Model-driven Design and Management of Professional Education Programmes

Dennis Wolters[1,*], Gregor Engels[1]

[1]*Department of Computer Science, Paderborn University, Germany*

Abstract

Digital transformation requires professional education programmes to help understand new business technologies and foster innovative thinking. This paper proposes a model-driven approach for designing and managing professional education programmes across all life cycle phases. As part of this, we present a visual notation for the analysis and design phase to describe dependencies and the temporal structure of those programmes. Furthermore, our approach allows the explicit linkage of processes to design and manage a programme to the model describing it. Web-based tool support is built on top of the collaborative online whiteboard Miro and by integrating with the project management tool OpenProject.

Keywords

professional education, model-driven development, instructional design

1. Motivation

Professional education programmes are essential for digital transformation as they help understand new business technology and foster innovative thinking. Furthermore, they play a significant role in developing, attracting and retaining talent [1]. Bespoke professional education programmes are needed to target a company's needs adequately. Such bespoke programmes are created in close cooperation with client companies and contributing educators. Programme designers and managers have to plan and coordinate the processes steering this collaboration, which, nowadays, often has to happen remotely. Models representing such programmes are well-suited as mediating artefacts in such collaborations [2]. Modelling languages often focus on specific phases and neglect the process perspective. For instance, approaches like MyScripting [3], STOPS [4] and CIAT [5] focus on programme development, but the process perspective is out of scope. The instructional engineering approach MISA [6] covers the process perspective by defining its own design process. However, MISA does not assist in process enactment and the flexibility to use other processes is limited.

The awareness of the problem that processes are managed separately from designing and running a professional education programme led to a start of a design science research cycle [7]. This extended abstract provides an overview of our suggested model-driven approach that couples professional education programme design and process management. In contrast to other approaches, we model such programmes on a high level of abstraction across their entire

ICSOB'22: 13th International Conference on Software Business, November 8-–11, 2022, Bolzano, Italy

*Corresponding author.

dennis.wolters@uni-paderborn.de (D. Wolters); engels@uni-paderborn.de (G. Engels)

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3316/poster-paper4.pdf

life cycle. We leave the option to use existing approaches and tools for detailed planning. Furthermore, our approach assists in enacting design and management processes without prescribing specific processes. Web-based tool support based on the online whiteboard Miro enables the use of our approach in collaborative remote settings. Section 2 gives an overview of our approach. Section 3 provides details on tool support, and Section 4 concludes the paper and gives an outlook on future work.

2. Modelling Professional Education Programmes

The basis for our model-driven approach to design and manage professional education programmes is a modelling language that can describe such programmes across their entire life cycle. As depicted in Figure 1.a, the metamodel of our Professional Education Programme Modelling Language (PEPML) is subdivided into several packages. The Foundation package defines concepts required by all packages, such as versions, iterations and a base class for all programme entities. Five additional packages are named after the instructional design process ADDIE [8] that covers all life-cycle phases of education programmes, i.e. Analysis, Design, Development, Implementation and Evaluation. Each of these packages contains concepts for the respective phases. The Enactment package provides support for enacting processes that relate to a programme. Structuring the metamodel based on ADDIE is solely done to provide a more transparent (internal) structure, and the Enactment package does not require following an ADDIE-based process. For instance, programmes designers can use our approach in combination with an agile instructional design process such as the Successive Approximation Model [9]. In that case, concepts of the different packages are utilised as needed by the design process.

PEPML supports two viewpoints (see Figure 1.b): (1) The dependency viewpoint covers the interrelation between the different entities of an education programme, e.g. learning goals,

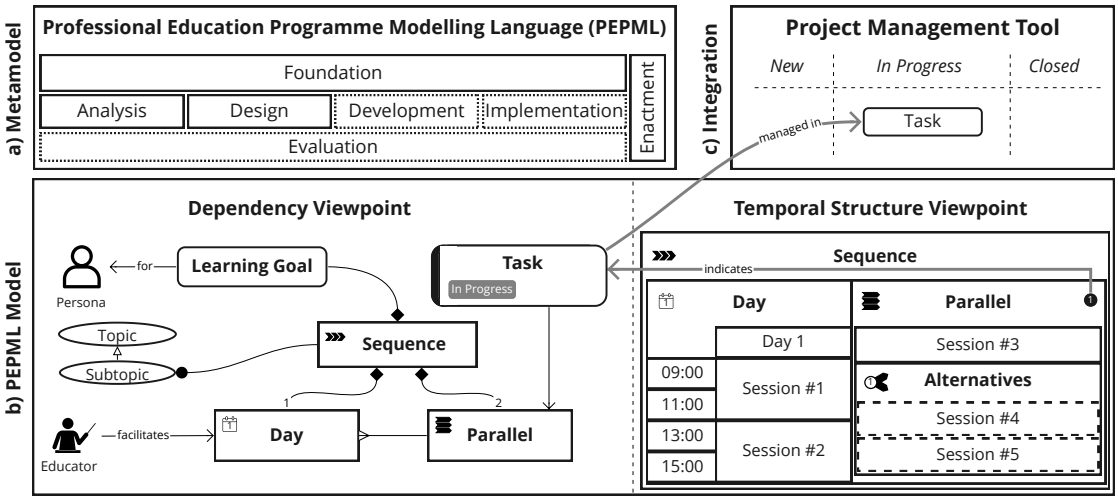

Figure 1: a) Packages of the PEPML metamodel (content of the packages with dashed borders is tentative), b) Example of a PEPML model from the dependency viewpoint (left) and the temporal structure viewpoint (right) , c) Integration with project management tool

topics or educational components like lectures or group work. This viewpoint is especially important for programme designers and their clients to capture and align learning goals, content and available resources. (2) The temporal viewpoint visualises how educational components are distributed over time. This viewpoint is relevant to all stakeholders to understand, communicate and discuss a programme's structure. Up to now, PEPML focuses primarily on analysing and designing professional education programmes. Additional viewpoints and refinements of the packages Development, Implementation and Evaluation are subject to future work.

Programme designers and managers must keep track of all tasks required to design and run an education programme. The Enactment package allows associating tasks with any entity of an education programme. In the dependency viewpoint, these tasks are connected via associations to the respective entities. At the same time, a small indicator is used in the temporal structure viewpoint (see ❶) to inform about the number of associated open tasks. Tasks linked to a PEPML model are synchronised with project management tooling to provide additional task management features, e.g. setting due dates or managing tasks in Kanban boards (see Figure 1.c).

Users can define completion criteria for tasks represented in PEPML models, which are evaluated when a task's status changes. When a task is completed but the completion criteria are not fulfilled, the task is put into a specified failure status. Thereby, it is possible to ensure that changes to a programme resulting from performing certain tasks are reflected in a PEPML model. For instance, a task "Find Instructor" associated with an instructor-led session could define that the instructor is represented in the PEPML model and associated with the respective session upon task completion.

3. Tool Support

Modelling tool support is provided as an app for the collaborative online whiteboard Miro[1]. We track PEPML elements on Miro boards to preserve Miro's modelling flexibility and still offer advanced features, e.g. validating the syntactical correctness and applying fixes to solve syntactical errors. PEPML language elements added to a Miro board via our custom toolbar or that are part of a PEPML frame are automatically tracked. Alternatively, users can manually define that an item represents a PEPML entity. When extracting the temporal structure, we allow slight misplacements and offer auto-positioning features to fix such misplacements.

A Node.js backend extracts PEPML models from Miro boards and persists them in a Neo4j graph database. Users can add persisted models to Miro boards via the PEPML toolbar. By persisting PEPML models in Neo4j, models become queryable using Neo4j querying language Cypher, which allows extracting (aggregated) information from models.

For enhanced task management, the tooling is integrated with the project management tool OpenProject[2]. Education programmes are represented as projects in OpenProject. Tasks defined in a PEPML model are added to OpenProject and vice versa. Webhooks inform about status changes of tasks, and users can specify completion criteria as Cypher queries, which are evaluated on the persisted PEPML model in Neo4j.

[1] https://miro.com
[2] https://openproject.org

4. Conclusion & Future Work

Digital transformation is not just a technical or organisational challenge but also an educational one. Bespoke professional education programmes are needed to prepare employees for digital transformation and help them shape it. In this extended abstract, we present a model-driven approach to design and manage such programmes. The basis for the approach is a modelling language covering all life cycle phases of professional education programmes. In contrast to purely focussing on programme design, our approach also supports the enactment of associated processes. Our modelling tool support builds upon the digital whiteboard Miro and is well-suited for remote collaboration. Enhanced process enactment support is realised by integrating with the project management tool OpenProject.

For future work, we plan to finalize the Development, Implementation and Evaluation packages and conduct a user study to evaluate our modelling language and tooling with external users to complete our design science research cycle. By allowing associating tasks with entities of an education programme, we already support the enactment of processes without actually prescribing any specific process. In the future, we want to extend our work to include process modelling and create a situational method engineering approach for instructional design.

References

[1] E. Kyndt, F. Dochy, M. Michielsen, B. Moeyaert, Employee Retention: Organisational and Personal Perspectives, Vocations and Learning 2 (2009) 195–215.

[2] G. Conole, The Role of Mediating Artefacts in Learning Design, in: Handbook of Research on Learning Design and Learning Objects: Issues, Applications, and Technologies, IGI Global, 2009, pp. 188–208.

[3] C. Müller, J. Erlemann, Educational Design for Digital Learning with myScripting, in: EDEN Conference'22, 2022, pp. 128–134.

[4] T. Auvinen, J. Paavola, J. Hartikainen, STOPS: A Graph-based Study Planning and Curriculum Development Tool, in: Koli Calling '14, ACM, 2014, pp. 25–34.

[5] A. I. Molina, F. Jurado, I. de la Cruz, M. A. Redondo, M. Ortega, Tools to Support the Design, Execution and Visualization of Instructional Designs, in: Cooperative Design, Visualization, and Engineering, Springer, 2009, pp. 232–235.

[6] G. Paquette, I. de la Teja, M. Léonard, K. Lundgren-Cayrol, O. Marino, An Instructional Engineering Method and Tool for the Design of Units of Learning, in: Learning Design: A Handbook on Modelling and Delivering Networked Education and Training, Springer, 2005, pp. 161–184.

[7] B. Kuechler, V. Vaishnavi, On Theory Development in Design Science Research: Anatomy of a Research Project, European Journal of Information Systems 17 (2008) 489–504.

[8] R. M. Gagne, W. W. Wager, K. C. Golas, J. M. Keller, J. D. Russell, Principles of Instructional Design, Wiley Online Library, 2005.

[9] M. W. Allen, R. Sites, Leaving ADDIE for SAM, American Society for Training and Development, 2012.