informatics

MDPI

Article

Governance-Oriented Decision Support for Internship Scheduling Using a GRASP Heuristic in University Psychology Clinics

Claudia Regina de Freitas [1], José Roberto Dale Luche [1,*] and Hugo Paris [2]

[1] Production Department, School of Engineering and Sciences, São Paulo State University—UNESP, Guaratinguetá 12516-410, Brazil; claudia.freitas@unesp.br
[2] École Centrale Lille, 59651 Villeneuve d'Ascq Cedex, France; hugo.paris@centrale.centralelille.fr
* Correspondence: dale.luche@unesp.br

Abstract

University psychology clinics represent complex coordination environments in which pedagogical requirements, supervision constraints, and institutional governance mechanisms intersect. Traditional coordination practices based on spreadsheets often fail to ensure transparency, consistency, and traceability in internship scheduling. This study addresses this challenge by adopting a Design Science Research (DSR) approach to design and evaluate a governance-oriented decision-support system for internship coordination. The proposed artifact models the scheduling problem as a rule-constrained allocation process and implements a modular architecture integrating a GRASP heuristic with randomized greedy construction, single-pass triadic repair, a post-generation validation layer, role-based access control, and audit logging mechanisms. The system generates feasible schedules while enforcing operational and pedagogical constraints and supporting peer-observation structures, including binary and triadic supervision arrangements. A real-world pilot deployment provided case-based evidence of scheduling feasibility and operational utility, while the controlled computational evaluation demonstrated complete coverage of the institutional instance and no detected structural violations in the evaluated schedules. Role-based access control, persistent audit logging, and the controlled approval/publication workflow were completed and technically verified subsequent to that pilot, extending rather than altering the scheduling and validation components exercised during it. The results provide case-based evidence of the artifact's utility and illustrate how a governance-oriented decision-support architecture can make rule validation, authorization, and scheduling provenance explicit within an institutional workflow. In the institutional instance evaluated with 30 randomized seeds, the single-pass triadic repair phase reduced the mean number of unassigned requirements from 3.60 to 0 while preserving feasibility across all runs.

Keywords: decision support systems; GRASP; internship scheduling; design science research; clinical education

Check for updates

Academic Editor: Weitian Tong

Received: 8 July 2026
Revised: 17 September 2026
Accepted: 26 September 2026
Published: 1 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Higher education institutions increasingly rely on digital infrastructures to support academic coordination, administrative efficiency, and governance. While substantial research has examined learning management systems and digital learning environments, comparatively less attention has been devoted to the digital transformation of operational

coordination processes in clinical-academic settings. University psychology clinics are particularly complex in this regard: they operate simultaneously as service-delivery units and supervised training platforms, coordinating educational, clinical, and administrative activities within the same institutional structure.

Digital transformation is understood as the integration of digital technologies into core processes in ways that reshape operational models and institutional practices [1,2], extending beyond technological adoption to include structural reconfiguration and governance redesign [3]. In higher education, transformation initiatives often prioritize student-facing technologies, while internal coordination systems remain comparatively underexplored [4,5].

The challenge here extends beyond digitization of existing procedures to the formalization of institutional coordination rules within infrastructures that support traceable and auditable decisions. Decision Support Systems (DSS) have long been recognized as mechanisms for supporting structured and semi-structured decision-making through the integration of data, analytical models, and organizational rules [6,7]. Their role becomes particularly relevant in coordination-intensive environments where operational decisions must remain consistent with institutional requirements.

Clinical-academic internship scheduling lies at the intersection of educational timetabling, constrained allocation, and organizational decision support. Educational timetabling encompasses multiple established problem formulations and benchmark traditions [8,9]. Recent medical-training studies allocate students or trainees to hospital internships, disciplines, rotations, and time periods under curricular, capacity, preference, fairness, and duty-hour constraints [10–12]. Related research has also formulated student–supervisor allocation as a constrained decision-support problem [13]. Together, these studies establish that internship, rotation, and supervision scheduling are not new as general problem classes. The specific challenge examined here is the coordination of peer-observation structures, individual supervision, rooms, availability windows, and institutional stages within a university psychology clinic.

A scheduling optimizer alone does not determine how institutional actors are authorized to initiate, inspect, validate, revise, and approve consequential allocations. Research on meaningful human control and effective human oversight emphasizes the importance of retaining the capacity of authorized users to question, revise, or disregard system recommendations [14,15], while auditing provides mechanisms for examining automated decision processes and their procedural conditions [16]. Within a design-science view of DSS artifacts [17], this study uses governance-oriented decision-support architecture as an operational framing rather than as an established DSS category. It denotes a configuration in which schedule generation is embedded in a controlled workflow that includes explicit rule validation, role-based authorization [18], auditable action provenance, reporting, and retained human approval. These mechanisms govern the organizational process surrounding the heuristic output rather than constituting part of the GRASP search itself.

Contemporary research highlights that digitally mediated decision environments increasingly require hybrid organizational structures in which automated systems and human actors jointly participate [19,20].

Clinical training programs in psychology introduce additional scheduling complexity. Interns must complete structured supervision requirements involving interconnected peer-observation structures and mandatory individual sessions. These pedagogical constraints interact with operational limitations such as room capacity, availability windows, and prohibition of double-booking, characterizing a rule-constrained resource allocation problem embedded within a pedagogical governance framework. Managing such interdependent

constraints through informal mechanisms becomes increasingly difficult as actors and scheduling interactions grow [21,22].

Spreadsheet-based coordination, although flexible, frequently lacks traceability, standardized rule enforcement, and scalability. As organizations grow and regulatory requirements intensify, the absence of audit trails, explicit access control, and systematic validation exposes institutions to coordination failures and compliance risks [23,24].

In response to these challenges, this study investigates the following research question: How can an integrated information system be designed to support governance, rule enforcement, and decision-making processes in university psychology clinics?

To address this question, the study adopts a Design Science Research (DSR) approach in which artifact development and evaluation are conducted iteratively. The resulting artifact integrates a formal representation of the scheduling problem, a GRASP-based scheduling service combining randomized greedy construction with single-pass triadic repair, and a modular web architecture implemented using FastAPI 0.115.6, PostgreSQL 16, and React 19.2. Evaluation combines computational assessment of the scheduling procedure across multiple randomized executions and observation of artifact operation within its intended organizational environment. This methodological framing aligns the study with the design-oriented tradition in information systems research, in which artifact construction contributes to both practical problem-solving and theoretical understanding.

This study contributes a case-grounded Design Science Research artifact for clinical-academic coordination [25,26]. First, it formalizes the specific coordination requirements of a university psychology clinic, including peer-observation structures, individual supervision, room and availability constraints, and internship stages. Second, it integrates an application-specific scheduling procedure with a distinct validation layer, role-based authorization, audit logging, reporting, and retained human approval within one decision-support architecture. Third, it provides an initial case-based evaluation of scheduling feasibility and artifact utility [27], while explicitly distinguishing implemented governance mechanisms from governance outcomes that were not directly measured.

By addressing internship coordination through the integration of scheduling, validation, and governance mechanisms, the study contributes to ongoing discussions regarding the role of information systems in supporting complex institutional decision processes [2,3].

The remainder of the paper is structured as follows. Section 2 analyzes the institutional context and formalizes the scheduling problem as a governance-oriented decision-support challenge. Section 3 presents the system architecture and design principles. Section 4 describes the research methodology and artifact development based on Design Science Research. Section 5 discusses the operational impact and its implications for coordination and governance. Section 6 concludes and outlines directions for future research.

2. Problem Context and Institutional Requirements

This section characterizes the institutional and organizational context that motivates the proposed system.

2.1. Digital Transformation in Clinical-Academic Settings

Digital transformation has been conceptualized as a multidimensional organizational process involving structural redesign, capability development, and governance reconfiguration [3,28]. Rather than mere technological substitution, it entails systematic integration of digital infrastructures into core decision processes, reshaping coordination mechanisms and institutional practices [2,29].

In higher education, digital transformation has been widely associated with teaching and learning technologies, digital student services, and online learning environments [4,5].

However, less attention has been given to internal coordination infrastructures that sustain institutional operations. This gap is particularly relevant in clinical-academic settings, where pedagogical requirements, service provision, supervision structures, and administrative routines must be coordinated within the same organizational process.

University psychology clinics combine educational supervision with the provision of clinical services, requiring the structured allocation of students, supervisors, rooms, and patients under formal pedagogical regulations. As enrollment expands, such coordination requires capabilities beyond automation: the strategic use of analytical and rule-based systems depends on organizational absorptive capacity and integration mechanisms [30]. More broadly, organizational capacity development depends on the coordinated development of dynamic capabilities and complementary organizational factors [31]. Related information-systems research has likewise shown that the organizational value of digital infrastructures depends on their alignment with organizational capabilities and decision processes, including those associated with IT-enabled organizational agility [32]. In this context, digitalization provides an infrastructure for integrating coordination, validation, access control, and reporting within a shared institutional workflow.

2.2. Scheduling and Constrained Allocation in Educational and Clinical Training Environments

Educational scheduling comprises a family of constrained allocation problems rather than a single canonical model. Recent reviews identify multiple standard educational-timetabling formulations, benchmarks, solution methods, runtime measures, and solution-quality criteria [8]. Contemporary university models may jointly assign people, events, rooms, modes, and time periods while balancing hard constraints and stakeholder objectives [9].

Clinical-training scheduling introduces additional curricular, placement-capacity, service, and individual-training requirements. Akbarzadeh et al. scheduled medical students to hospital internships under university requirements, hospital capacity, student preferences, and stakeholder requests [10]. Zanazzo et al. modeled medical-student scheduling across disciplines, hospitals, and periods and evaluated simulated annealing against exact-solution evidence [11]. These studies confirm that internship and rotation scheduling is an established and active application area.

User-facing decision-support systems also exist in neighboring domains. Shahraki et al. developed a MILP-based DSS for monthly rotations and daily trainee assignments incorporating program policies, staffing requirements, preferences, fairness, and duty-hour constraints [12]. Castaño and Velasco combined mathematical programming and variable-neighborhood search in a tool supporting generation, validation, and revision of medical-trainee rotation plans [33]. Student–supervisor allocation has likewise been formulated as a multi-objective decision-support problem [13].

The psychology-clinic problem shares the constrained-allocation structure of these studies but combines a specific supervision topology involving reciprocal peer observation, individual sessions, internship stages, rooms, and availability windows. The present study therefore does not propose a new general scheduling paradigm; the GRASP procedure is an application-specific scheduling service within a broader information-systems artifact.

2.3. Decision Support, Governance Mechanisms, and Human Oversight

Decision Support Systems combine data, analytical models, and interactive representations to support semi-structured organizational decisions [6,7]. DSR-oriented DSS research, in turn, emphasizes the construction and evaluation of purposeful artifacts [17]. In scheduling applications, this baseline generally includes a model or heuristic, institutional data, and an interface through which planners generate or compare allocations. Governance

introduces an additional design question: how the organization controls, checks, attributes, and authorizes the use of algorithmic outputs.

The governance mechanisms implemented in this study serve distinct functions. A post-generation validation step checks generated schedules against the institutional rule specification immediately after generation, within the same application process rather than at a separate service boundary. Role-based access control binds permissions to organizational roles and constrains authorized actions [18]. Audit logging preserves attributable action and state histories, supporting later inspection, while reporting exposes scheduling and validation results to authorized users. Human approval retains organizational authority for acceptance or revision of generated allocations.

These mechanisms should not be equated automatically with demonstrated governance outcomes. Research on meaningful human control emphasizes that human involvement is effective only when authorized users can understand, question, and appropriately disregard or revise system recommendations [14,15]. Similarly, auditing can support procedural regularity and transparency, but its effectiveness depends on technical and organizational conditions [16]. The present artifact therefore implements mechanisms intended to support inspectability, attributable decision processes, and retained human authority; their perceived effectiveness remains an empirical question.

2.4. Formalization of Coordination as an Information Systems Problem

Prior to system implementation, internship coordination relied primarily on spreadsheet-based tools and decentralized documentation. Although flexible, such mechanisms lacked embedded validation layers, traceability of changes, and systematic enforcement of pedagogical supervision requirements. As coordination complexity increased, conflict detection became dependent on manual verification and tacit organizational knowledge [23].

From an information systems standpoint, this situation can be viewed as a coordination problem in which multiple actors, resources, and institutional requirements must be synchronized through structured allocation and scheduling mechanisms. Similar challenges have been identified in complex service systems where organizational rules and resource constraints interact across multiple decision layers [21].

The conceptual abstraction shown in Figure 1 connects the institutional coordination context to a structured information systems design problem. It shows how clinical supervision structures, operational constraints, and pedagogical requirements generate coordination restrictions that can be formalized as a rule-constrained scheduling problem. This abstraction provides the foundation for a governance-oriented decision-support architecture capable of combining rule-constrained scheduling with explicit validation, access control, and traceability.

The core design challenge is therefore broader than the generation of feasible timetables. It involves the integration of pedagogical constraints, operational restrictions, validation procedures, and audit and authorization mechanisms within a coherent digital infrastructure capable of supporting traceable and auditable coordination processes.

The scheduling problem is formally expressed as a binary integer programming model presented in Appendix A. The formulation captures the allocation of students to internship sessions under pedagogical and operational constraints, including availability, supervision requirements, peer-observation relationships, and triadic supervision cycles. Exact optimization of the full institutional instance was tested under a fixed computational budget and is reported in the Computational Tractability Note in Appendix A. That test did not, by itself, establish optimal coverage; optimal coverage was instead established separately by mapping a complete GRASP-generated schedule into the exact formulation

(Section 4.5). The exact test did narrow, without resolving, the optimal number of triadic cycles. The proposed system therefore adopts the GRASP heuristic described in Section 3.4, which generated complete schedules for the evaluated institutional instance; a controlled runtime measurement, together with the complete computational environment and timing protocol, is reported in Section 4.6.

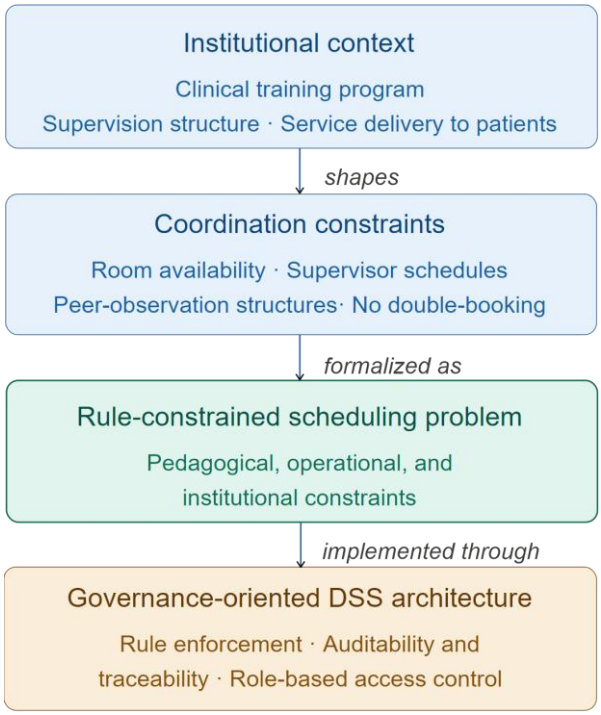


Figure 1. Conceptual abstraction of the clinical scheduling problem from institutional context to governance-oriented DSS architecture.

In this study, governance-oriented decision-support architecture denotes a DSS configuration in which schedule generation is embedded in a controlled organizational workflow: institutional rules are represented computationally; generated schedules are checked through a distinct validation layer; permissions are bound to organizational roles; consequential actions and state changes are logged; outputs and validation results are reportable; and authorized humans retain approval and revision authority. The term is used as an analytic framing for this study rather than as a claim of a separate canonical DSS taxonomy.

Recent internship and rotation-scheduling studies address institutional requirements, capacities, preferences, fairness, and operational planning [10–12]. Related work also considers student–supervisor allocation [13] and the generation, validation, and repair of medical-trainee rotation plans [33]. Across these closest educational and clinical-training scheduling studies, however, we did not identify the same explicit combination of role-based authorization, post-generation rule validation, auditable action provenance, reporting, and retained human approval as co-equal artifact requirements. The resulting gap is therefore architectural rather than algorithmic: it concerns the integration of scheduling support with explicit governance mechanisms within the same information-systems artifact.

3. System Architecture and Implementation

Beyond computational efficiency, the proposed architecture was designed to support structured decision-making, accountability, and controlled interaction between automated procedures and human oversight. The system translates the coordination requirements identified in the university psychology clinic into a governance-oriented decision-support

artifact capable of enforcing institutional rules while maintaining operational flexibility. The architecture combines scheduling functionality, validation mechanisms, access control, and traceability services within an integrated web-based environment.

3.1. Architectural Design Principles

Three architectural principles guided the system design: separation of concerns, embedded governance, and support for incremental evolution.

The architecture follows a layered structure that separates presentation, application services, and data persistence. Interface components, domain logic, and database models can evolve independently while minimizing structural coupling. Within the application layer, dedicated services encapsulate authentication, scheduling, validation, and reporting responsibilities.

Governance capabilities are incorporated directly into the system architecture. Role-based access control constrains authorized actions, audit logging records attributable events and state changes, and a post-generation validation step checks generated schedules against institutional rules. Together, these mechanisms make authorization, validation, and scheduling-state changes explicit and traceable within the workflow. Within this architecture, schedule generation, validation, authorization, logging, reporting, and human approval occupy distinct workflow functions, defined conceptually in Section 2.3.

The architecture also supports incremental evolution and deployment portability. Its modular organization enables the integration of additional capabilities, including clinical documentation modules and alternative scheduling approaches, without requiring substantial redesign. The web-based implementation further facilitates deployment across different institutional infrastructures, including on-premises and cloud environments.

3.2. Architectural Overview

Figure 2 presents the system architecture within its clinical-academic environment. The architecture operationalizes the requirements identified in Section 2 by connecting stakeholders, application services, and persistent data resources within a web-based decision-support environment.

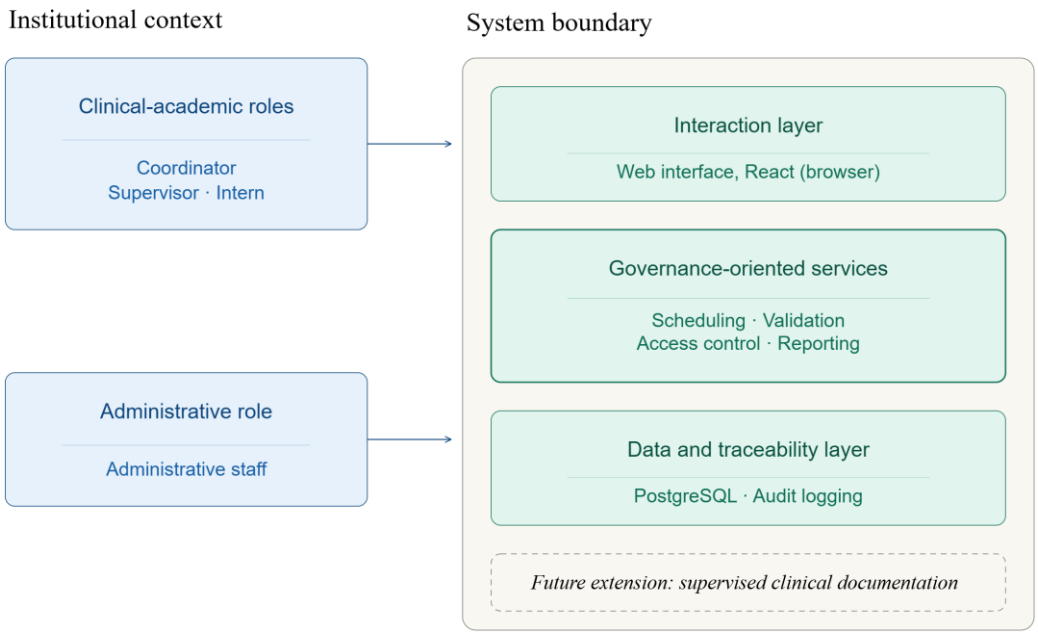


Figure 2. Institutional context and system boundary of the governance-oriented clinical scheduling system.

Beyond schedule generation, the system incorporates rule enforcement mechanisms, access-control and audit mechanisms, and extensibility provisions that support future integration of clinical records and additional decision-support capabilities.

The architecture follows a layered design composed of presentation, application, and persistence layers. Layered and modular architectures have been associated with sustainable digital transformation because they enable incremental capability development while preserving architectural stability [3,29].

The system is accessible through standard web browsers on desktop and mobile devices without requiring local installation. This approach increases accessibility while reducing deployment and maintenance effort.

The architecture comprises three primary layers:

- Presentation Layer (React frontend);
- Application Layer (FastAPI backend services);
- Data Layer (PostgreSQL database).

The interaction among stakeholders, application services, and persistent data resources is illustrated in Figure 2.

3.3. Application Layer and Core Services

Within the application layer, the main services translate institutional requirements into executable system operations, including authentication, schedule generation, validation, reporting, and audit-supported coordination. These services constitute the operational core of the proposed decision-support architecture, connecting user actions in the presentation layer to the domain logic and persistent data resources.

Figure 3 presents the system architecture and illustrates the interaction between the presentation layer, application services, and data persistence components, highlighting the central role of the scheduling service.

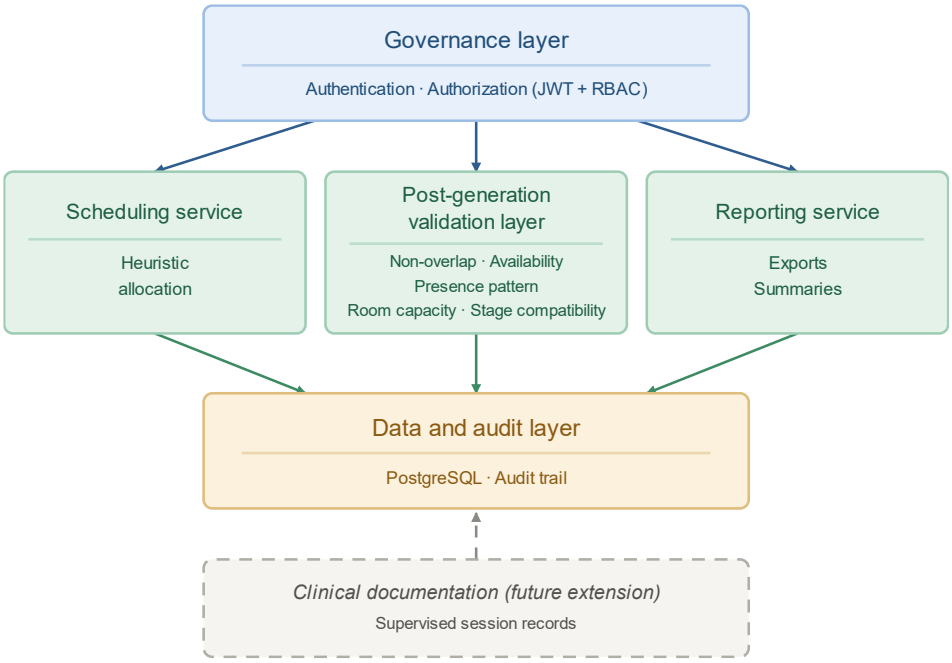


Figure 3. Modular backend architecture showing the governance layer, core services, and data persistence.

Authentication and Access Control Service. This service manages authentication and authorization through role-based access control (RBAC) supported by JSON Web Tokens

(JWT). Permission boundaries are defined according to organizational roles, including coordinators, supervisors, interns, and administrative staff, ensuring that access privileges remain consistent with institutional policies.

Scheduling Service. This service implements the allocation logic responsible for generating internship schedules under pedagogical and operational constraints. Constraint enforcement, consistency verification, and schedule generation are integrated within a single decision-support process. The underlying heuristic allocation procedure is described in detail in Section 3.4.

The computational scheduling logic is decoupled from the governance and reporting components. Consequently, alternative optimization or scheduling approaches can be incorporated in future versions without altering the surrounding workflow interfaces.

Post-generation validation layer. Immediately after the scheduling service generates a candidate schedule, this in-process step checks the resulting assignments against institutional rules and operational requirements. It executes within the same application process as schedule generation, rather than as an independent service, an external validator, or a separate service boundary, and never feeds back into or modifies the generated solution; a validation failure is surfaced as an explicit, blocking outcome rather than a silent correction.

Reporting Service. This service provides structured access to scheduling information for operational monitoring and managerial decision-making. Users can visualize schedules, identify conflicts, analyze allocation outcomes, and support coordination activities through a unified reporting interface.

Figure 4 complements the architectural view by illustrating the scheduling workflow, including heuristic allocation, automated validation, human approval, and audit logging.

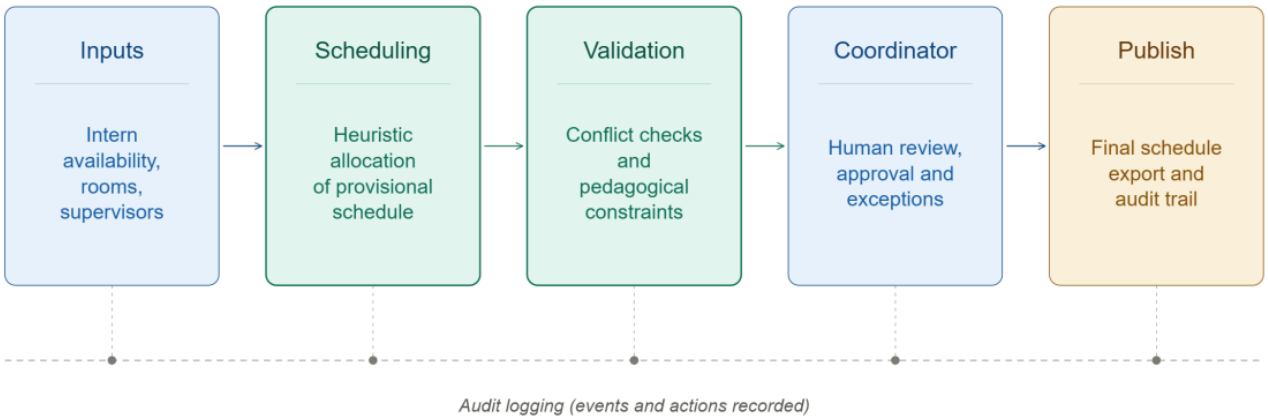


Figure 4. Governance-oriented scheduling workflow integrating heuristic allocation, automated validation, human approval, and audit logging. The dotted connectors and circular markers indicate that events and actions at each workflow stage are recorded in the audit log.

Operationally, these services compose a single governed sequence. An authenticated actor with the required permission requests schedule generation; role-based access control gates whether that specific action is authorized. The scheduling service (Section 3.4) then generates a candidate schedule without receiving any role, permission, audit, or approval state. Immediately afterward, the post-generation validation layer checks the resulting assignments (currently rechecking student non-overlap, availability, the required presence pattern, room capacity, and stage compatibility) without feeding back into or modifying the generated solution. The outcome persists as a schedule run carrying its validation state; a failed validation blocks approval and publication. A coordinator may subsequently approve or reject a validated run and publish an approved one, and every state-changing step is recorded as a transactional audit event with actor, timestamp, action,

and entity provenance. Governance mechanisms therefore surround the GRASP procedure operationally; they do not alter its construction, search, or scoring logic.

3.4. Scheduling Service and Heuristic Allocation Strategy

The Scheduling Service is responsible for generating feasible internship schedules under multiple pedagogical and operational constraints. Exact optimization of the full institutional instance, tested under a fixed computational budget (Appendix A), reached a partial-coverage incumbent within that budget; optimal coverage for this instance was instead established separately by mapping a complete GRASP-generated schedule into the exact formulation. Consequently, the proposed system adopts a GRASP-based heuristic as its core allocation mechanism.

The allocation procedure follows the GRASP metaheuristic framework [34], combining a randomized greedy construction phase with a single-pass triadic repair phase. Difficulty-based prioritization is inspired by the Minimum Remaining Values (MRV) principle commonly employed in constraint satisfaction problems [35], favoring the allocation of students with fewer feasible scheduling alternatives.

The heuristic combines three complementary mechanisms: difficulty-based prioritization of pending requirements, fitness-proportionate probabilistic selection among feasible candidate pairings, and a single-pass triadic repair procedure that introduces triadic supervision cycles when such transformations reduce the number of unassigned requirements.

Rather than pursuing global optimality through exhaustive search, the method prioritizes the construction of feasible schedules while preserving flexibility for subsequent improvements. This strategy is consistent with the operational objectives of the clinic, where complete and valid schedules are more valuable than computationally expensive attempts to identify provably optimal solutions.

Algorithm 1 summarizes the proposed procedure. During the construction phase, pending internship requirements are processed according to their estimated scheduling difficulty. For each requirement, feasible partner sets are identified, and candidate allocations are selected through roulette-wheel sampling. When binary allocations cannot satisfy all remaining requirements, the algorithm attempts the construction of feasible triadic supervision cycles.

The secondary term N_{sess} counts total session records. Across all 60 schedules evaluated in Section 4.5, $N_{sess} = 2(R - U)$, where R is the total number of requirements, so this term was invariant to binary-versus-triadic composition and did not affect incumbent selection in the reported experiments. The GRASP comparator therefore remains distinct from the exact formulation's secondary objective of minimizing triadic cycles.

After the randomized construction phase, the single-pass triadic repair procedure revises partially unresolved allocations and applies targeted binary-to-triadic transformations when they improve schedule completeness without violating institutional constraints. The heuristic therefore follows a feasibility-first strategy in which constraint satisfaction is enforced during construction and reinforced during repair.

Algorithm 1. GRASP heuristic with single-pass triadic repair used in the scheduling engine

Input:
A: set of students
E(a): pending internship stages for student a
S: set of rooms
H: set of time slots
T: set of scheduling days
K: number of GRASP iterations
Output:
Best feasible schedule found and remaining unassigned requirements

best_solution ← null
best_score ← (+infinity, +infinity)
For iteration k = 1,. . .,K:
Initialize empty schedule X and pending requirement set R
Phase 1—Randomized greedy construction:
For each student a ordered by difficulty:
For each pending stage e in E(a):
If pending count in e is odd OR no binary partner is feasible:
Attempt triadic construction within stage e
Otherwise:
Compute Ω(a,e), the feasible binary partner set
For each candidate b in Ω(a,e), compute difficulty d(b)
Select b* by roulette-wheel probability proportional to d(b)
Allocate reciprocal binary pair a → b* and b* → a
Attempt to allocate the required individual sessions for a and b*
Update availability, room occupancy, and pending requirements
Phase 2—single-pass triadic repair (only if enabled):
FLet U0 be the requirements still unassigned in X at this point
FFor each requirement (a,e) in U0:
For each existing binary pair (P,Q) in stage e:
Attempt to transform (P,Q) into triad (P,Q,a)
If successful: attempt to allocate the required individual session for a, mark a as allocated, and stop searching partners for this requirement
Score and update best_solution (once per complete iteration, after both phases):
U ← number of unassigned requirements in X; N_sess ← number of session records in X
score(X) ← (U, N_sess)
If score(X) < best_score, compared lexicographically (U first, then N_sess; an exact tie leaves best_solution unchanged):
best_solution ← X; best_score ← score(X)
If best_solution has no unassigned requirements: stop early
Return best_solution and remaining unassigned requirements

3.5. Data Architecture and Persistence Model

The data layer is implemented in PostgreSQL, selected for its transactional reliability, support for referential integrity, and compatibility with institutional information technology environments. Within the layered architecture described in Section 3.1, the persistence layer stores operational scheduling data and supports auditability across coordination activities.

The schema reflects the clinical scheduling domain and includes entities associated with users and roles, interns, supervisors, patients, rooms, availability records, appointments, and audit events. Referential constraints preserve consistency across actors, locations, and supervision relationships, providing a reliable foundation for schedule generation, validation, and reporting services.

Audit logging is incorporated as a core persistence-layer capability. Schedule generation events, data modifications, and administrative actions are recorded together with timestamps and user identifiers, supporting attributable provenance and historical traceability.

Beyond its operational role, the database also preserves historical scheduling information across academic cycles. This repository enables retrospective analysis, supports administrative oversight, and provides a structured record of coordination activities over time.

3.6. Deployment Strategy and System Evolution

The proposed architecture supports deployment on institutional on-premise servers as well as cloud-based virtual private servers, allowing adaptation to different information technology policies and infrastructure constraints. Containerization technologies such as Docker can be employed to improve reproducibility and maintain consistency across development, testing, and production environments.

The current web-based system originated from an earlier desktop prototype developed to formalize scheduling rules and evaluate operational feasibility during the initial stages of the project. The prototype supported the assessment of supervision requirements and contributed to the identification of key coordination constraints through stakeholder interaction.

Migration to the web-based architecture introduced centralized data management and multi-device accessibility while preserving the validated scheduling logic developed during the prototype phase. Role-based access control, persistent audit logging, and the controlled approval/publication workflow described in Section 3.3 were completed and technically verified in a subsequent refinement phase, after the pilot deployment reported in Section 4.4, extending rather than altering the scheduling and post-generation validation exercised during that pilot.

This development path enabled the gradual incorporation of new functionalities while reducing implementation risks and facilitating organizational adoption [3,29]. The resulting architecture provides a stable foundation for future extensions without requiring substantial structural modifications.

Figure 5 summarizes the evolution of the system from a scheduling-focused prototype to a broader clinical-academic information system.

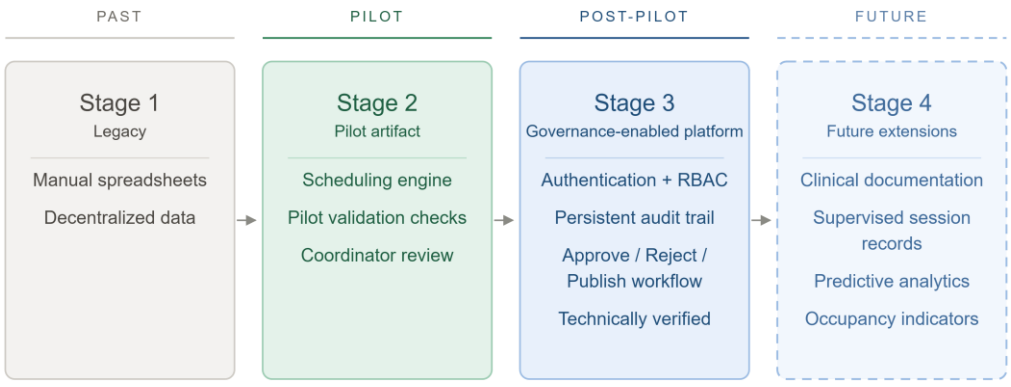


Figure 5. Evolutionary roadmap of the clinical-academic information system.

Future developments may include integration of supervised clinical documentation, analytical modules, and additional decision-support capabilities, extending system functionality while preserving the architectural principles described in this study.

4. Research Methodology and Artifact Development

4.1. Design Science Research Approach

This study adopts a DSR approach to design, implement, and evaluate an information system intended to address the coordination challenges identified in the university psychology clinic. DSR is particularly appropriate in information systems research when knowledge contributions emerge through the development and evaluation of artifacts designed to solve relevant organizational problems [36,37].

The artifact developed in this study is a governance-oriented decision-support system that supports internship scheduling, supervision coordination, and rule enforcement within a clinical-academic environment. Its design integrates scheduling functionality, access control, validation mechanisms, and audit capabilities within a unified architecture.

The research process followed an iterative design cycle comprising problem identification, requirements elicitation, artifact development, demonstration in an organizational setting, and evaluation. Feedback obtained from institutional stakeholders informed successive refinement cycles, contributing to both the system architecture and the governance mechanisms incorporated into the final solution.

Figure 6 summarizes the DSR process adopted in the study and illustrates the relationship between organizational context, artifact development activities, evaluation procedures, and resulting knowledge contributions.

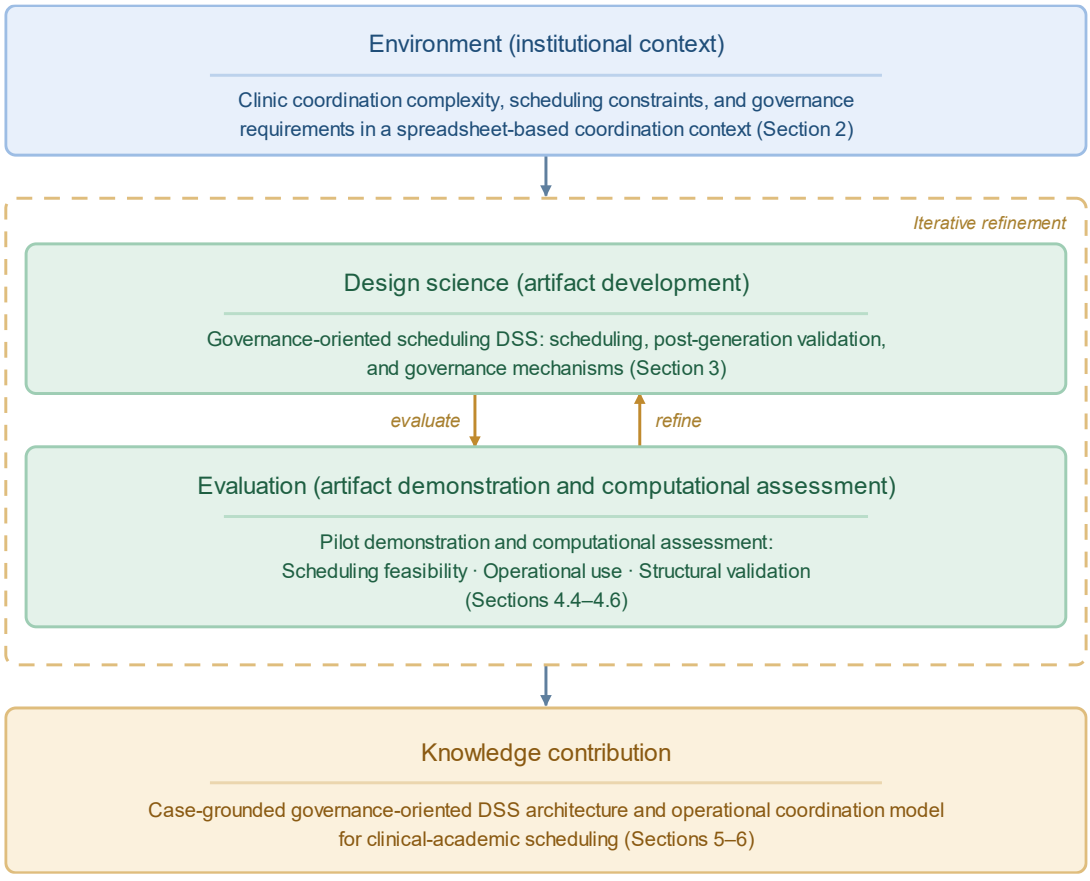


Figure 6. DSR logic applied in the study: iterative relationship between institutional context, artifact development, and evaluation.

4.2. Stakeholder Analysis and Requirements Elicitation

Requirements elicitation was conducted through four consultation meetings during the first academic semester of 2025, involving 21 unique participants across the four meetings: 2 coordinators, 14 clinical supervisors, 4 interns, and 1 administrative staff member. Each meeting lasted approximately 50 min and focused on existing scheduling practices, recurring allocation conflicts, supervision rules, resource constraints, and expectations regarding digital coordination support. The information discussed during the meetings was documented by the clinic coordinators in reports shared among the participating teams. After each consultation cycle, the identified rules and operational needs were consolidated into a working requirements list. Overlapping statements were merged, while divergences were addressed during supervision sessions and meetings with the coordinators, with decisions guided by institutional regulations and established procedures. The consolidated requirements were then classified according to their primary design purpose into three clusters: supervision-structure formalization, automated conflict detection, and transparent traceability, which are summarized in Table 1 and subsequently informed artifact design. Requirements elicitation was considered sufficiently stable for artifact development when subsequent discussions primarily refined previously identified requirements rather than introducing new operational rules or recurrent scheduling constraints. These activities constituted an engineering requirements process intended to support artifact design and refinement rather than a formal qualitative study.

Table 1. Stakeholder-derived requirement clusters informing artifact design.

Requirement Cluster	Description	Stakeholder Groups
Formalization of peer-observation structures	Formalization of peer-observation relationships within internship allocation	Supervisors, Coordination
Automated conflict detection	Prevention of double-booking and resource over-allocation	Coordination, Administration
Transparent traceability	Auditability and visibility of rule enforcement	All groups

The elicitation process revealed three dominant requirement clusters, summarized in Table 1. The first concerned formalization of supervision structures and enforcement of scheduling rules. The second focused on automated detection of resource conflicts and operational inconsistencies. The third addressed transparency, accountability, and visibility of coordination decisions.

These requirements directly influenced the design of the proposed artifact. Scheduling and the post-generation validation layer were developed to address supervision and conflict-management requirements, while audit logging, access control, and reporting capabilities were incorporated to support traceability, controlled access, and inspection of coordination actions.

4.3. Iterative Artifact Development

Artifact development began with a desktop-based prototype designed to test whether the clinic's pedagogical and operational constraints could be represented computationally. This stage enabled the implementation and validation of supervision structures, scheduling rules, and conflict-detection procedures while providing early feedback from coordination staff regarding workflow requirements.

Subsequent iterations resulted in the transition to a web-based architecture implemented with FastAPI, PostgreSQL, and React, as described in Section 3. Beyond improving accessibility and centralized data management, this transition incorporated the scheduling

service, the post-generation validation layer, and integrated reporting capabilities described in Section 3.3. Role-based access control, persistent audit logging, and the controlled approval/publication workflow were completed and technically verified in a subsequent refinement phase, after the pilot deployment described in Section 4.4. These additions extended the governance workflow without altering the GRASP scheduling logic, while the post-generation validation layer was also expanded during the refinement phase with additional room-capacity and stage-compatibility rechecks.

Throughout the development process, the scheduling logic was continuously refined while preserving compatibility with previously validated coordination rules. The resulting artifact combines scheduling support, governance mechanisms, and operational management functions within a unified clinical-academic information system.

4.4. Artifact Demonstration

Following development, the artifact was deployed during a pilot semester in a university psychology clinic. Deployment activities included user configuration, migration of operational data, user onboarding and support activities, and parallel use alongside existing spreadsheet-based procedures.

During the demonstration phase, coordinators employed the system to generate schedules, review the then-implemented validation outputs, and inspect allocation outcomes; coordinators reviewed generated allocations and decided whether they were suitable for operational use, consistent with the semi-structured nature of the coordination process. This review was not tracked through a software-enforced approval state; the formal approve/reject/publish workflow described in Section 3.3 was completed and technically verified subsequently, after this pilot.

Figure 7 presents a dashboard rendering for one representative execution (seed 1) from the evaluated institutional-instance configuration, illustrating allocation results and post-generation validation outputs. In this execution, all 150 requirements were allocated, producing 300 scheduled sessions with 63 binary and 8 triadic supervision cycles while all five post-generation validation checks were satisfied.

In addition to timetable visualization, the dashboard provides automated verification of scheduling constraints, including availability compliance, room-capacity validation, double-booking prevention, and supervision-structure consistency. Room-capacity and stage-compatibility rechecks were incorporated into the post-generation validation layer during the subsequent refinement phase described in Section 3.3, rather than during the pilot deployment itself. Operational indicators summarize allocation outcomes, including the number of scheduled sessions, unassigned requirements, room utilization, and the composition of binary and triadic supervision cycles.

The interface exemplifies how scheduling support and the post-generation validation layer were integrated within the artifact used during the pilot, providing coordinators with a unified view of operational activities. Role-based access control, audit logging, and the formal approval/publication workflow described in Section 3.3 were incorporated into the decision-support architecture subsequently.

Beyond the pilot demonstration, the subsequently completed governance workflow, comprising authentication, RBAC, workflow-state transitions, validation gating, audit persistence, and migration compatibility, was subjected to automated integration and regression testing, including schedule-output invariance checks confirming that governance wrapping did not alter the scheduling output of the GRASP procedure under fixed inputs and seeds. This testing establishes that the implemented mechanisms exist and operate as specified; it is a technical-verification exercise and not a stakeholder-outcome or governance-effectiveness evaluation.

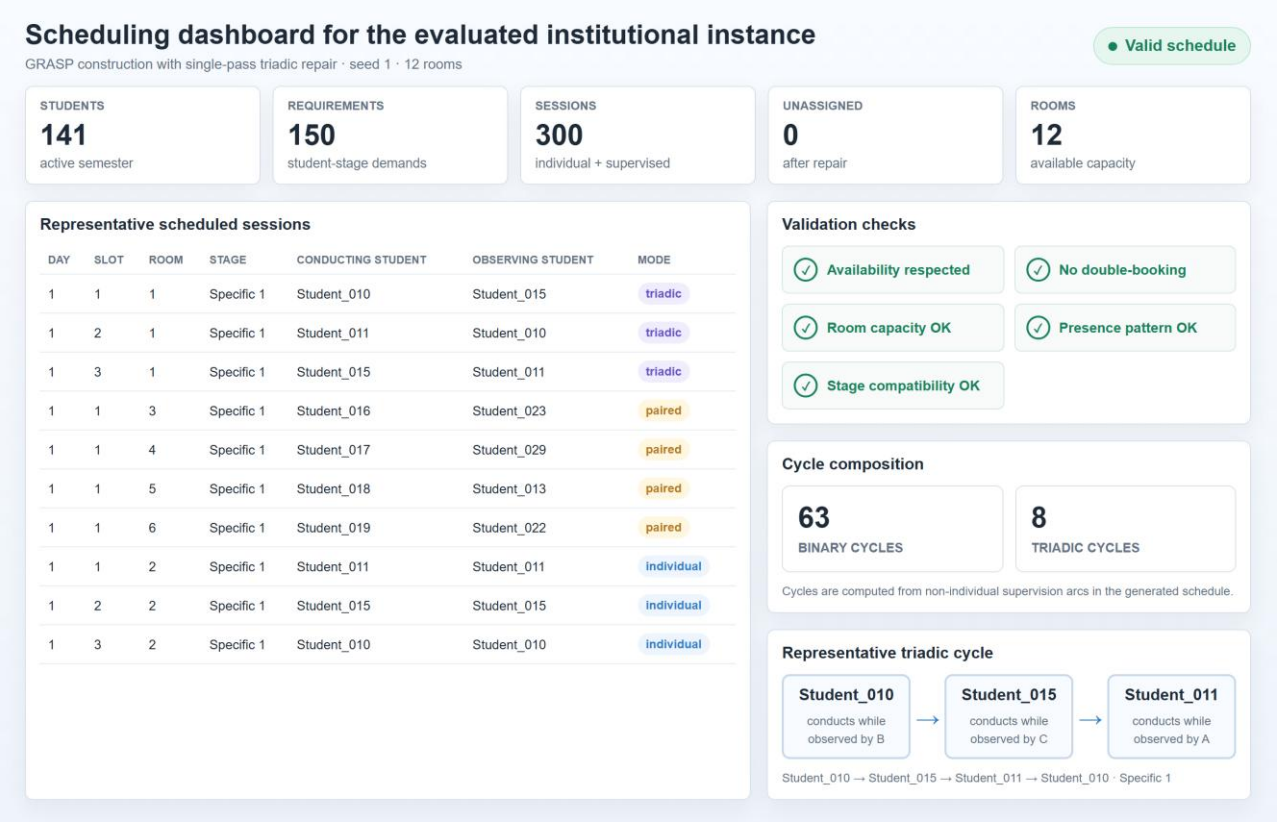


Figure 7. Scheduling dashboard for a representative execution of the evaluated institutional instance (seed 1), showing allocation results, representative scheduled sessions, supervision-cycle composition, and post-generation validation checks.

4.5. Computational Evaluation of the Scheduling Artifact

To complement the artifact demonstration and case-based operational assessment, a computational evaluation was conducted using a real scheduling instance previously managed through manual coordination procedures.

The evaluated instance contained 141 students, 12 clinical rooms, 13 time slots per day, 5 weekdays, 4 internship stages, and 150 student-stage requirements. The same institutional availability and stage-compatibility data were used in all computational experiments.

Two variants of the scheduling heuristic were evaluated. The first variant, construct_only, applies only the randomized greedy construction phase. The second variant, construct_plus_local_search, is the experimental identifier retained for the variant that applies the same construction procedure followed by the single-pass triadic repair described in Algorithm 1, which can transform binary supervision structures into triadic supervision cycles within the same internship stage.

The evaluation was conducted using 30 randomized executions of each heuristic variant. This design allows assessment of the contribution of the single-pass triadic repair while maintaining the operational characteristics of the institutional scheduling instance.

Table 2 reports the mean and standard deviation of the principal scheduling indicators across the 30 executions for each variant.

The single-pass triadic repair consistently improved schedule completeness across all executions. The mean number of satisfied requirements increased from 146.40 to 150.00, eliminating all unassigned requirements and producing complete schedules in every run.

This improvement was achieved through the introduction of additional triadic supervision cycles. The mean number of binary cycles decreased from 66.60 to 63.00, while

the mean number of triadic cycles increased from 4.40 to 8.00. No supervision structures involving more than three students were generated.

Table 2. Performance comparison of constructive and single-pass triadic-repair heuristic variants.

Metric	Construct_Only (Mean ± sd)	Construct_Plus_ Local_Search (Mean ± sd)	Delta
n_requirements_satisfied	146.40 ± 0.50	150.00 ± 0.00	+3.60
n_unassigned	3.60 ± 0.50	0.00 ± 0.00	−3.60
n_cycles_binary	66.60 ± 0.50	63.00 ± 0.00	−3.60
n_cycles_triadic	4.40 ± 0.50	8.00 ± 0.00	+3.60
occupancy_rate	0.3754 ± 0.0013	0.3846 ± 0.0000	+0.0092

All generated schedules were subsequently checked by a separately implemented validator against five structural feasibility rules: room capacity, student double-booking, student availability, stage compatibility, and the required individual/conducting/observing presence pattern for every requirement considered covered. No violations of these structural rules were detected in either variant across all 60 evaluated schedules. To challenge the separately implemented validator beyond acceptance of generated schedules, five rule-specific negative tests were constructed by deliberately injecting one violation of each checked structural category into otherwise valid schedules. The validator detected all five injected violations, including the intended rule category in every case, while the corresponding unmodified control schedules remained accepted. These tests exercise room capacity, student non-overlap, availability, stage compatibility, and the required presence pattern and complement the structural checks reported for the computational experiments. This structural check is distinct from schedule completeness: for the construct_only variant, the 3.60 mean unassigned requirements reported above are requirements the constructive phase left incomplete; they are recorded explicitly, with a stated reason, and are excluded from the structural-violation count by definition. The construct_plus_local_search variant left no requirement incomplete in any of the 30 evaluated runs.

Overall, the results indicate that the single-pass triadic repair improves schedule completeness while preserving all institutional constraints. These findings provide evidence of the practical utility of the proposed scheduling artifact within the evaluated clinical-academic environment.

To relate this heuristic evidence to the exact formulation in Appendix A, one complete construct_plus_local_search schedule (covering all 150 requirements with 63 binary and 8 triadic cycles) was mapped directly onto the exact model's decision variables and verified, constraint by constraint, to satisfy the formulation in Appendix A without modification. Full coverage was thereby separately established as optimal by combining this feasible mapped solution with the cardinal upper bound of 150 requirements: coverage cannot exceed the number of requirements, so exhibiting one feasible solution reaching 150 is sufficient. This conclusion does not rely on, and is independent of, the exact-solver experiment reported in Appendix A.9, which reports the corresponding CPLEX incumbent, best bound, and gap in detail.

4.6. Scalability and Exact-Comparison Evaluation

To assess the empirical behavior of the scheduling heuristic beyond the institutional instance and provide an exact reference on tractable cases, a synthetic instance generator was developed specifically for this artifact. The generator produces synthetic student populations by nonparametric bootstrap resampling, with replacement, of complete empirical availability/stage-set profiles drawn from the 141 real student profiles in the institutional

dataset used elsewhere in this study; this preserves the empirical joint structure between a student's availability and required internship stage(s) rather than sampling these attributes independently. Five sizes were evaluated (n = 50, 100, 150, 250, and 500 requirements), with the number of rooms scaled accordingly (4, 8, 12, 20, and 40 rooms), while the number of days (5), time slots per day (13), and internship stages (4) were held fixed. Ten independently generated instances were produced per size, and the GRASP heuristic reported in Section 3.4 was run on each instance with five algorithm seeds, without modification to the scheduler. Results are aggregated hierarchically: the five seed-level runs are first summarized within each generated instance, and size-level statistics are then computed across the ten resulting instance-level summaries.

High empirical coverage was maintained across the evaluated range, with mean coverage ratio ranging from 0.9580 at n = 50 to 0.9991 at n = 500 (Table 3). The evaluated synthetic instances with $n \geq 100$ contained no structurally isolated requirements under the candidate-generation rules used for the compatibility diagnostic, whereas the n = 50 instances contained a small number on average; this is reported descriptively for the tested instances and is not interpreted as a causal effect of problem size on compatibility or coverage.

Table 3. Scalability of the GRASP heuristic across the evaluated synthetic instance-size range.

n	Rooms	Instances	Seeds/ Instance	Mean Coverage Ratio (SD)	Range of Instance Means	Mean Runtime (ms)	Median Runtime (ms)	Range of Instance-Mean Runtime (ms)	Validation
50	4	10	5	0.9580 (0.0417)	0.860–1.000	83.50	92.86	4.89–165.43	100%
100	8	10	5	0.9890 (0.0114)	0.964–1.000	284.61	342.04	16.49–551.61	100%
150	12	10	5	0.9923 (0.0127)	0.963–1.000	551.68	105.39	43.53–2528.94	100%
250	20	10	5	0.9986 (0.0038)	0.987–1.000	878.44	389.46	168.69–3830.95	100%
500	40	10	5	0.9991 (0.0015)	0.996–1.000	7626.41	1430.95	893.45–25,905.53	100%

Statistics are hierarchical: seed-level runs (5 per instance) were first summarized within each generated instance, and the size-level statistics above were then computed across the resulting 10 instance-level summaries per size. Validation denotes the proportion of the 50 runs at each size (10 instances × 5 seeds) that passed the separately implemented structural validator.

GRASP computation time (timing the production solve_with_diagnostics() call directly, as described below) increased with problem size and showed substantial between-instance variability, particularly at larger sizes: at n = 500, instance-level mean runtime ranged from 893.45 ms to 25,905.53 ms. This runtime includes the full production solve—construction, single-pass triadic repair, and scoring—with early stopping upon reaching full coverage. Because early stopping is part of the production solver, the reported runtime represents the aggregate computational effort of construction, repair, scoring, and any repeated iterations before termination. Actual iteration counts were not recorded in the production campaign.

In one generated instance, two requirements were structurally isolated under the candidate rules, and GRASP consistently reached the resulting coverage optimum of 48/50 across all five seeds. In another instance with no structurally isolated requirements, GRASP consistently reached 43/50; this illustrates that local compatibility alone neither proves global feasibility nor guarantees that the heuristic will attain full coverage.

Separately from the synthetic scaling series above, a controlled runtime measurement was performed on the real institutional instance (141 students, 150 requirements, 12 rooms) reported in Section 4.5, using 30 independent algorithm seeds (1–30) of the same, unmodified GRASP heuristic. All 30 runs reached full coverage (150/150), passed structural validation, and produced 63 binary and 8 triadic cycles in every run. GRASP computation time (t_grasp_ms, timing solve_with_diagnostics() directly with time.perf_counter(), a monotonic wall-clock timer, around the production call only) averaged 69.03 ms (SD 34.41 ms; median 44.48 ms; range 41.48–179.93 ms). This was a controlled 30-seed measurement

conducted under the following computational environment: Windows 11; AMD Ryzen 7 5800H (8 physical/16 logical cores); 31.36 GB RAM; CPython 3.13.7. A maximum of 20 GRASP iterations was configured, with early stopping upon reaching full coverage.

The synthetic and institutional instances with similar nominal requirement counts should not be expected to exhibit identical runtime because search effort depends on the realized compatibility structure and on the number of iterations required before early stopping. Consistent with this interpretation, the synthetic $n = 150$ group showed substantial right-skewed runtime variability, with an instance-level range from 43.53 to 2528.94 ms despite a median runtime of 105.39 ms.

To provide an exact-solver reference for smaller tractable instances, the exact binary-cycle formulation presented in Appendix A was solved to proven optimality on two small institutional-subset instances distinct from the full 141-student instance analyzed above: an 8-student, single-stage instance and a 9-student, single-stage instance with triadic cycles enabled. The GRASP heuristic was then run on the identical instances (same students, availability, stage requirements, rooms, days, and slots) with 30 independent algorithm seeds each.

GRASP reached the CPLEX-proven coverage optimum in all 30/30 seeds for both instances (Table 4). Because the exact model additionally minimizes triadic-cycle count, whereas the GRASP comparator described in Section 3.4 does not, the comparison is based primarily on coverage; triad counts are descriptive, and scalar objective values are not compared directly between the two methods.

Table 4. Comparison between exact-optimal coverage and the GRASP heuristic on two small instances.

Instance	CPLEX-Proven Optimal Coverage	CPLEX Solver Time	GRASP Seeds Reaching Optimum	GRASP Mean Runtime
8-student, stage 1	8/8	0.582 s	30/30	0.424 ms
9-student, stage 1, triads enabled	9/9	0.392 s	30/30	0.570 ms

CPLEX values are solver-reported solve times and exclude model-generation overhead, whereas GRASP values time the production scheduling function directly. Runtime is therefore reported descriptively rather than as a controlled speed comparison.

4.7. Methodological Limitations

The evaluation reported in this study was conducted within a single university psychology clinic, which limits direct generalization of the findings to other institutional contexts. Consequently, the results should be interpreted as evidence of artifact utility in the studied environment rather than as statistical generalization.

In addition, the empirical assessment was performed during an initial pilot deployment period. The evaluation therefore focused on operational performance, schedule quality, and stakeholder use of the artifact within a specific coordination cycle. Longitudinal observations covering multiple semesters and broader organizational impacts were outside the scope of the present study.

The synthetic scalability instances (Section 4.6) use nonparametric bootstrap resampling with replacement from 141 empirical joint availability/stage profiles. This preserves the empirical profile structure while permitting larger problem sizes, but exact profile repetition becomes unavoidable and substantial when the synthetic population exceeds the source population; the larger synthetic instances should therefore not be interpreted as containing hundreds of independently unique empirical profiles.

The structural-isolation diagnostic used in the scalability evaluation (Section 4.6) identifies requirements that are locally impossible to cover under the candidate-generation rules, but the absence of such isolated requirements does not establish global feasibility and does not guarantee full heuristic coverage.

These limitations are inherent to the case-based evaluation strategy adopted in Design Science Research and should be considered when interpreting the reported results.

5. Discussion and Implications

5.1. Artifact Mechanisms and Operational Effects

Figure 8 distinguishes the organizational requirements shaping the artifact, the mechanisms implemented in the system, and the evidence reported for those mechanisms. The computational evaluation demonstrated full allocation for the institutional instance and no detected structural violations in the evaluated schedules, whereas the subsequently completed governance mechanisms were subjected to technical verification rather than organizational-outcome evaluation.

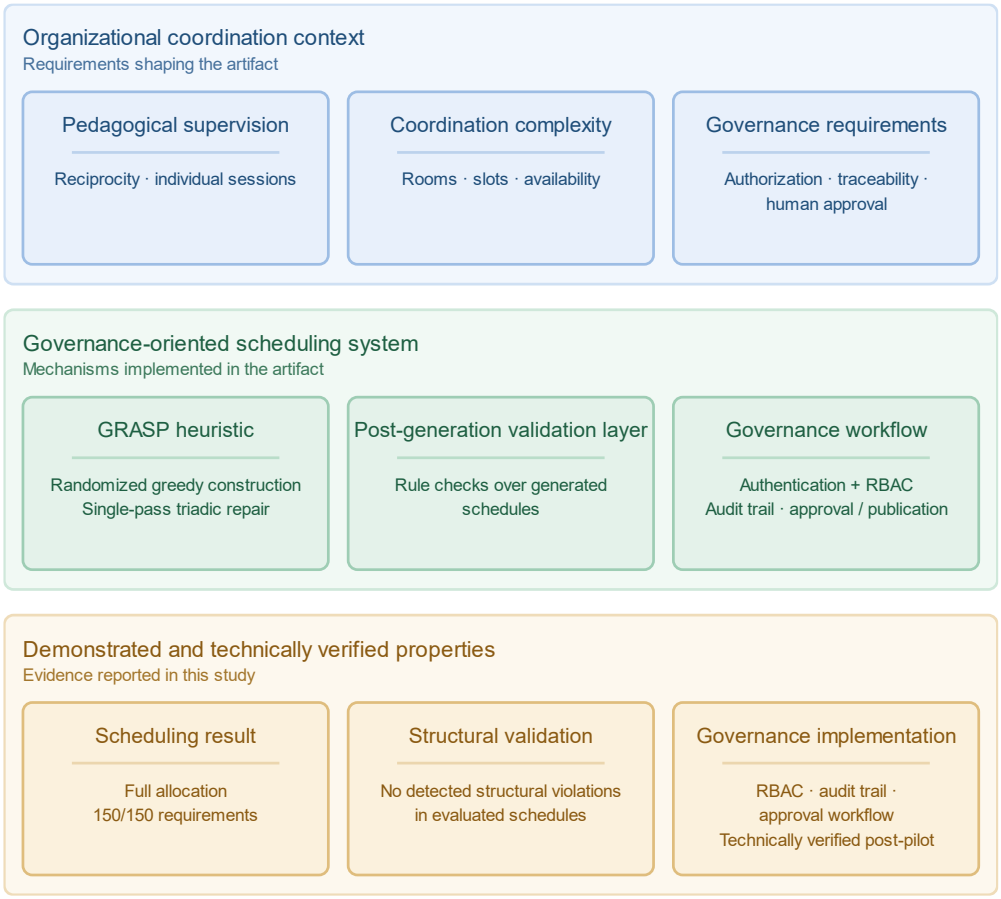


Figure 8. Relationship between organizational requirements, implemented scheduling and governance mechanisms, and the corresponding demonstrated or technically verified properties of the artifact.

In the proposed artifact, the scheduling service functions as the computational mechanism that translates institutional constraints into feasible allocation structures. Its role is complemented by the post-generation validation layer and governance mechanisms, which provide explicit rule checks and auditable workflow records around the computational outputs.

The post-generation validation layer systematically checks the represented institutional rules that were previously verified through manual procedures. Availability restrictions, room-capacity limits, supervision structures, and student non-overlap requirements are evaluated automatically, reducing dependence on manual cross-checking during schedule review.

Governance mechanisms, including role-based access control and audit logging, complement the scheduling and validation processes by providing mechanisms intended to support inspectability and attributable decision processes throughout coordination activities. Rather than functioning as external administrative controls, these mechanisms are integrated directly into operational workflows, making scheduling actions and validation outcomes more visible and traceable within the system.

Together, these mechanisms show that the artifact's contribution extends beyond heuristic scheduling to an integrated workflow in which allocation, post-generation checks, authorization, audit, and human approval are explicit. The computational and technical evidence reported here supports these implemented properties, while their broader organizational effects remain to be evaluated.

5.2. Governance and Organizational Implications

Beyond operational improvements, the study illustrates how coordination-intensive institutional processes can benefit from the integration of decision-support functionality and governance mechanisms within a single digital environment. The artifact transforms scheduling activities that previously depended on spreadsheet-based coordination and tacit organizational knowledge into a structured process supported by explicit rules, automated validation, and centralized information management.

The resulting coordination process remains consistent with the semi-structured nature of the organizational environment. Human oversight continues to play a central role in reviewing and approving scheduling decisions, while computational mechanisms provide systematic support for conflict detection, rule enforcement, and information consistency.

This combination structures the coordination process around explicit rules, automated checks, and retained human review and is intended to reduce risks associated with decentralized scheduling practices. It also makes allocation decisions, validation procedures, and administrative actions more visible and traceable within the system, providing mechanisms intended to support inspectability and attributable decision processes.

5.3. Implications for Information Systems and DSS Research

The artifact's governance orientation arises from the allocation of functions across the decision workflow rather than from the GRASP procedure itself. The scheduling service proposes allocations; the post-generation validation step checks them against institutional rules; RBAC constrains authorized actions; audit logging preserves attributable state transitions; reporting supports inspection; and authorized users retain approval and revision authority. These mechanisms are consistent with principles of meaningful human control and effective human oversight [14,15], while auditability supports inspection of automated decision processes and their procedural conditions [16]. However, the present case evaluation does not establish accountability or meaningful human oversight as organizational outcomes.

From a DSR perspective, the artifact can be understood within the design-science tradition of DSS research [17]. Its contribution is best characterized as a contextually grounded configuration and evaluation of established scheduling and governance mechanisms within a clinical-academic coordination artifact [25,26], rather than as the invention of a new optimization paradigm.

5.4. Limitations and Future Research Directions

The pilot deployment evaluated scheduling feasibility and operational use; it did not evaluate the subsequently completed governance mechanisms (role-based access control, audit logging, and the approval/publication workflow) as stakeholder-perceived outcomes. Transparency, accountability, acceptance, trust, and governance effectiveness were not

measured in this study. Prospective stakeholder-centered evaluation of these mechanisms, once they have been in operational use, remains future work.

Several opportunities exist for extending both the scheduling methodology and the information systems architecture proposed in this study.

From an algorithmic perspective, future research may investigate additional neighborhood structures, alternative repair operators, and preference-aware allocation mechanisms. While the current single-pass triadic repair procedure focuses on binary-to-triadic transformations within the same internship stage, broader search strategies may further improve schedule quality under secondary objectives such as preference satisfaction, workload balancing, or resource utilization.

The scheduling problem itself also presents opportunities for multi-objective optimization. Future studies may explore formulations that simultaneously consider schedule feasibility, minimization of room requirements, compactness of student schedules, satisfaction of availability preferences, and reduction of triadic supervision structures. In the current artifact, the exact reference formulation already incorporates triad count as a secondary objective term, while the GRASP heuristic's best-solution comparator does not; future work may incorporate explicit triad minimization or other secondary preference criteria directly into the heuristic solution-selection criterion, thereby aligning it more closely with the exact reference formulation.

From an information systems perspective, longitudinal evaluation across multiple internship cycles would provide deeper understanding of organizational learning effects, adoption dynamics, and sustained operational benefits. Multi-site deployments could further assess the transferability of the proposed design principles to other educational and healthcare environments.

Future architectural evolution may also include integration with academic management systems, electronic clinical record platforms, and advanced analytical modules. Such extensions would broaden the scope of the artifact while preserving the governance, validation, and traceability mechanisms described in the present study.

6. Conclusions

University psychology clinics represent coordination-intensive environments in which pedagogical requirements, supervision structures, operational constraints, and accountability demands must be reconciled within a single scheduling process. Traditional spreadsheet-based approaches provide flexibility but offer limited support for systematic validation, transparency, and traceability. This study addressed these challenges through the design and evaluation of a decision-support system for internship coordination.

Using a Design Science Research approach, the study developed and evaluated an information systems artifact that formalizes institutional coordination requirements as a rule-constrained scheduling problem and operationalizes them through a modular web-based architecture. The proposed solution integrates heuristic scheduling, automated validation, access-control mechanisms, and audit capabilities within a unified coordination environment.

The evaluation demonstrated that the artifact generated structurally valid, high-coverage schedules across the evaluated experiments and complete schedules for the institutional instance. As reported in Section 4.5, the incorporation of the single-pass triadic repair phase eliminated residual unassigned requirements in the evaluated instance, while the post-generation validation checks detected no violations of the represented pedagogical and operational constraints in the evaluated schedules.

The study's principal contribution is the design and initial evaluation of an integrated decision-support architecture for a specific clinical-academic coordination setting. Its contri-

bution does not depend on a novel scheduling algorithm; it lies in configuring scheduling, distinct rule validation, role-based authorization, auditable action histories, reporting, and retained human approval within one artifact. The scheduling procedure and the then-implemented post-generation validation checks were exercised operationally during the pilot deployment described in Section 4.4; role-based authorization, auditable action histories, and the retained human approval workflow were completed and technically verified in the subsequent refinement phase described in Section 3.3, rather than during that pilot. The case provides evidence of scheduling feasibility and operational utility, while broader claims concerning governance effectiveness, transferability, and meaningful human oversight require longitudinal, multi-institutional, and stakeholder-centered evaluation.

The study has limitations. Evaluation was conducted within a single institutional context during an initial deployment period, restricting direct generalization of the findings. Furthermore, the single-pass triadic repair procedure currently focuses on binary-to-triadic transformations within the same internship stage. Additional neighborhood structures, alternative repair operators, and broader benchmark validation remain opportunities for future research.

Future work should extend the evaluation across additional internship cycles and institutional settings while also exploring integration with academic management systems, electronic clinical records, and cost-aware resource-expansion scenarios.

By linking organizational coordination challenges to information systems design and demonstrating the practical utility of the resulting artifact, this study provides both a viable solution for university psychology clinics and a case-grounded framework that may inform the design of similar coordination artifacts in other institutional environments.

Author Contributions: Conceptualization, C.R.d.F. and J.R.D.L.; methodology, C.R.d.F. and J.R.D.L.; software, J.R.D.L.; validation, C.R.d.F., J.R.D.L. and H.P.; formal analysis, J.R.D.L.; investigation, C.R.d.F.; resources, C.R.d.F.; data curation, C.R.d.F.; writing—original draft preparation, C.R.d.F., J.R.D.L. and H.P.; writing—review and editing, C.R.d.F. and J.R.D.L.; visualization, J.R.D.L.; supervision, J.R.D.L.; project administration, C.R.d.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: This study did not involve the collection of personal, clinical, or sensitive data from human participants. Requirements elicitation was conducted through iterative consultations with institutional stakeholders involved in internship coordination, focusing exclusively on operational practices, scheduling procedures, and system requirements. These activities were performed as part of artifact development and pilot use of the artifact, and did not constitute formal human-subject research.

Data Availability Statement: The reproducibility package supporting this study, including the scheduling implementation, structural validator and validation challenge, synthetic-instance generation code, experiment configurations, governance technical-verification tests, and publicly releasable archived result files, is available at https://doi.org/10.5281/zenodo.22769604. The institutional dataset underlying the real-world instance derives from operational institutional records and has not been cleared for public release. Selected archived outputs and configuration metadata are provided to support verification of the reported results, while experiments based entirely on public-package inputs can be rerun directly.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Formal Model of the Cycle-Based Scheduling Formulation

This appendix presents the exact algebraic formulation implemented and solved in this study. The model is a cycle-based binary integer program in which candidate supervision

cycles, directed arcs, and physical session placements are enumerated in advance from institutional data, and the optimization selects among these pregenerated candidates.

Appendix A.1. Sets

A: set of students with at least one pending internship requirement.
R: set of internship requirements. Each $r \in R$ is a (student, internship stage) pair. For the institutional instance, $|R| = 150$.

$$C = C^{\text{bin}} \cup C^{\text{tri}}$$

G: set of candidate-directed supervision arcs. Each arc represents one conducts/observes relationship and belongs to exactly one candidate cycle. A binary cycle contains two directed arcs, whereas a triadic cycle contains three.

$$P = P^{\text{ind}} \cup P^{\text{arc}}$$

P: set of candidate physical session placements. P^{ind} contains placements for individual sessions, and P^{arc} contains placements realizing directed supervision arcs. Each placement specifies a concrete (day, time slot, room) combination.
K: set of room-time resource units, with k = (day, slot, room), used to enforce room capacity.
DT: set of student-time units, with dt = (day, slot), independent of room, used to enforce student non-overlap.

Appendix A.2. Incidence Structures

The following incidence relationships are fixed during candidate generation and are therefore input data to the optimization model rather than decision variables: reqCycle(r, c), arcCycle(g, c), indPlace(p, r), arcPlace(p, g), roomPlace(p, k), and personDaytimePlace(p, a, dt). The last relation indicates that student a participates in placement p at student-time unit dt, independently of the room in which the placement occurs.

Appendix A.3. Candidate-Generation Feasibility Rules

Availability and stage compatibility are enforced during candidate generation rather than through explicit inequalities in the optimization model. A candidate cycle is generated only for students who share the required internship stage. Its associated arcs and candidate placements are generated only for day-slot combinations in which all participating students are available. Consequently, infeasible cycles and placements are excluded before optimization and cannot be selected by the model. If no feasible candidate cycle is generated for a given requirement, that requirement cannot be covered, and its corresponding decision variable y_r is forced to zero through (C1).

Appendix A.4. Decision Variables

$y_{\text{r}} \in \{0, 1\}$—1 if requirement r is covered, and 0 otherwise.
$z_{\text{c}} \in \{0, 1\}$—1 if candidate cycle c is selected, and 0 otherwise.
$w_{\text{p}} \in \{0, 1\}$—1 if candidate placement p is used, and 0 otherwise.

Appendix A.5. Objective

$$\textbf{maximize } \boldsymbol{Z} = \boldsymbol{M} \sum_{\boldsymbol{r} \in \boldsymbol{R}} \boldsymbol{y_r} - \sum_{\boldsymbol{c} \in \boldsymbol{C}^{\text{tri}}} \boldsymbol{z_c}, \boldsymbol{M} = \textbf{1000} \tag{A1}$$

Appendix A.6. Constraints

Requirement–cycle linkage:

$$\sum_{c:\text{reqCycle}(r,c)} z_c = y_r \; \forall r \in R \tag{A2}$$

Individual-session linkage:

$$\sum_{p:\text{indPlace}(p,r)} w_p = y_r \; \forall r \in R \tag{A3}$$

Arc-realization linkage:

$$\sum_{p:\text{arcPlace}(p,g)} w_p = \sum_{c:\text{arcCycle}(g,c)} z_c \; \forall g \in G \tag{A4}$$

Room capacity:

$$\sum_{p:\text{roomPlace}(p,k)} w_p \leq 1 \; \forall k \in K \tag{A5}$$

Student non-overlap:

$$\sum_{p:\text{personDaytimePlace}(p,a,dt)} w_p \leq 1 \forall a \in A, \mid \; dt \in DT \tag{A6}$$

Domains:

$$\text{yr} \in \{0,1\} \; \forall \; \text{r} \in \text{R}, \; \text{zc} \in \{0,1\} \; \forall \; \text{c} \in \text{C}, \; \text{wp} \in \{0,1\} \; \forall \; \text{p} \in \text{P} \tag{A7}$$

Appendix A.7. Partial Coverage

Because the coverage variables y_r are not constrained to one, uncovered requirements can be represented explicitly. In particular, the assignment $y_r = 0$ for all $r \in R$, $z_c = 0$ for all $c \in C$, and $w_p = 0$ for all $p \in P$ satisfies Equations (A2)–(A7) and is always feasible for this formulation. Partial coverage is therefore represented directly through the decision variables rather than through model infeasibility.

Appendix A.8. Coverage-Priority Dominance

Let

$$N_{\text{cov}} = \sum_{r \in R} y_r$$

denote the number of covered requirements, and let

$$N_{\text{tri}} = \sum_{c \in C^{\text{tri}}} z_c$$

denote the number of selected triadic cycles. By Equations (A2), no two selected cycles can share a requirement. Because each triadic cycle covers three distinct requirements,

$$0 \leq N_{\text{tri}} \leq \left\lfloor \frac{\mid R \mid}{3} \right\rfloor = \left\lfloor \frac{150}{3} \right\rfloor = 50.$$

Consider any two feasible solutions such that

$$N_{\text{cov},1} > N_{\text{cov},2}.$$

Their objective-value difference is

$$Z_1 - Z_2 = M(N_{\text{cov},1} - N_{\text{cov},2}) - (N_{\text{tri},1} - N_{\text{tri},2}).$$

Since

$$N_{\text{cov},1} - N_{\text{cov},2} \geq 1$$

and

$$N_{\text{tri},1} - N_{\text{tri},2} \leq 50,$$

it follows that

$$Z_1 - Z_2 \geq M - 50 > 0$$

whenever $M > 50$. With $M = 1000$, a one-unit increase in coverage, therefore, dominates any possible change in the number of triadic cycles. The objective consequently implements a strict coverage-first priority, with triadic-cycle minimization acting only as a secondary criterion among solutions with equal coverage.

Appendix A.9. Computational Tractability

The formulation was solved with CPLEX 22.1.2 under a 600 s time limit (zero optimality-gap tolerance). CPLEX exported a feasible incumbent covering 116 of the 150 requirements (58 binary cycles, 0 triadic cycles; $Z = 116{,}000$). A separately implemented validator, following a distinct code path from the scheduling engine but based on the same institutional specification, confirmed structural validity by checking room capacity, student non-overlap, availability, stage compatibility, and presence pattern. The reported best bound was 149,999, one unit below the objective's theoretical ceiling $M|R| = 150{,}000$, giving an absolute gap of 33,999 and a solver-reported relative gap of 22.67%. CPLEX did not attain the coverage optimum of 150/150 within this 600 s budget; it exported a genuine, positive, 116/150 incumbent, not an empty or zero-coverage one.

The coverage optimum was established separately, not by this solve. A complete schedule produced by the GRASP heuristic (Section 3.4), covering all 150 requirements with 63 binary and 8 triadic cycles, was mapped, requirement by requirement, cycle by cycle, arc by arc, and placement by placement, into this formulation's y_r, z_c, and w_p variables and verified to satisfy Equations (A2)–(A7) exactly (Section 4.5). Full coverage was thereby separately established as optimal by combining this feasible mapped solution with the cardinal upper bound of 150 requirements. Coverage cannot exceed $|R| = 150$, and a feasible solution achieving $N_{\text{cov}} = 150$ was exhibited; therefore, $N^*_{\text{cov}} = 150$ follows directly, without any CPLEX optimality proof and independently of the 116/150 incumbent obtained in the 600 s CPLEX run.

This mapped solution has objective value $Z_{\text{GRASP}} = 1000 \times 150 - 8 = 149{,}992$, which provides a valid lower bound on the true optimum Z^*. Combined with the CPLEX best bound, this yields $149{,}992 \leq Z^* \leq 149{,}999$. Any solution with coverage $N'_{\text{cov}} \leq 149$ can attain an objective value of at most $1000 \times 149 = 149{,}000$, which is strictly below 149,992. Therefore, any optimal solution to the scalar objective must satisfy $N^*_{\text{cov}} = 150$, and hence $Z^* = 150{,}000 - N^*_{\text{tri}}$. Substituting the bounds on Z^* gives $1 \leq N^*_{\text{tri}} \leq 8$. Thus, at least one triadic cycle is mathematically necessary for full coverage of this instance, although the exact minimum number of triadic cycles remains unresolved by the evidence reported here.

These results establish optimal coverage for the institutional instance by direct construction rather than by a CPLEX optimality proof, while narrowing the admissible range for the optimal triad count without determining its exact value. They do not support a general claim that exact optimization is impractical for this problem class.

References

1. Bharadwaj, A.; El Sawy, O.A.; Pavlou, P.A.; Venkatraman, N.V. Digital business strategy: Toward a next generation of insights. *MIS Q.* **2013**, *37*, 471–482. [CrossRef]
2. Vial, G. Understanding digital transformation: A review and a research agenda. *J. Strateg. Inf. Syst.* **2019**, *28*, 118–144. [CrossRef]
3. Verhoef, P.C.; Broekhuizen, T.; Bart, Y.; Bhattacharya, A.; Dong, J.Q.; Fabian, N.; Haenlein, M. Digital transformation: A multidisciplinary reflection and research agenda. *J. Bus. Res.* **2021**, *122*, 889–901. [CrossRef]
4. Bond, M.; Marín, V.I.; Dolch, C.; Bedenlier, S.; Zawacki-Richter, O. Digital transformation in German higher education: Student and teacher perceptions and usage of digital media. *Int. J. Educ. Technol. High. Educ.* **2018**, *15*, 48. [CrossRef]
5. Benavides, L.M.C.; Tamayo Arias, J.A.; Arango Serna, M.D.; Branch Bedoya, J.W.; Burgos, D. Digital transformation in higher education institutions: A systematic literature review. *Sensors* **2020**, *20*, 3291. [CrossRef] [PubMed]
6. Power, D.J. Decision support systems: Concepts and resources for managers. *Stud. Inform. Control* **2002**, *11*, 349–350.
7. Arnott, D.; Pervan, G. Eight key issues for the decision support systems discipline. *Decis. Support Syst.* **2008**, *44*, 657–672. [CrossRef]
8. Ceschia, S.; Di Gaspero, L.; Schaerf, A. Educational timetabling: Problems, benchmarks, and state-of-the-art results. *Eur. J. Oper. Res.* **2023**, *308*, 1–18. [CrossRef]
9. Davison, M.; Kheiri, A.; Zografos, K.G. Modelling and solving the university course timetabling problem with hybrid teaching considerations. *J. Sched.* **2025**, *28*, 195–215. [CrossRef] [PubMed]
10. Akbarzadeh, B.; Wouters, J.; Sys, C.; Maenhout, B. The scheduling of medical students at Ghent University. *INFORMS J. Appl. Anal.* **2022**, *52*, 303–323. [CrossRef]
11. Zanazzo, E.; Ceschia, S.; Dovier, A.; Schaerf, A. Solving the medical student scheduling problem using simulated annealing. *J. Sched.* **2025**, *28*, 233–246. [CrossRef]
12. Shahraki, N.; Sir, M.Y.; Prindle, T.; Ramar, K. A decision-support system to schedule rotations for trainees. *ATS Sch.* **2022**, *3*, 425–432. [CrossRef] [PubMed]
13. Şimşek, A.B. A decision support tool for the student-supervisor allocation problem of postgraduate programs. *Expert Syst. Appl.* **2022**, *190*, 116068. [CrossRef]
14. Cornelissen, N.A.J.; van Eerdt, R.J.M.; Schraffenberger, H.K.; Haselager, W.F.G. Reflection machines: Increasing meaningful human control over decision support systems. *Ethics Inf. Technol.* **2022**, *24*, 19. [CrossRef]
15. Haselager, P.; Schraffenberger, H.; Thill, S.; Fischer, S.; Lanillos, P.; van de Groes, S.; van Hooff, M. Reflection machines: Supporting effective human oversight over medical decision support systems. *Camb. Q. Healthc. Ethics* **2024**, *33*, 380–389. [CrossRef] [PubMed]
16. Mökander, J.; Morley, J.; Taddeo, M.; Floridi, L. Ethics-based auditing of automated decision-making systems: Nature, scope, and limitations. *Sci. Eng. Ethics* **2021**, *27*, 44. [CrossRef] [PubMed]
17. Arnott, D.; Pervan, G. A critical analysis of decision support systems research revisited: The rise of design science. *J. Inf. Technol.* **2014**, *29*, 269–293. [CrossRef]
18. Sandhu, R.S.; Coyne, E.J.; Feinstein, H.L.; Youman, C.E. Role-based access control models. *Computer* **1996**, *29*, 38–47. [CrossRef]
19. Shrestha, Y.R.; Ben-Menahem, S.M.; Von Krogh, G. Organizational decision-making structures in the age of artificial intelligence. *Calif. Manag. Rev.* **2019**, *61*, 66–83. [CrossRef]
20. Kellogg, K.C.; Valentine, M.A.; Christin, A. Algorithms at work: The new contested terrain of control. *Acad. Manag. Ann.* **2020**, *14*, 366–410. [CrossRef]
21. Ala, A.; Chen, F. Appointment scheduling problem in complexity systems of the healthcare services: A comprehensive review. *J. Healthc. Eng.* **2022**, *2022*, 5819813. [CrossRef] [PubMed]
22. Munavalli, J.R.; Rao, S.V.; Srinivasan, A.; van Merode, G.G. An intelligent real-time scheduler for out-patient clinics: A multi-agent system model. *Health Inform. J.* **2020**, *26*, 2383–2406. [CrossRef] [PubMed]
23. Panko, R.R. Spreadsheets and Sarbanes-Oxley: Regulations, risks, and control frameworks. *Commun. Assoc. Inf. Syst.* **2006**, *17*, 29. [CrossRef]
24. Kankanhalli, A.; Charalabidis, Y.; Mellouli, S. IoT and AI for smart government: A research agenda. *Gov. Inf. Q.* **2019**, *36*, 304–309. [CrossRef]
25. Gregor, S.; Hevner, A.R. Positioning and presenting design science research for maximum impact. *MIS Q.* **2013**, *37*, 337–355. [CrossRef]
26. Tuunanen, T.; Winter, R.; vom Brocke, J. Dealing with complexity in design science research: A methodology using design echelons. *MIS Q.* **2024**, *48*, 427–458. [CrossRef]
27. Baskerville, R.; Pries-Heje, J.; Venable, J.R. MEDS: Methodology for evaluation in design science. *Eur. J. Inf. Syst.* **2026**, *advance online publication*. [CrossRef]
28. Gong, C.; Ribiere, V. Developing a unified definition of digital transformation. *Technovation* **2021**, *102*, 102217. [CrossRef]
29. Warner, K.S.R.; Wäger, M. Building dynamic capabilities for digital transformation: An ongoing process of strategic renewal. *Long Range Plan.* **2019**, *52*, 326–349. [CrossRef]
30. Elbashir, M.Z.; Collier, P.A.; Sutton, S.G. The role of organizational absorptive capacity in strategic use of business intelligence to support integrated management control systems. *Account. Rev.* **2011**, *86*, 155–184. [CrossRef]

31. Pinto, S.D.L.; Muniz, J., Jr.; Freitas, C.R.D.; Dale Luche, J.R. A Framework for the Innovation Management Capacity: Empirical Evidence from the Porto Digital Cluster in Brazil. *Adm. Sci.* **2025**, *15*, 191. [CrossRef]
32. Tallon, P.P.; Queiroz, M.; Coltman, T.; Sharma, R. Information technology and the search for organizational agility: A systematic review with future research possibilities. *J. Strateg. Inf. Syst.* **2019**, *28*, 218–237. [CrossRef]
33. Castaño, F.; Velasco, N. Exact and heuristic approaches for the automated design of medical trainees rotation schedules. *Omega* **2020**, *97*, 102107. [CrossRef]
34. Resende, M.G.; Ribeiro, C.C. Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In *Handbook of Metaheuristics*; Springer: Boston, MA, USA, 2010; pp. 283–319. [CrossRef]
35. Russell, S.J. *Artificial Intelligence a Modern Approach*; Pearson Education Inc.: Boston, MA, USA, 2020.
36. Hevner, A.R.; March, S.T.; Park, J.; Ram, S. Design science in information systems research. *MIS Q.* **2004**, *28*, 75–106. [CrossRef]
37. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A design science research methodology for information systems research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.