Project management information system data model development and explanation

Project Management Information System Data Model Development and Explanation

Filippo Maria Ottaviani[a], Massimo Rebuglio[b] and Alberto De Marco[c]

Department of Management and Production Engineering, Politecnico di Torino, Turin, 10129, Italy

fi

Keywords: Project Management, Information System, Data Model.

Abstract: The Project Management (PM) discipline is evolving towards the adoption of digital technologies, which are to be integrated into a project management information system (PMIS). Despite the fact that many PMIS solutions are already available, there is no standard data model for PMIS development, nor is the logic underlying PMIS. Therefore, organizations struggle to integrate other business applications into the PMIS and cannot leverage the data collected to improve both project management and execution. To address these issues, this paper aims to provide a standard PMIS model as a foundation for database design and software development. The PM objects are first identified and then represented in a data model that outlines their attributes and methods and the relationships between the classes. All classes are structured to accommodate in their interface the core PM processes, such as task and resource management, project scheduling, risk management, and progress control. The study evaluates the impact and benefits of implementing this standard model while acknowledging its limitations and providing recommendations for practical implementation.

1 INTRODUCTION

Project Management (PM) involves the use of techniques and tools to plan and execute a unique set of tasks, performed by resources, to attain specific objectives (BSI, 2012). PM has been increasingly digitized in recent years thanks to technological advancements and the spread of artificial intelligence models and frameworks. Different PM software for task management, status tracking, resource allocation, and online collaboration is now available to help streamline and automate PM processes. Therefore, it is now imperative for organizations to make use of information systems (ISs) for managing a complex project or, more generally, to improve any project's chances of success.

An information system (IS) is used to collect, store, process, and disseminate information to support decision-making and manage the flow of information within an organization. In the PM context, a Project Management Information System (PMIS) is designed to support various project processes. Its essential functions are represented by project cost and schedule planning and control. Furthermore, an IS would allow for improved risk management and simulations or the training of machine learning models for performance analyses (Ottaviani et al., 2022). Although, for a PMIS to be effective, it must be built upon a solid data model and be used in conjunction with other PM tools and techniques.

Developing an IS entails a number of essential steps, including data modeling and the definition of procedures. The former involves the creation of a visual representation of the data and relationships within a system, providing a comprehensive understanding of the system's structure and organization. This allows for more efficient decision-making and problem-solving, as the organization's components and interactions are more clearly understood. By depicting the flow of processes within the system, it is possible to identify and understand the relationships between different steps in a process by identifying and understanding the relationships between them. This can assist in ensuring that the process is as efficient as possible by identifying bottlenecks and areas for improvement. A further element of attention is the interface, i.e., how the software interacts with the user or other software.

Although the number of PMIS offerings has increased significantly, their data storage logic remains

[a] https://orcid.org/0000-0002-1150-9211
[b] https://orcid.org/0000-0002-2636-4537
[c] https://orcid.org/0000-0002-4145-2287

Ottaviani, F., Rebuglio, M. and De Marco, A.
Project Management Information System Data Model Development and Explanation.
DOI: 10.5220/0012052200003546
In *Proceedings of the 13th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2023)*, pages 210-217
ISBN: 978-989-758-668-2; ISSN: 2184-2841

unclear: no standard interfaces exist for extending software functionality or integrating with other business applications. In this regard, this study aims to develop a standardized data model and interfaces that can serve as a reference to develop a PMIS as it provides a generic and highly comprehensible model for PM. The data model can be used directly by the user or as a basic infrastructure for computational optimizations that can be interchanged between existing software programs. The study describes the PM objects, their attributes and relationships, and the procedures by which these objects are analyzed and modified. Particular emphasis is placed on the interfaces that facilitate data analysis, enhanced scheduling, risk optimization, and the underlying data and procedures that enable their functionality.

The paper is structured as follows. Section 1 introduces the PMIS and the different steps for its development and implementation. In Section 2, studies pertaining to PMIS are first grouped into different categories and then analyzed one by one. Section 3 consists of a detailed description of the proposed model and its content. Lastly, Section 4 concludes by addressing the limitations and presenting the conclusions for future research.

2 LITERATURE REVIEW

Scientific studies on Project Management Information Systems (PMIS) offer a comprehensive and varied range of information that organizations can utilize to enhance the design and implementation of PMIS and determine the most suitable PMIS for their specific requirements. These studies are crucial for organizations seeking to enhance their PM processes and attain successful project execution. Specifically, research on PMIS can be classified into three categories: modeling, evaluation, and comparison.

2.1 Modeling Studies

The first category of PMIS studies centres on providing different PMIS architectures. These studies utilize various modeling methods, such as data flow diagrams, entity-relationship diagrams, and object-oriented modeling, to craft a standard PMIS architecture that can be adapted by organizations of varying sizes and across multiple industries. Moreover, these studies aim to optimize the design and deployment of PMIS by determining the most efficient methods for structuring and organizing the data. In addition, they offer a standardized framework for PMIS development.

Studies by (Raimond, 1987) and (Björk, 1992) evaluated the viability of using conceptual data modeling to depict the information system within the PM system. This involved modeling the structure of the information that describes a project's products, processes, resources, and other elements. The two studies contrasted conceptual modeling with traditional methods and emphasized the potential for integrating PMIS with other ISs within the organization. On the other hand, (Froese, 1992) introduced the idea of standard models, consisting of a data model to represent the information, a domain model to express project concepts, and a project model that stores the project data and domain models.

Other studies have discussed the importance of integrating the PMIS with other applications used by different business functions. (Schoultz von et al., 1996) introduced the concept of integrated PMIS that provides support to a set of PM processes and ensures uniform data access. Likewise, (Jaafari and Manivong, 1998) emphasized the requirement for a centralized approach to information management. In contrast, (Garcia et al., 2016) proposed a meta-model for managing a single project that is based on a conceptual architecture that can be extended to the enterprise level.

Several studies have aimed at developing reference models for PMIS that cater to the key functions of PM, such as cost, time, scope, and quality management. Both (Karim and Adeli, 1999) and (Fadillah and Fitriana, 2019) utilized object-oriented programming to provide a PM information model that relates to different PM classes. On the other hand, (Yeganegi and Safaeian, 2012) stressed the importance of mapping the influence of stakeholders in the PMIS. Meanwhile, (Bashashin et al., 2016) documented the information related to the project monitoring and control process.

A number of studies have been carried out in developing ISs for project management, albeit with slight variations in their objectives. (Ahlemann, 2009) introduced a reference information model to expedite the setup of project ISs. (Li et al., 2015) developed a portfolio management IS based on complexity-based management methods to decompose information processing complexity. In (Teixeira et al., 2016), the PMIS was depicted using the UML class diagram graphical technique. Conversely, (Waheed et al., 2019) created a meta-model to map project integration management, with an emphasis on data automation.

2.2 Evaluation Studies

The second category of PMIS research is centred on implementing a PMIS and its impact on project management and execution.

These studies analyze the benefits of PMISs, such as improved communication, increased efficiency, and enhanced collaboration. For example, (Amami et al., 1993) stressed the importance of communication between stakeholders in determining a project's success and how PM fits into the organization's strategy. (Jalal Karim, 2011) proposed a PMIS model and assessed how PMISs facilitate decision-making in each phase of the project life-cycle.

In examining the software quality, the information output quality, as well as the influence of the PMIS user on project success, (Kahura, 2013) and (Taniguchi and Onosato, 2018) concluded that the use of PMIS enabled the project to be successful while respecting the project constraints and meeting the project objectives at the same time. Similarly, (Nguyo, 2014), (Park et al., 2018), and (Nyandongo and Lubisi, 2019) demonstrated a strong and positive correlation between project success and PMIS quality, information quality, system user, and system. (Borštnar and Pucihar, 2014) demonstrated that PMIS implementation could be successful if the organization is aligned from human resources, financial management, and collaboration perspectives.

Other studies also explored the critical success factors of PMIS implementation, including user training, change management, and data security. (Caniëls and Bakens, 2012) and (Rahman et al., 2018) examined how the use of a PMIS is advantageous to project managers, while no adverse effects were observed due to project and information overload. (Braglia and Frosolini, 2014) remarked that any PMIS has key requisites that relate to the project scope, resource allocation (Corrigan et al., 2019), time management, deliverables, assignments, risk management, project monitoring (Fachrizal et al., 2020), and quality (Bielova et al., 2019).

These studies help organizations to understand the potential impact of PMIS on their project management processes and to identify the best practices for successful PMIS implementation. For instance, (Raymond and Bergeron, 2008) examined the impact of PMIS on both project management and project execution performance, which confirmed that PMIS adoption contributes to improving budget cost and time control. (Mccarty and Skibniewski, 2014) proposed a multi-dimensional framework for PMIS training initiatives and (Nguyen et al., 2016) proposed a success model of PMIS for ERP projects validated through a survey study with path analysis.

2.3 Comparison Studies

The third category of PMIS research focuses on comparing various PMIS solutions and determining the criteria for selecting the most suitable PMIS for a specific organization. These studies assess PMISs based on functionality, scalability, cost-effectiveness, and other relevant factors. The goal of these studies is to assist organizations in choosing the PMIS that best meets their specific needs.

(Liu et al., 2008) presented a practical approach to analyze PMIS requirements, taking into account acquisition rules and the requirement analysis process. (Berziša and Grabis, 2011) highlighted the importance of PM knowledge during a PMIS configuration and detailed the knowledge acquisition and utilization processes. Using a questionnaire, (Lee and Yu, 2012) developed and validated the ASP-PMIS success model; this study served as a foundation for positioning and comparing PMIS success research. Finally, studies by (Kostalova et al., 2015), (Bellah et al., 2018), and (van Besouw and Bond-Barnard, 2021) compared free and licensed PMISs and concluded that the choice depends on the project complexity and requirements, as free PMIS lack advanced features, while licensed PMISs have a steeper learning curve.

2.4 Summary

Analysis of the literature in the field of PMIS has identified a significant gap regarding the development of basic conceptual models. Specifically, there is a lack of a standardized management model that can be applied across different areas. This gap highlights the need for a literature-based approach to PMIS development that focuses on creating a universal standard for project management.

Moreover, studies evaluating commercial PMIS software have revealed another significant gap in the field. These solutions often come with no conceptual data model, which must be reverse-engineered to integrate them with other software within the organization. This lack of standardization makes it difficult for organizations to leverage the data collected by PMIS for improved project management and execution. Therefore, it is crucial to develop a standardized approach to PMIS development that includes a conceptual data model to enable better integration with other business applications. Such an approach would enhance the efficiency of project management and execution while also facilitating the analysis of project data.

3 MODEL DEVELOPMENT

3.1 Diagrams and Notations Used

UML class diagram is structural diagram that shows the static structure of a system, its classes, attributes, interfaces, and the relationships between them. A class diagram is one of the most widely used UML diagrams and is often used to design, document, and communicate the structure of an object-oriented system (Rumbaugh et al., 2004). Classes consist of objects that share a common structure and behavior. Attributes represent class properties. An interface consists of a set of methods or operations to be implemented by a class. The solid lines connecting classes represent the associations between them, and the numbers at their ends indicate their cardinality. The solid lines ending with a diamond shape indicate a composition, i.e., an association by which a child class cannot exist without the parent class. Arrows indicate an inheritance relationship between a generic class and its specializations. Instead, the dotted arrows indicate when an interface of one class makes use of data from another class. Doing so helps ensure that the PMIS accurately reflects the business requirements and facilitates effective PM by providing a clear and comprehensive understanding of the data being managed.

3.2 Classes and Interfaces

The PMIS data model, which represents the various data entities and their relationships within the PMIS, is illustrated using the UML diagram depicted in Figure 1.

The `Organization` class indicates the organization that uses the PMIS to manage its project portfolios (`Portfolio` class), programmes (`Programme` class), and projects (`Project` class). A project portfolio is a collection of programmes and individual projects that are performed to achieve the organization's goals. In turn, a programme comprises multiple interdependent projects that are coordinated to achieve a shared outcome through efficient resource management. Lastly, a project consists of multiple tasks (`Task` class) carried out using the organization resources (`Resource` class). The aforementioned classes have a single attribute related to a primary identifier (`ID`); each of them is connected with `mandatory one` cardinality to the hierarchically superior class, and with `many optional` cardinality to the hierarchically inferior class.

The `Project` class presents two interfaces: the `cost()` interface returns the project Budget at Completion (*BAC*), while the `time()` interface returns its Planned Duration (*PD*).

The `Risk` class and its specializations, `OrganizationRisk`, `ResourceRisk`, and `TaskRisk`, represent risks associated with the different modeled entities. The class is defined by an ID, a validity period (`start` and `end` attributes), the `probability` of occurrence during that time, and the `impact` on resource productivity or task requirements.

The resources available to the organization to carry out the tasks provided by the projects are represented using the `Resource` class. The `designation` attribute is used to distinguish between resources. The `type` attribute indicates whether the resource is of the `work` or `material` type. A `work` resource indicates any resource whose `cost` scales with the time the resource is used. Instead, the amount or units used determines the `material` resource cost. The `max` attribute defines the resource maximum work time per day (`Work`) or units available (`Material`); while the association with the `ResourceRisk` class identifies possible risks associated with the resource.

Resources are managed and coordinated at the programme level through the `ProgrammeAllocation` association class, where the `max` attribute indicates the amount of resources.

The `Task` class presents the overhead costs attribute (`OH`) to account for any indirect cost, expressed in monetary units per unit of time. A task can have none or multiple predecessors or successors; class `Task` is therefore associated with itself with `optional many - optional many` cardinality (`Precedes` association). A task can also includes one or more subtasks, and it can have no more than one parent; class `Task` is therefore associated with itself with `optional many - optional one` cardinality (`Includes` association). Since a child task has no meaning without its parent task, the `Includes` association is a composition. The model allows for tasks with no associated requirements in order to represent tasks that have the sole purpose of grouping other tasks in a parent-child relationship. The class `Task` is associated with the class `Schedule` with `optional one` cardinality: each instance of Schedule represents the time information of a task instance, and it is possible that, transiently, a task may have no `Schedule` instances connected. The class `Task` is also associated with the class `Resources` by means of the association class `Requirement`, which represents the allocated resources with their respective quantities; and with the class `TaskRisk`, which describes the risks. In both cases, cardinality is `optional many`. Another association is with the `Progress` class, whose cardinality is `optional many`: each

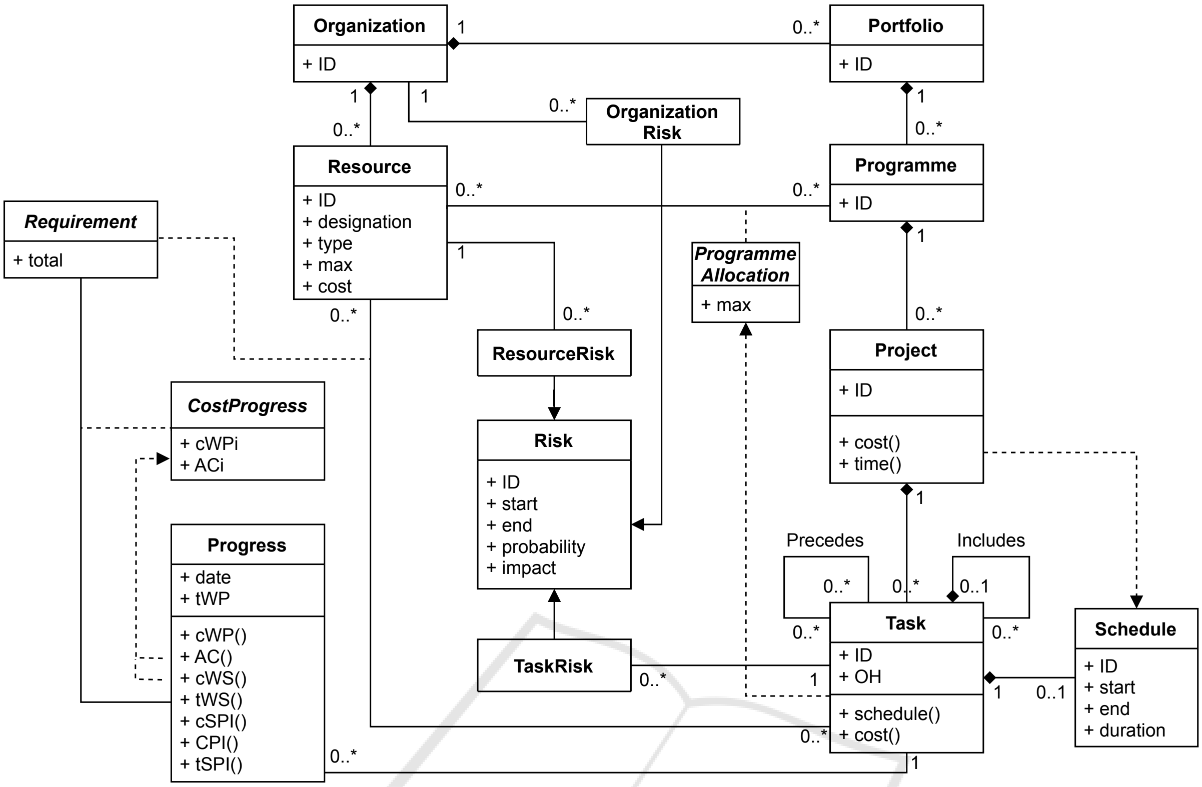

Figure 1: UML diagram showing classes, attributes and main interfaces of the model.

instance of `Progress` represents a temporal snapshot of the actual progress of the task. The `Task` class includes a `cost()` and a `schedule()` interfaces. The `cost()` interface returns the task *BAC* based on the resources allocated (`Requirement` class), the planned schedule (`Schedule` class), and the task overhead costs. The `schedule()` interface is used to schedule the task start, respecting the precedence and resource utilization constraints (`max` attribute of `ProgrammeAllocation` class).

The `Schedule` class is conceptually an extension of the `Task` class as it contains information about when and how many resources inherited from the `Requirement` class will be allocated to perform the work required by the task. Each schedule corresponds to exactly one task (`mandatory one` cardinality). The attributes `start`, `duration` and `end` indicate the expected start time, expected duration and expected end time of the work, respectively. The duration only represents the time required by the task in question, while the `end` attribute may be moved forward to wait for the end of a child task.

The `Progress` class provides snapshots of the progress of a task at a specific date. The task is always identifiable by virtue of the relationship with `mandatory one` cardinality with the class `Task`, while the date to which the snapshot refers is contained in the `date` attribute. The task scheduled progress is determined through the `tWS()` interface and compared with the progress performed, `tWP`. Instead, the task scheduled payments are determined through the `cWS()` interface and compared with the actual payments, `cWP()`, and the actual costs incurred, `AC()`. Interfaces are also provided to calculate performance indices: `cSPI()`, for time analysis based on the work done; `cSPI()`, for time analysis based on expenditure; `CPI()`, for cost analysis. Both `cWP()` and `AC()` are determined through interfaces as they association class `CostProgress`, which allows for a single input for each resource. For the model to work, an instance of `Progress` with progress indicators of zero must be entered at the task start date.

The association class `CostProgress` contains cost-based progress indicators, one for each resource associated with the task. Its `cWPi` and `ACi` attributes represent a disaggregated form of z and k, broken down by resource.

3.3 Features

The proposed model, as designed, allows three core PMIS functionalities to be implemented: scheduling, simulation, and project performance analysis.

Table 1: Aggregated logical view of interfaces and attributes aimed at cost and time control. Underlined data are attributes, non underlined data are interfaces. All attributes and functions are of the class `Progress` or its aggregation class `CostProgress`, except for `cost`, which is an interface of the class `Task`.

Performed (detail)		cWPi	ACi
Performed	tWP	cWP	AC
Total	tWS	cWS	Task.cost
Performance index	tSPI	cSPI	CPI

3.3.1 Scheduling

Project scheduling is the process of creating a detailed plan that outlines the sequence of activities, timelines, and resources required to complete a project. The proposed PMIS is designed to identify the critical path (CP) and estimate the project duration by establishing the Precedence Diagram Matrix (PDM) using the `precedes` association relationships between `Tasks`. The PDM is used by the PMIS to determine the different paths in the project, calculate their duration using the `schedule()` method, and identify the path with the longest duration as the CP. This capability is essential for accurately estimating the project duration during both the planning and execution phases. By identifying the CP, the PMIS enables project managers to focus their attention and resources on the most critical tasks, helping to ensure that the project is completed on time and within budget.

3.3.2 Simulation

The proposed PMIS data model can be used for simulation of project scheduling by incorporating risk factors into the model. As detailed in the preceding subsection, distinct project schedules are associated with differing expected risk values, given the defined start (`Risk.start`) and end (`Risk.end`) dates of each risk and the probability (`Risk.probability`) and impact (`Risk.impact`) values influenced by `Task` timing. Simulation of project execution takes into account these risks and their impact on `Tasks`, yielding varying schedules and potential modifications to project time and budget. By running simulations featuring diverse risk scenarios, the model generates a spectrum of potential project completion times (*PD*) and costs (*BAC*), thereby enabling a comprehensive analysis of project time and cost estimates. This approach facilitates identification of the most probable time and cost range for project completion through a joint analysis of project time and cost.

3.3.3 Performance Analysis

Project performance analysis involves evaluating the progress of project activities with regard to cost and schedule. For each task in the project schedule, the planned progress, payments, and budget spent are calculated. These values are determined using the `tWS()`, `cWS()`, and `cWS()` methods, respectively.

At each project review (denoted by `i`), actual time advances (`tWP()`), payments (`cWPi`), and costs (`ACi`) are provided as inputs for each task. This data allows the PMIS to calculate performance indices for scheduling (`tSPI()`), payments (`cSPI()`), and costs (`CPI()`). These indices can then be interpreted to evaluate performance, and can also be used to calculate estimates at completion for project time, payments, and costs.

4 CONCLUSIONS

The purpose of this study is to establish a standardized PMIS data model. To that end, a UML data model has been devised that illustrates the interconnections between tasks and resources and, thereby, project schedule and cost.

The paper begins with a description of the Project Management Information Systems (PMIS), a type of information system (IS) designed specifically to support project planning and cost and schedule control. To be effective, a PMIS must be based on a solid data model and must be used in conjunction with other project management tools and techniques. Therefore, the goal of the study is to provide a standardized data model and interfaces.

A literature review on PMIS is conducted, identifying three main categories of studies: modeling studies, which provide PMIS architectures, standard models, and reference models for PMIS development; evaluation studies, which assess the benefits of PMIS, including improved communication, efficiency, and collaboration, and explore the critical success factors of PMIS implementation; and comparison studies, which examine how to select the best PMIS for an organization.

A PMIS data model is developed, including interfaces for both task scheduling and control. These interfaces are present at the project level, with the possibility of expanding to the portfolio and program level as well. The model incorporates attributes for monitoring costs, overheads, risks and progress. Using a UML diagram, a high-level representation of the relationships between the different classes, their attributes, and interfaces is provided. The model provides a framework for cost control at several levels, including monitoring of actual costs incurred, comparison with planned costs and identification of deviations from the plan.

The developed model serves as the foundation for the database design and software development of a PMIS system, providing a logical framework that binds the various components of project management. Additionally, its transparency and simplicity make it compatible with integration into external management applications.

The proposed PMIS data model was specifically designed to facilitate the implementation of methods that can perform three critical functions of project management: scheduling, simulation, and performance analysis. These methods are essential in meeting time and cost targets, which are crucial aspects of project quality management. Unlike inherent project functionality and quality, which are often unique to a specific project type or stakeholder requirements, meeting time and cost targets are universal constraints that can be standardized. Our goal in developing the PMIS was to create a system that could effectively manage these universal constraints and enable project managers to achieve their time and cost targets across a broad range of projects. By implementing methods for scheduling, simulation, and performance analysis, the PMIS enables project managers to plan and manage projects more effectively, optimize resource utilization, and identify potential issues before they arise. The system's ability to standardize these critical functions also makes it easier to maintain consistency and ensure that best practices are followed across all projects.

The proposed model is immediately suitable to fulfill the main functions of a PMIS, such as activity management, resource allocation, scheduling and static risk management. However, for the implementation of advanced functions such as stakeholder management, dynamic risk management, prescriptive decision support tools and so on, modifications and extensions to the model may be necessary. In the current model's design, we took the implementation perspective into account, but did not describe interfaces to make the model talk to other software or an optimizer.

Future research directions will address the limitations of the model and seek to enhance it by adding methods and relationships to existing classes.

REFERENCES

Ahlemann, F. (2009). Towards a conceptual reference model for project management information systems. *International Journal of Project Management*, 27(1):19–30.

Amami, M., Beghini, G., and La Manna, M. (1993). Use of project-management information system for planning information-systems development projects. *International Journal of Project Management*, 11(1):21–28.

Bashashin, M. V., Kekelidze, D. V., Kostromin, S. A., Korenkov, V. V., Kuniaev, S. V., Morozov, V. V., Potrebenikov, Y. K., Trubnikov, G. V., and Philippov, A. V. (2016). NICA project management information system. *Physics of Particles and Nuclei Letters*, 13(5):618–620.

Bellah, J. C., Chen, L., and Zimmer, J. C. (2018). Development of a Project Management Software Tool: A Design Case. *International Journal of Designs for Learning*, 9(1):158–170.

Berziša, S. and Grabis, J. (2011). Knowledge reuse in configuration of project management information systems: A change management case study. *INES 2011 - 15th International Conference on Intelligent Engineering Systems, Proceedings*, pages 51–56.

Bielova, O. I., Hisham, S., and Elbaruni, J. E. (2019). Benefits of Integrating the Total Quality Management and Management Information System Into Project Management.

Björk, B.-C. (1992). A unified approach for modelling construction information. *Building and Environment*, 27(2):173–194. Special Issue Integrated Database and Data Models.

Borštnar, M. K. and Pucihar, A. (2014). Impacts of the Implementation of a Project Management Information System – a Case Study of a Small R&D Company. *Orga*, 47(1):14–23.

Braglia, M. and Frosolini, M. (2014). An integrated approach to implement Project Management Information Systems within the Extended Enterprise. *International Journal of Project Management*, 32(1):18–29.

BSI (2012). BS ISO 21500:2012 BSI Standards Publication Guidance on project management. Technical report, International Organization for Standardization.

Caniëls, M. C. and Bakens, R. J. (2012). The effects of Project Management Information Systems on decision making in a multi project environment. *International Journal of Project Management*, 30(2):162–175.

Corrigan, M. J., van der Poll, J. A., and Mtsweni, E. S. (2019). The Project Management Information System as Enabler for ICT4D Achievement at Capability Maturity Level 2 and Above. In Krauss, K., Turpin, M., and Naude, F., editors, *Communications in Computer and Information Science*, volume 933 of *Communications in Computer and Information Science*, pages 295–310. Springer International Publishing, Cham.

Fachrizal, M. R., Wibawa, J. C., and Afifah, Z. (2020). Web-Based Project Management Information System in Construction Projects. *IOP Conference Series: Materials Science and Engineering*, 879(1).

Fadillah, A. P. and Fitriana, D. (2019). Design of Project Data Management Information System. *IOP Conference Series: Materials Science and Engineering*, 662(2).

Froese, T. M. (1992). *Integrated Computer-Aided Project Management through Standard Object-Oriented Models*. PhD thesis, Stanford University, Stanford, CA, USA. UMI Order No. GAX92-34096.

Garcia, I., Pacheco, C., Arcilla-Cobián, M., and Calvo-Manzano, J. A. (2016). MyPMP: A plug-in for implementing the metamodeling approach for project management in small-sized software enterprises. *Computer Science and Information Systems*, 13(3):827–847.

Jaafari, A. and Manivong, K. (1998). Towards a smart project management information system. *International Journal of Project Management*, 16(4):249–265.

Jalal Karim, A. (2011). Project Management Information Systems (PMSI) Factors: An empirical Study of their impact on project management decision making (PMDM) Performance. *Research Journal of Economics, Business and ICT*, 2:22–27.

Kahura, M. N. (2013). The Role of Project Management Information Systems towards the Success of a Project: The Case of Construction Projects in Nairobi Kenya. *International Journal of Academic Research in Business and Social Sciences*, 3(9).

Karim, A. and Adeli, H. (1999). OO Information Model for Construction Project Management. *Journal of Construction Engineering and Management*, 125(5):361–367.

Kostalova, J., Tetrevova, L., and Svedik, J. (2015). Support of Project Management Methods by Project Management Information System. *Procedia - Social and Behavioral Sciences*, 210:96–104.

Lee, S. K. and Yu, J. H. (2012). Success model of project management information system in construction. *Automation in Construction*, 25:82–93.

Li, Y., Lu, Y., Kwak, Y. H., and Dong, S. (2015). Developing a city-level multi-project management information system for Chinese urbanization. *International Journal of Project Management*, 33(3):510–527.

Liu, W., Zhao, S., Sun, Y., and Yin, M. (2008). An approach to project management information system requirements analysis. *Proceedings - International Conference on Intelligent Computation Technology and Automation, ICICTA 2008*, 2:957–961.

Mccarty, A. J. and Skibniewski, M. J. (2014). Toward a Framework for Project Management Information Systems Training 1. *PM World Journal Toward a Framework for Project Management*, III(9):1–11.

Nguyen, T. D., Nguyen, D. T., and Nguyen, T. M. (2016). Information systems success: The project management information system for ERP projects. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 165:198–211.

Nguyo, N. R. (2014). Influence of Arbitration on Dispute Resolution in the Construction Industry : a Case of Nairobi County , Kenya By. Master's thesis, University Of Nairobi.

Nyandongo, K. M. and Lubisi, J. (2019). Assessing the use of project management information systems and its impact on project outcome. *Proceedings of the International Conference on Industrial Engineering and Operations Management*, (July):1501–1512.

Ottaviani, F. M. et al. (2022). Multiple Linear Regression Model for Improved Project Cost Forecasting. *Procedia Computer Science*, 196:808–815.

Park, S.-H., Lee, T., and Kim, S.-C. (2018). Investigating the Impacts of the Quality of Project Management Information System on Project Performance and User Satisfaction. *Journal of Society of Korea Industrial and Systems Engineering*, 41(3):50–60.

Rahman, H., Shafique, M. N., and Rashid, A. (2018). Project Success in the Eyes of Project Management Information System and Project Team Members. *Abasyn Journal of Social Sciences*, (July):18.

Raimond, L. (1987). Information systems design for project management: a data modeling approach. *Project Management Journal*, 18(4):94–99.

Raymond, L. and Bergeron, F. (2008). Project management information systems: An empirical study of their impact on project managers and project success. *International Journal of Project Management*, 26(2):213–220.

Rumbaugh, J., Jacobson, I., and Booch, G. (2004). *Unified Modeling Language Reference Manual, The (2nd Edition)*. Pearson Higher Education.

Schoultz von, F., Malzahn, U., and Schulz, R. (1996). An Integrated Project Management Information System. Technical Report 33, Turku Centre for Computer Science.

Taniguchi, A. and Onosato, M. (2018). Effect of Continuous Improvement on the Reporting Quality of Project Management Information System for Project Management Success. *International Journal of Information Technology and Computer Science*, 10(1):1–15.

Teixeira, L., Xambre, A. R., Figueiredo, J., and Alvelos, H. (2016). Analysis and Design of a Project Management Information System: Practical Case in a Consulting Company. *Procedia Computer Science*, 100:171–178.

van Besouw, J. and Bond-Barnard, T. (2021). Smart project management information systems (Spmis) for engineering projects – project performance monitoring & reporting. *International Journal of Information Systems and Project Management*, 9(1):78–97.

Waheed, F., Azam, F., Waseem Anwar, M., and Kiran, A. (2019). A Meta-model for Planning and Execution Activities in Software Project Integration Management. *Proceedings of the 9th International Conference on Information Communication and Management*.

Yeganegi, K. and Safaeian, S. (2012). Design of Project Management Information Systems. *International Conference on Industrial Engineering and Operations Management*, pages 2545–2551.