Developing a Standards Based Project Management Information System[1, 2]

Part 2: Project Schedule Functions[3]

By Jeremy C. Bellah, Ph.D., PMP

University of Oklahoma
Norman, Oklahoma, USA

Introduction

This is one of a series of articles that describes the development of a project management information system (PMIS). Using the design science research methodology, I created an artifact, and that artifact is the research contribution. In the first article, I described the overall development process. In subsequent articles, I describe in depth specific functions in the system. In the following pages, I describe the design of the project schedule functions in the PMIS. I conclude with future direction and pending questions. My hope is to spark interest and engage project management practitioners in the project.

To develop the PMIS, I used a low-code development tool called Mendix. This allowed me to develop the functionality much more quickly than I would have been able to otherwise. Additionally, it allows me to show program logic visually in these articles. Key portions of program logic will be displayed where appropriate.

Processes and Data

The PMIS is designed to mirror the standards in "A Guide to the Project Management Body of Knowledge," more commonly known as the PMBOK® Guide (Project Management Institute, 2017); therefore, the functionality is described according to the content in that standard. The first article in my series explains the overall design of the system (Bellah, 2021). In that article, I explained how some of the documents (usually inputs and outputs of project management processes) are integrated into the system using database tables and custom interfaces. The

[1] This series of articles describes the development of a new standards-based project management information system by prof Jeremy Bellah and student researchers at the University of Oklahoma. Dr. Bellah is director of the Center for MIS Studies at the University and teaches courses on data analysis, web design, information systems and project management. See Dr. Bellah's background at the end of this article.

[2] As this is an ongoing development project, the author would welcome comments, feedback or suggestions. Please email your thoughts or ideas to jbellah@ou.edu

[3] How to cite this article: Bellah, J. C. (2021). Developing a Standards Based Project Management Information System – Part 2: Project Schedule Functions, *PM World Journal,* Vol. X, Issue III, March.

following documents used in processes in the Project Schedule Management knowledge area are integrated in the PMIS:

- Activity list
- Activity attributes
- Milestone list
- Network diagrams
- Duration estimates
- Basis of estimates
- Project schedule
- Schedule baseline
- Work performance information

A database supports all of the integrated documents in the PMIS. Figure 1 shows an Entity Relationship Diagram (ERD) of the tables required to implement the functionality for the documents listed above. To save space, most attributes are not included in the ERD.

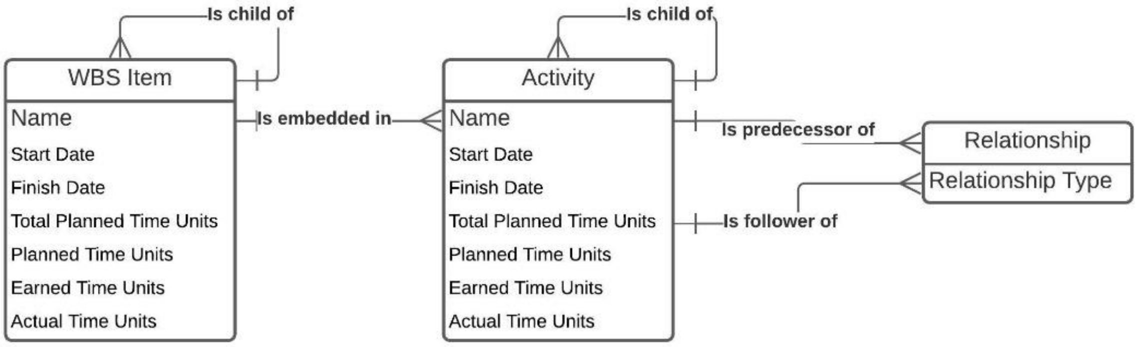


Figure 1: Entity Relationship Diagram

Activity List

The interface for creating the Activity List allows for adding, editing, and deleting activities. Each activity must be embedded within a work package, which is the lowest level of a work breakdown structure (WBS) item. Activities can be decomposed so that there are lower-level activities embedded within higher level activities. Additionally, activities can be designated as milestones, which have 0 durations and are used to track important accomplishments in project work. When the Activity List is approved, the system designates each activity as: an activity group, an activity, or a milestone.

Design Decisions: Even though the work package to which an activity belongs can be derived by traversing relationships (activity >> activity group >> activity group >> work package), I decided to assign the relationship to the work package for all activities, even those logically embedded within activity groups. This made other functions more efficient because the work package was easily accessible through the relationship.

Because so many items are related to the activities in the PMIS, changes to those activities might cause problems. For example, activities (rather than activity groups or milestones) can consume resources and can participate in precedence relationships. If an activity is changed to an activity group or milestone, those relationships are corrupted. Even though the database might still have the data that relates the tables, the interfaces are designed for only working with activities, so the user would be unable to see or modify the data. Deleting an activity would cause similar issues. For this reason, there is an extensive error checking algorithm that runs when the user clicks Approve Activity List.

Activity Attributes

This document was easy to implement, as the Activity List interface handled the creation and naming of activities. This document displays additional attributes, such as description, constraints, and assumptions, as well as information accessible through relationships in the PMIS, such as WBS ID, predecessors, and successors.

Milestone List

This document was also easy to implement because the interface for creating the Activity List allows for designating activities as milestones.

Network Diagrams

I did not implement network diagrams because of the graphical complexity; however, I developed an interface for the creation and modification of precedence relationships. I defined four types of logical relationships: finish-to-start, finish-to-finish, start-to-start, and start-to-finish.

Design Decisions: Only lowest-level activities can participate in precedence relationships. The first time I developed this interface, I allowed activity groups to participate as predecessors and successors. Although I was able to make it work, the complexity was not worth the benefit of flexibility.

Duration Estimates

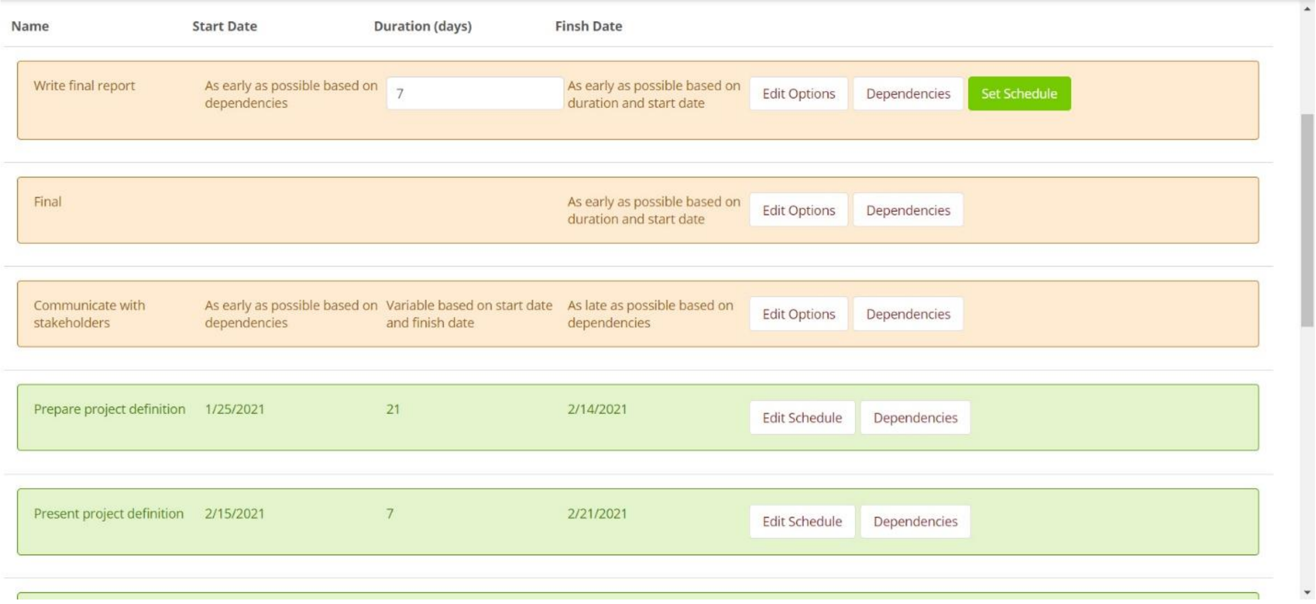
I designed the interface for entering duration estimates in the most simple way possible. This is an interface I plan to improve. In the current system, all duration estimates are in days. I plan to add functionality to allow durations to be entered in different time increments. I also plan to add functionality for three point estimates so that the system can use the program evaluation and review technique (PERT) to estimate the project duration.

Basis of Estimates

This document was easy to implement, as it required only an extra attribute in the Activity entity.

Project Schedule/ Schedule Baseline

For the project schedule interface, I tried to balance power and flexibility. Figure 2 shows the scheduling interface.

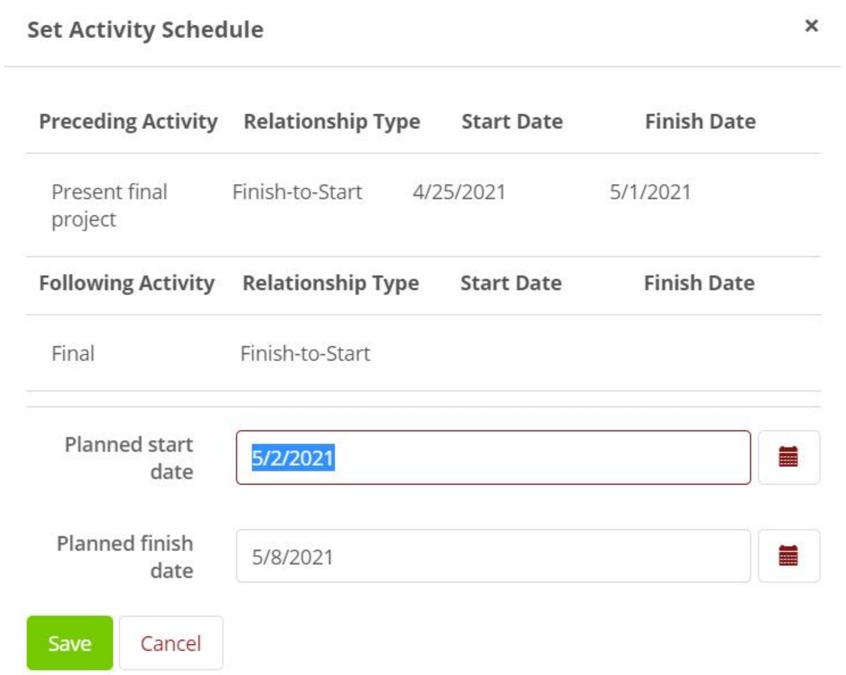


Name	Start Date	Duration (days)	Finsh Date	
Write final report	As early as possible based on dependencies	7	As early as possible based on duration and start date	Edit Options Dependencies Set Schedule
Final			As early as possible based on duration and start date	Edit Options Dependencies
Communicate with stakeholders	As early as possible based on dependencies	Variable based on start date and finish date	As late as possible based on dependencies	Edit Options Dependencies
Prepare project definition	1/25/2021	21	2/14/2021	Edit Schedule Dependencies
Present project definition	2/15/2021	7	2/21/2021	Edit Schedule Dependencies

Figure 2: Scheduling interface

Activities that need to be scheduled appear with an orange background, and they are listed first. If the appropriate preceding and following relationships have been scheduled, the green Set Schedule button appears. Activities that have been scheduled appear with a green background.

To maximize the power of the scheduling function, multiple options are available for how the system will schedule each activity. The start date can be fixed, or it can be variable based on precedence relationships (as early as possible or as late as possible). The duration can be fixed or variable. The end date can be fixed, or it can be variable based on precedence relationships (as early as possible or as late as possible). The consequence of implementing these powerful options is that activities must be scheduled one at a time. An activity is scheduled by clicking the green Set Schedule button. That pops up an entry screen that can be used to schedule start and finish dates. This screen is shown in Figure 3. The dates are already filled in based on configuration options. Additionally, preceding and following activities are displayed for reference.



Set Activity Schedule ×

Preceding Activity	Relationship Type	Start Date	Finish Date
Present final project	Finish-to-Start	4/25/2021	5/1/2021

Following Activity	Relationship Type	Start Date	Finish Date
Final	Finish-to-Start		

Planned start date: 5/2/2021

Planned finish date: 5/8/2021

Figure 3: Activity schedule screen

To make this work in the system, when the Set Schedule button is clicked, preceding and following relationships are traversed to determine the appropriate values for the start and finish dates. If the activity is configured to start as early as possible, the finish-to-start and start-to-start relationships are examined to determine the earliest date that can follow the preceding activities. If the activity is configured start as late as possible, the start-to-start and start-to-finish relationships are examined to determine the latest date that can precede the following relationships. A similar process happens for the finish date. When the user clicks Save, the Total Planned Time Units are calculated as the difference between the Start Date and Finish Date. Then the system looks at preceding and following activities to see if scheduling the current activity made other activities eligible to be scheduled. This update process is accomplished using the code in Figure 4.

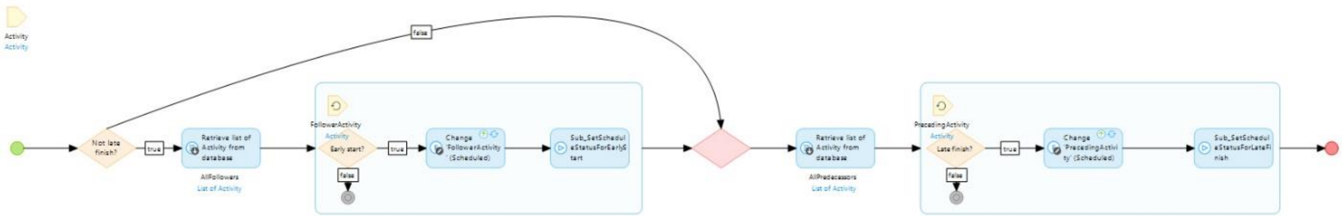

Activity
Activity
Not late finish?
true
false
Retrieve list of Activity from database
AllFollowers
List of Activity
FollowerActivity
Activity
Early start?
true
false
Change 'FollowerActivity' (Scheduled)
Sub_SetScheduleStatusForEarlyStart
Retrieve list of Activity from database
AllPredecessors
List of Activity
PrecedingActivity
Activity
Late finish?
true
false
Change 'PrecedingActivity' (Scheduled)
Sub_SetScheduleStatusForLateFinish

Figure 4: Update dependent activity schedule status (code in Mendix)

After all activities are scheduled, the Approve Schedule button can be clicked. The information from activities is then aggregated up through the project structure. For activity groups, the Start

Date is the earliest Start Date of child activities. The Finish Date is the latest Finish Date of child activities. The Total Planned Time Units is the sum of Total Planned Time Units of child activities. After values are calculated for activity groups, those values are aggregated up to WBS items, and to the project as a whole.

Work performance information

The system allows users to enter information needed for earned value analysis. Specifically, actual cost and percent complete can be entered for each activity. Each day, an algorithm runs to update the planned values of activities. All of the information is aggregated up through the project structure so that earned value information can be calculated at every level.

Conclusion

In this article, I described the scheduling functionality in the PMIS. By using a database to connect information from different project management processes and documents, the system is able to provide power and flexibility in the scheduling domain. Most of the functionality happens in the project schedule document, but the information in other documents is essential for that to work. In creating the interfaces, I followed the core information in the PMBOK® Guide (Project Management Institute, 2017). However, I have ideas about improving the system about which I would like to hear from professionals.

Following are the specific ideas about which I need feedback:

- What would be the best way to analyze schedule performance? The standard earned value analysis calculations are already implemented. I plan to add support for using Earned Schedule (Lipke, 2014) as well. Additionally, the way I collected data allows for different calculations. Suggested calculations are as follows:

  Total Planned Time Units = number of total days an activity is scheduled
  Planned Time Units = number of days elapsed since the activity was scheduled to start
  Actual Time Units = number of days elapsed since the activity actually started
  Earned Time Units = (Percent Complete) X (Total Planned Time Units)

  Having calculated those values for lowest level activities, they can be aggregated up through the project structure. Then they can be used to measure schedule performance. This makes sense to me practically, and I like the way it shifts the calculation of schedule performance away from budgeted amounts. But I would be interested to hear from those of you working in various industries.

- Are there schedule-related functions you would like to see in a project management system that are not in commercially available systems? The benefit of sharing the development of this system openly is that it allows for collaboration. Also, because I am not planning to commercialize it, the collaboration can happen without impediments.

Feel free to use the system. Following is the URL:

https://trjproject-sandbox.mxapps.io/index.html?profile=Responsive

Remember that the system is still being developed, so there could be errors. You can email me if you encounter errors, and I will try to resolve them.

I hope you enjoyed reading about the development of the PMIS. I look forward to hearing back from you. Please email me at jbellah@ou.edu with feedback related to my specific questions or with general feedback. In the next article, I plan to go into detail with the budgeting functionality.

References

Bellah, J. C. (2021). Developing a Standards Based Project Management Information System – Part 1: Motivation, History, Design; Series, *PM World Journal,* Vol. X, Issue I, January Available online at https://pmworldlibrary.net/wp-content/uploads/2021/01/pmwj101-Jan2021-Bellah-developing-standards-based-PMIS-1-motivation-history-design.pdf

Lipke, W. (2014). "Introduction to Earned Schedule," *PM World Journal*, Vol. III, Issue XI, November 2014. https://pmworldlibrary.net/wp-content/uploads/2014/11/pmwj28-nov2014-Lipke-introduction-to-earned-schedule-utd-second-edition.pdf

Project Management Institute. (2017). *A guide to the Project Management Body of Knowledge (PMBOK guide)* (6th ed.). Project Management Institute.

About the Author

Jeremy Bellah, PhD

Oklahoma, USA

Jeremy Bellah serves in a teaching position at the University of Oklahoma's Price College of Business. He teaches classes related to programming, web development, data analysis, and project management. Through an experiential learning teaching style, he coaches and mentors students to develop into successful working professionals. He also serves as the Director of the Center for MIS Studies, an industry-academic partnership within the MIS Division. In this role, he interacts with industry partners of the Division to facilitate input on curriculum, collaborative research projects, and engagement with students.

Prior to earning a Ph.D. and pursuing academic positions, Jeremy developed information systems professionally. Most of the systems were custom applications built for small- and medium-sized businesses. This experience gives Jeremy the ability to relate to industry professionals and to teach from a perspective of practical expertise, rather than academic training. Dr. Bellah can be contacted at jbellah@ou.edu.