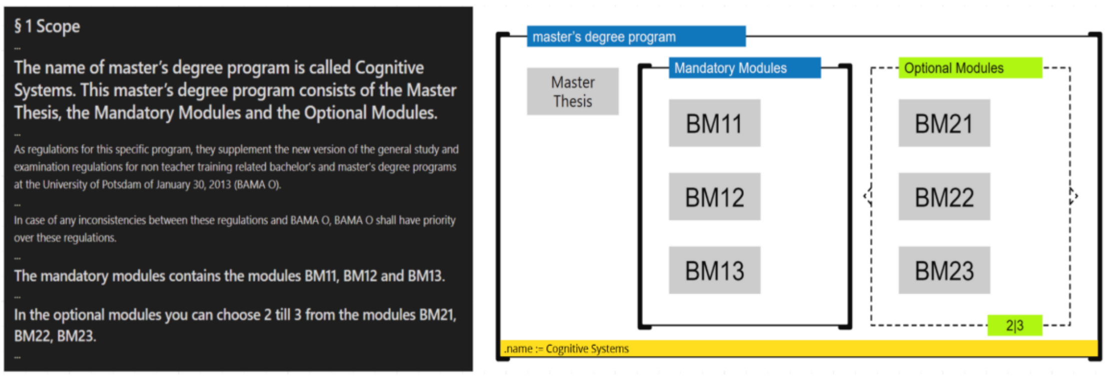

Figure 4: Editing study regulations with an extended version of the Obsidian editor.

Figure 4 displays the Obsidian editor during a potential editing process. On the left-hand side, the author writes the study regulations in formalised, yet natural language. As the text is constantly sent via the API to the Semantic Language Service to be parsed and interpreted in the background, the graphical response is continuously generated. On the right-hand side, all recognised rules are displayed as an overview diagram generated by the SemaLogic service. As demonstrated by von der Heyde & Goebel (2021), the analysis of structural content of study regulations is of high value in the comparison of curricula, and potentially in the comparison of versions of the same study programme as well. The parsing result could alternatively display the rules in technical standard language or convert the contained knowledge into a standardised JSON structure ready to be read into ASP.

The strong service orientation enables the benefits of different applications to be combined without making the whole system dependent on individual components. Each component can be replaced at any time with limited effort, thus optimally supporting the agile approach and the objectives of the project. The modular interchangeability of the individual services is a major advantage, as the decision on which components to operate on a permanent basis is made at the end of the project, after the overall benefits have been demonstrated.

3.4 Supporting the ongoing use of study regulations

Working with formalised study regulations supports individual study planning and institutional instructional planning on various levels. To support the application of the concept in other HEI, we align our use cases along the HERM in Table 1.

Table 1: Overview of use cases matching the Business Capability Model (BCM) of the Higher Education Reference Model (HERM)

HERM Capability	Use Case
BC003 Curriculum Design	The interactive development of the curriculum with real-time feedback on the coded rules reduces minor errors during the editing process and improves consistency within the model of the study programme. The structured result is used in the other use cases.
BC007 Accreditation	The internal validation of each study regulation follows the local quality management procedures. The ancillary conditions of underlying regulations and external requirements for accreditation can be supported by formal examination routines.
BC004 Curriculum Production	From a curriculum planning perspective, a recommendation can be made as to which modules should receive special consideration in future course offerings based on the regulations to be covered in a teaching unit and the courses offered in previous semesters.
BC235 Offering Management	All requirements for available teaching staff time slots and room capacities are fitted into schedules that correspond to the requirements of the cohorts (like number or special needs of students) in the modelled study programmes.
BC005 Education Commercialisation	The number of study progressions that can be studied for a degree in the standard period of time can be generated from the formalised regulations in comparison with the previous course offerings and form a basis for planning and advising processes.
BC027 Timetable Management	From the student's perspective, a recommendation can be made as to which courses would be possible for a degree in the chosen major based on the applicable regulations, previous performance, individual preferences and current course offerings.

Particular attention is paid to supporting students with disabilities. In this context, specific restrictions (beyond the standard set formulated in the regulations) must be taken into account. First of all, the module descriptions already recorded in a central database and supplemented by risk assessments provide information in this regard. In addition, the disadvantage compensations practised in individual cases must be taken into account. In this way, the activities of student advisors, which have so far been based heavily on empirical knowledge, can be supported more systematically and contribute to fairness throughout the university.

In general, we take the protection of personal data into account by only allowing the person concerned to enter the data into the assistance system themselves, where they are only used for their individual study planning. Previous study performance is only included in the recommendations if the credentials of this user are transferred. Other personal data cannot be read via the interfaces[8] used. At the same time, this expresses our basic ethical understanding that AI systems must be transparent, serve

[8] API available under https://apiup.uni-potsdam.de/devportal/apis (Kiy et al., 2017)

people and support their sovereignty. This is what is required by the relevant guidelines[9] and what we practise in our research and development (Müller et al., 2020). Thus, the contribution of our assistance system to a sovereign decision becomes at the same time a quality criterion, which is considered in the design and the evaluation of the system. However, the variety of assistive tools supporting such processes is outside the focus of this article and thus omitted here.

4 Conclusion and outlook

The digital processes to generate and change student curricula are underdeveloped in comparison to other parts of the student life cycle. When editing study regulations, the teachers' intuitive ideas often differ from the implementation of rules and constraints in student information systems. The overall goal of the approach presented in this article is to prototype multiple use cases in Curriculum Management (BC001) with reference to the HERM schema. An AI-based assistance system, which consists of multiple use case-oriented modules, is provided to support e.g., the generation, validation, accreditation and use of study regulations. The objectives of this approach are to simplify individual study planning and to improve the overall consistency of the regulations. A common semantics of natural language and abstract logic acts as a bridge between the legal norm and the course offerings; thus, a symbolic approach to the validation and solvability of regulations is pursued.

To be able to develop multiple use cases, which often share common tools and interfaces, we have proposed a generic application architecture. This microservice architecture allows a robust and flexible integration into campus IT, reusing existing interfaces and also providing additional ones. This paper has focused on the use case of the editing process of study regulations. Single steps as rule extraction, collaborative editing, and real-time user feedback were introduced to support the process with further quality measures. We demonstrated the flexible integration of standard infrastructure components (Docker, Git, Spring Boot etc.) and standard protocols (JSON) via OpenAPI defined interfaces.

The study regulations are formalised in a specification language (SemaLogic) and are thus subject to validation procedures which support the accreditation process by automated quality measures. Further use of the formalised study regulations is motivated with respect to the remaining capabilities of Curriculum Management in HERM. Our next steps will address those additional use cases and develop prototypes accordingly.

5 Acknowledgements

This work is funded by the German Federal Ministry of Education and Research under contract number 16DHBKI024. We are deeply grateful to the whole CAVAS+ project team as well as to our partners in various organisational units at the University of Potsdam. Without their help and support, the work presented here would not have been possible.

6 References

Banbara, M.; Inoue, K.; Kaufmann, B.; Okimoto, T.; Schaub, T.; Soh, T.; Wanko, P. (2019). teaspoon: solving the curriculum-based course timetabling problems with answer set programming. Annals Operation Research, 275(1), 3–37. DOI: 10.1007/s10479-018-2757-7

[9] https://gi.de/ueber-uns/organisation/unsere-ethischen-leitlinien, https://claire-ai.org/vision/, https://www.humane-ai.eu/vision/, etc.

Gebser, M.; Kaufmann, B.; Schaub, T. (2012). Conflict-driven answer set solving: From theory to practice. Artificial Intelligence, 187, 52–89. DOI: 10.1016/j.artint.2012.04.001

Hafer, J.; Kostädt, P.; Lucke, U. (2021): Das Corona-Virus als Treiber der Digitalisierung? Eine kritische Analyse am Beispiel der Universität Potsdam. In: Wie Corona die Hochschullehre verändert / U. Dittler, C.Kreidl (Eds.), Springer, 219-242. DOI: 10.1007/978-3-658-32609-8_15

von der Heyde, M. (2022). Ergebnisse der Umfrage zur Digitalisierung der Hochschulen. vdH-IT. DOI: 10.5281/zenodo.6948103

von der Heyde, M.; Goebel, M. (2020). Die Sprache «SemaLogic» als semantische Repräsentation—Eine anforderungsbasierte Sprache zur Modellierung von Prüfungsordnungen und Abbildung von Studienverläufen. INFORMATIK 2020, Back to the Future, 521–536. DOI: 10.18420/inf2020_48

von der Heyde, M.; Goebel, M. (2021). Structural comparison of curriculum design — Modelling international study programs using a logical language and its graphical representation. 15th annual International Technology, Education and Development Conference (INTED). DOI: 10.21125/inted.2021.0631.

von der Heyde, M., Otunuya, H. C., Goebel, M., Zoerner, D., & Lucke, U. (2023). Automatic and Interactive Validation of Study Regulations in Higher Educations Accreditation Processes. European University Information Systems Conference (EUNIS) 2023. EPiC Series in Computing, EasyChair.

Kiy, A.; List, C.; Lucke, U. (2017): A Virtual Environment and Infrastructure to ensure future readiness of Data Centers. European Journal of Higher Education IT (EJHEIT) 2017-1.

Merigoux, D.; Chataing, N.; Protzenko, J. (2021). Catala: A Programming Language for the Law. arXiv:2103.03198 [cs]. http://arxiv.org/abs/2103.03198

Müller, I.; Moebert, T.; Lucke, U. (2020): Lessons Learned from Designing Adaptive Training Systems: An Ethical Perspective. In: Artificial Intelligence Supported Educational Technologies / N. Pinkwart, S. Liu (Eds.), Advances in Analytics for Learning and Teaching, Springer, 273-290. DOI: 10.1007/978-3-030-41099-5_16

7 Author biographies

Dr. Markus von der Heyde received his PhD with topics in cognition research at the Max Planck Institute for Biological Cybernetics in Tübingen. Since 2011, Dr. von der Heyde has been advising colleges, universities, and public cultural and research institutions on a wide range of digitalisation topics (governance, organisation, strategy) as part of vdH-IT, and conducts independent research on these topics (see ResearchGate). Since 2018, he has been an Adjunct Professor at the School for Interactive Arts and Technology (SIAT) at Simon Fraser University, Vancouver. Dr. von der Heyde is also active as a volunteer in a variety of non-profit organisations (GI, ZKI, EUNIS, Educause). In 2020, he founded SemaLogic UG to use semantic and structural logic technologies to automatically map and validate natural language regulatory texts. See further details at LinkedIn, or Google Scholar.

Matthias Goebel has been active in numerous IT projects for the introduction or optimisation of SAP systems and SAP-based applications since 2000. For more than 10 years he managed the SAP divisions of various companies with regard to the company-wide SAP strategy and architecture. Application-related focal points are enterprise application integration, programme and DB-based performance optimisation, data warehousing and the redesign and modification of digitally supported processes.

Dr. Dietmar Zoerner studied computer science at TU Berlin. He has experience in the areas database and software design, software engineering and software architecture. From 2010 to 2018 he worked as a research assistant as a member of the chair for Complex Multimedia Application Architectures. In 2021 he did his PhD in the field of digital game-based learning. Since December 2021 he is working as a project coordinator at the chair for Complex Multimedia Application Architectures.

Dr. Ulrike Lucke is professor of computer science at the University of Potsdam, Germany. She obtained her PhD at the University of Rostock, Germany. Her research activities include institutional infrastructures for education, research and administration. Currently, among other activities, she coordinates a large-scale national initiative to create a digital ecosystem for education across Germany and acts as an independent evaluator for two European projects on policy experimentation in the educational sector. Until 2018, she was responsible for e-learning and IT strategy as Chief Information Officer of the University of Potsdam. She is a founding member and was vice chair of the German University CIO Association until 2020. Since 2020, she is a Vice President of the German Informatics (GI) society.