Creating Vividness through Executable Models: A Teaching Case for Conceptual Modelling

Fabienne Lambusch[a], Håkan Enquist[b] and Tomas Jonsson[c]

[a] *University of Rostock, Business Information Systems, Rostock, Germany*
[b] *University of Gothenburg, Gothenburg, Sweden*
[c] *Genicore AB, Gothenburg, Sweden*

Abstract
When teaching enterprise modelling in academic settings, it is challenging to combine theory with hands-on experience. We have created an assignment for iterative prototyping of a self-management application for students using executable models as a means of vividness. The completion of the task included the examination of some self-management theories, the design of an interview guide, exploratory interviews with domain representatives, conceptual modelling of key phenomena to satisfy information needs, and creating a prototype. We present the design of the assignment and the procedure of task completion combining gaining theoretical knowledge with hands-on experiences in the whole development cycle. As this procedure could function as a blueprint for other teaching cases, we contribute a generalised representation of the vivid approach and reflect on what to consider when reusing it. This approach for vivid conceptual modelling in assignments will be further developed and is intended to be used in a broader range of teaching cases in the information systems curriculum.

Keywords
Teaching, Conceptual Modelling, Self-Management, Executable Model, Information Systems

1. Introduction

Model-driven development offers high flexibility in the creation of information systems (IS), as conceptual models can serve as specifications for automatically generating and editing running systems. For this purpose, powerful computer-aided software engineering (CASE) tools have emerged that provide integrated modelling and generation capabilities. A continued collaboration has evolved among us authors with regard to conceptual modelling and the design of IS. One purpose of this collaboration is to increase awareness for these topics and their interplay, which led to the initiation of a student project in a course of enterprise modelling applications.

In academic teaching contexts, IS modelling is often taught more theoretically and separated from running systems. With tools that integrate modelling and model execution, it is easy to create a first running prototype of an IS from a conceptual model and thus, to create vividness

PoEM'20 Forum: 13th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modelling, Forum, November 25–27, 2020, Riga, Latvia
EMAIL: fabienne.lambusch@uni-rostock.de (F. Lambusch); hakan@enquist.se (H. Enquist); tomas@genicore.se (T. Jonsson)
ORCID: 0000-0002-0303-1430 (F. Lambusch); 0000-0001-7907-8596 (H. Enquist); 0000-0002-3948-6525 (T. Jonsson)

out of models. It has been shown for teaching conceptual modelling that integrated model execution significantly improves students' learning results [1]. Proposed teaching cases typically use predefined descriptions of domains to be modelled and define an expected result model for the evaluation of the results presented by the students. While this is important in teaching the syntax and semantics of a modelling language, a broader focus is needed when teaching modelling for enterprise IS. For the latter case, we think that a domain exploration of an enterprise domain in combination with information modelling and model execution could well support students' learning and train their view for the bigger picture as needed in real life situations. However, it is not clear so far how to design and supervise such a student task.

Hence, we prepared a student task covering a whole design loop, from domain exploration modelling through information systems modelling to system model executions. The final result to be presented was a model and prototype of an information tool for student self-management (see Section 4.1). The design loop included clarification of theoretical basics, surveying domain representatives, conceptual modelling of key phenomena, and creation of a running prototype. The development tool used is CoreWEB [2], which is provided by Genicore in agreement with the University of Rostock. It is a web-based graphical CASE-tool that supports an iterative process of conceptual information modelling and execution, instance data creation as well as user interface view configuration. By including conceptual modelling in the whole loop, resulting in a working prototype, we intend to make the teaching approach vivid. In this paper, we describe the teaching case and discuss our experiences from the task execution. Furthermore, we contribute a generalised representation of the approach that could be used in other teaching cases and reflect on its adaption.

The remainder of the paper is structured as follows. The next section presents other works related to teaching conceptual modelling and explains what we add with this article. Section 3 describes the academic setting for our teaching case. Section 4 then presents our vivid approach to conceptual modelling. In this we provide both, a description of our concrete teaching case as well as a generalised representation of the procedure covering a whole design loop for iterative prototyping. Furthermore, we give advice on the reuse of the approach for other teaching cases. We discuss our results in Section 5 and conclude the work in Section 6.

2. Related Work

A model is an instrument with a function and a purpose. The function of a model can, for instance, be to document and communicate a domain of interest, in either a descriptive or prescriptive scenario [3]. A conceptual model is a model enhanced with concepts often used in information system development scenarios [4]. The conceptual model can be a description of a domain of interest for a community of intended users or it can be a prescription for the construction of an information system.

In basic IS-related education scenarios for conceptual modelling, students are given small samples and tasks to model, which are given as textual system requirements, i.e. prescription for system construction. These requirements are to be interpreted by students and expressed in conceptual models, in form of a data model and/or a process model, i.e. prescriptive models. CaMeLOT, based on a revised Bloom's taxonomy, is an educational framework for structuring

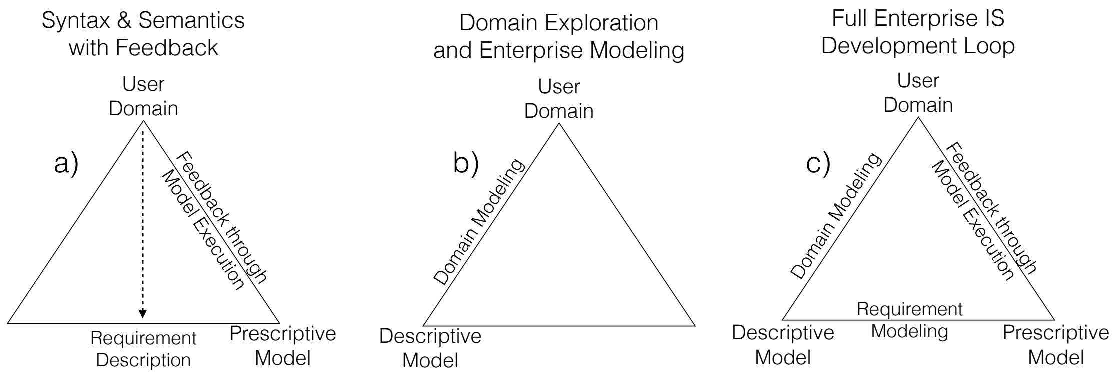

Figure 1: The figure shows three different conceptual modelling teaching scenarios. Teaching a) information modelling syntax and semantics with feedback through model execution, b) domain exploration and enterprise modelling, and c) our case, combining both domain descriptive model and IS prescriptive model with feedback through model execution.

such courses, but could be used in more general cases [5].

The correlation between the prescriptive model and the final system in this approach is not explicitly explored. Thus, there is a chance that the models are perceived as abstract artefacts rather than meaningful instruments. By using modelling tools which allow for model execution, students can get feedback and explore how their models affect the system function and if the function corresponds to stated requirements (see Fig.1 a), which greatly improves understanding of conceptual models as an instrument [1].

In practice of developing IS, the analyses and documentation of a user domain is a crucial task for which descriptive conceptual modelling can be used. For enterprise information systems development, enterprise modelling is a means to describe an enterprise as a domain of interest (see Fig.1 b). In this context, Petersen and Krogstie describe experiences of teaching descriptive enterprise modelling to students already familiar with basic IS conceptual modelling [6].

The project described in this paper was designed to let students apply their enterprise and conceptual modelling knowledge in a real world development scenario. This project includes both, the domain exploration with associated conceptual modelling and requirement specification in form of a prescriptive conceptual model, as shown in Fig.1 c). The prescriptive modelling is made in a tool which supports transformation of a model into an executing information system.

3. Academic Setting

In this article, we describe our experiences with the student task that we developed and supervised together. While the next section describes our teaching approach, this section describes the general setting in which the student task was implemented in order to inform about the circumstances of task completion.

3.1. Course Settings

The student task was part of a master's level course on enterprise modelling applications. The course comprises lectures on enterprise modelling by the course leader, which focus on four areas: methods of modelling, analysis of enterprise models, quality aspects, and frameworks and standards. Furthermore, the students have to complete a task on current research issues in the subject area, which shall give them a deeper understanding of the course contents by working on a specific case. To this end, the students work in small teams. It is intended that each student team works on a different task. For this purpose, the various academic staff members draw up different task descriptions, which they in turn supervise, so that a list of alternative tasks with different supervisors is available. A task can optionally be supervised by additional persons, such as in our case, where a practitioner and a tool expert were additional supervisors for a task. Each student team then has to choose a task from the list based on interests and skills. For the successful completion of the course, the students have to write a seminar paper on their results for the chosen task and present them in a colloquium. There is no additional exam, but instead the final grade is based on the seminar paper and colloquium.

3.2. Supervision

A team of two students chose our task for iterative prototyping of an information tool for student self-management. As we supervised the student team together, the supervisors had different backgrounds ranging from academic through practice and consulting to tool development. As the supervision during the semester was relatively free in its arrangement, the supervisors of each student team could choose on the roadmap and appointments. We defined a rough roadmap for our student team beforehand based on an iterative prototyping process, but the details were planned together in supervision meetings. Due to the Covid-19 crisis the meetings were held via a web conference tool. While the students got initial material for the steps to be taken, in the meetings not only their questions were answered, but also methodology and practical knowledge e.g. for tool use regarding the task completion were taught in more detail. So the students could advance their knowledge in a specific area of modelling, in this case regarding IT-support for student self-management.

4. Vivid Approach

In this section, we first provide information about the concrete teaching case on student self-management that we tested with a student group. The prepared task covered a whole design loop for iterative prototyping, what is reflected in the procedure of task completion. The procedure worked well in the first run and seems promising for use in other teaching cases, optionally with another domain in enterprise modelling. Thus, we show a generalised representation of the vivid approach and in the second subsection give advice on its reuse.

4.1. Task Description and Completion

Following on from an existing research project on IT-based support for occupational self-management [7, 8], a conceptual data model and prototypical information tool for self-management

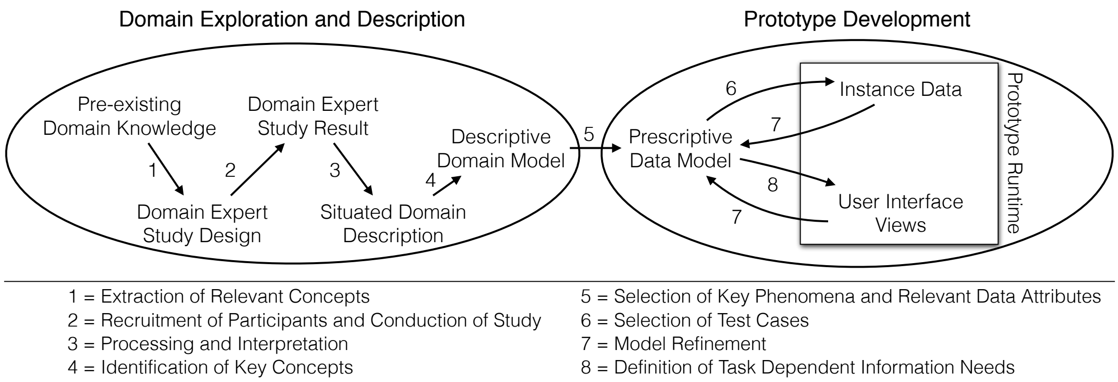

Figure 2: Procedure of the vivid approach. There are two phases illustrated by the ellipses. The first is the exploration and description of the domain and the second is the development phase. Within the ellipses, a node represents a task deliverable, while an edge represents an activity to be pursued. The development phase can consist of iterative steps, where several cycles can be passed through to refine the model and thus, the prototype.

should be developed for a limited target group. Due to time constraints, a domain was chosen which students are familiar with and easily find interview partners: student self-management. Similar to knowledge-intensive work, student life is often characterised by weakly structured processes and higher degrees of freedom. Self-management is in particular relevant for persons who enjoy considerable freedom of choice [9], which is why student self-management is an important case.

Task Description

The task description, which comprised about half a page of text, motivated the topic, briefly explained the most important terms, described the aim, and suggested driving questions and a rough process on how to proceed. The aim of the task was to gather information about relevant self-management aspects through literature and interviews with other students, develop a conceptual model based on these, create a prototype, and test it against the needs stated in the interviews.

Procedure of Task Completion

The task was designed to put conceptual modelling into a context as a bridge between domain exploration and IS design. Thus, it covers a full enterprise IS development loop from domain exploration through IS modelling to system model executions. We generalised the representation of the final procedure of the task completion as shown in Figure 2 and describe in the following text the implementation in our concrete teaching case on student self-management.

Before the start of the student project, initial material was provided, especially for self-management theory [10, 11] and the modelling ontology supported by the CoreWEB [12] as well as instructional videos for the modelling and execution tool.

In the first phase of ***domain exploration and description***, the students first studied the *pre-existing domain knowledge* in terms of the self-management theories. As the preset instrument for the domain study was interviewing, they took terms from the self-management literature to refine an initial interview guide given by the supervisors as part of their *domain expert study design*. After recruiting and interviewing peer students on their self-management, they used the *domain expert study result* in terms of the recorded interviews to create transcriptions of the interviews and a description of student self-management as part of the *situated domain description*. Through identifying key information, they created a mind map of student self-management concepts as kind of a *descriptive domain model*.

In order to pass over to the second phase of ***prototype development***, the students had to select key phenomena and corresponding data from the mind map concepts to create a conceptual data model as a type of a *prescriptive data model* for the information system. They then tested the model for relevance by executing the model. The CoreWEB tool produces a default generated user interface which allows the system to be populated with data. By selecting some of the student self-management cases, they created *instance data* in the prototype runtime. They then considered whether they needed to refine the model according to their experiences with the test data. Once the initial model version was deemed adequate,the default user interface was further configured with some task oriented *user interface views*, while iteratively refining the model adding additional properties and rules and re-checking with the instance data.

Results

The student team had to present their work in a colloquium and to submit a seminar paper to their supervisors with a structure similar to a research paper. In this, they introduced the topic, described the terms, methods, and tools considered, described and discussed their results, and concluded their work. The student team interviewed six fellow students in total.

The first model contained most concepts from the mind map predominantly as entities and relations, which lead to, for users, a complicated system. It was a revelation to the students how the model and executing system interplay and that an easy to understand system needs an uncomplicated conceptual model. The final conceptual model uses five different phenomenon categories and contains a total of 13 phenomenon types with attributes and 20 relations. For their prototypical system generated from the model, they created two different views and inserted some exemplary data.

4.2. Advice on Reusing the Approach

The vivid approach reflected in Figure 2 could be implemented in many different shapes. As a lecturer or supervisor, one has to determine the overall setting for the task before reusing the approach, e.g. what material to prepare for the students and what methods and tools to consider. In the following, we provide a frame for this. For information on our lessons learnt when using the approach, see the discussion in Section 5.

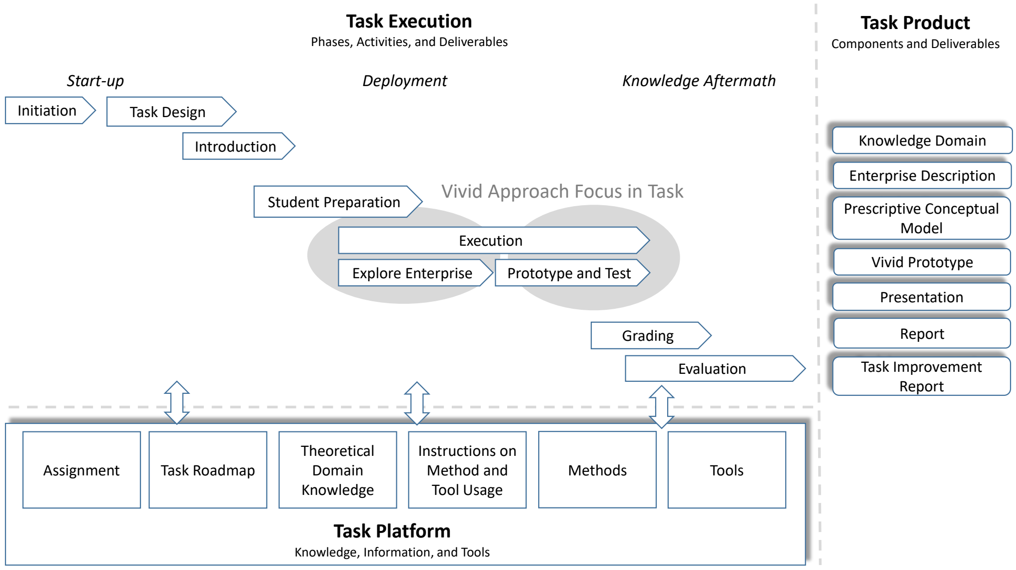

Figure 3: Task preparation canvas. This canvas gives an overview on what to consider when creating and executing a student project according to the vivid approach.

4.2.1. Canvas for Supervision

The canvas shown in Figure 3 provides advice on what to think through when planning for a student project according to the vivid approach. It can be used to work through the content of all three main sectors shown - task execution, task product and task platform - in order to plan for the task.

The sector ***task execution*** provides a generic set of steps that need be prepared and executed throughout the whole case with the vivid approach. Thus, it is intended to constitute a basis for lecturers to create a roadmap and overall plan for student activities and supervision actions. It contains three main phases, namely startup, deployment, and knowledge aftermath. The startup phase comprises the *initiation* of the task idea and the supervisor team as well as the *task design*, in which also the supervision strategy and material to be prepared is determined for the final task description. The deployment phase includes engaging and *introducing* the students to the task and supervising them in *preparations* for executing the task. The *execution* phase is in focus of the vivid approach, since here the hands-on modelling of the *enterprise domain* and the vivid *prototyping* take place. Finally, knowledge aftermath comprises *grading* activities, the *evaluation* of the approach, and reporting (internally or by publications). The sector ***task product*** presents the kinds of deliverables to be produced by the students pursuing a vivid case. The sector ***task platform*** represents the set of resources to be prepared and made available to the students in order to direct their focus and work efforts while executing the project.

We recommend to pre-equip the task platform in order to emphasise the vividness and thus, the process of execution by which several deliverables are created. In our concrete teaching case on student self-management, we pre-equipped the task platform for example with planning directives, literature on self-management, an initial interview guide, literature on methods, and the CoreWEB tool for conceptual modelling with integrated model execution.

When reusing this canvas for creating instances of the vivid approach for student tasks, a tailoring effort is recommended. This includes working through the content of all three sectors and deciding and making a choice on specific conditions, difficulty level of students preparation efforts and prototyping task as well as on expected workload of supervisors by right-sizing pre-loaded contents of the task platform. A rich content will empower students to a more independent project execution and may serve as a focusing instrument by being thin where students should work hard and being filled for parts that are just enabling the vivid section of the task.

4.2.2. Methods and Tools for the Vivid Approach

In order to execute this teaching approach, adequate tool support is required. For the descriptive work, there is a multitude of tools depending on the domain and modelling method, such as expert survey or observation, enterprise modelling, process modelling, or goal modelling. For the prototype development a tool is desirable, which allows for direct execution of the conceptual model, for two reasons. The executing system should reflect exactly the semantics of the conceptual model and the students' attention should be focused on model semantics, not model annotations, additional coding, or user interface graphics design. Apart from the CoreWEB tool used in this case, the MERODE modelling tools used in [1] has been proven adequate.

The conceptual prescriptive model is, however, only a part of a prescriptive IS model, except in very trivial cases. Concepts and related data presented to users at any point in time, should be contexts selected from the conceptual model and related to specific user tasks. These contexts, called views, can be described with a views model as in the CoreWEB tool.

In order for the prototype to be a vivid instrument and meaningful to evaluate the prescriptive model, the prototype needs to be populated with a consistent set of data that represents and illustrates instances of relevant domain phenomena. Preferably, intended users or domain experts should participate in populating and evaluating the prototype.

5. Discussion

As a first run, we mainly tested the clarity of the task, the feasibility regarding the time frame and foreknowledge of the students, and the adequacy of the procedure with only one team of students. While we asked the student group for feedback at the end, we did no holistic evaluation in this first run. We aim at evaluating the approach in a broader range of teaching cases in the future, but first feedback from students is quite promising. In the following, we first discuss our concrete teaching case on student self-management. As we give advice on reusing the approach in Section 4.2, we follow on with a discussion on the reuse of the approach.

5.1. Teaching Case of Student Self-Management

Addressing a well-known domain, namely student life, supported understanding and thus, the feasibility of the task. As IT-support for self-management is also *an important topic for the students themselves*, this was motivating. An *equal participation of supervisors* enabled coordinated advice to students and joint learning from the procedure of task completion. The empowerment-oriented supervision by a group with *complementary skills and experience* (academic, practitioner, tool expert) promoted engaging the students in discussions on different perspectives or ideas and helped them identifying with the topic.

It was new to the students how the conceptual model and executed system interplay, so that an easily understandable system needs an uncomplicated conceptual model. Thus, it would have been better to convey this understanding to students earlier by *assigning some small example tasks with the tool* instead of letting them start with the more complex self-management model after a short tool introduction. However, the students stated that the topic of student self-management was also difficult to condense to an uncomplicated model, because the domain is quite complex. For future projects it might thus be better to *choose just a few specific aspects in a domain to be modelled or to choose a domain with rather clear phenomena and relations.* This is also reflected in the prototype created by the students. It is only informative, reflecting parts of student self management without support for self-improvement. In the end, they found that certain information was not included in the model and they would have liked to do *another interview round with the same partners* after the first development cycle to improve the model, what was unfortunately not possible due to time constraints.

Furthermore, typical enterprise aspects like collaboration between the people in the enterprise were not a prominent part in this self-management topic. In order to highlight the enterprise context, a domain with stronger enterprise focus could be chosen, as long as it is well approachable for students.

5.2. Reuse of the Approach

So far, we cannot make statements on whether the learning process of the students was more effective than with other approaches. However, the *purposefulness of our vivid approach was perceived as high* in the first run and the students stated to be positive to future assignments with such an approach. We think that implementing the *full spectrum from domain analysis to prototype development* can motivate students and support their understanding. Thus, we want to *test the approach with other domains.* We furthermore hope to inspire other lecturers to also try the vivid approach and report about their experiences. Section 4.2 is intended to illustrate that *the approach could be implemented in many different shapes* and just requires some characteristics of used methods and tools to take into account the vivid nature. The canvas shown provides a frame for this purpose, as it presents steps for executing a task with the vivid approach together with expected deliverables and the task platform to consider. Even with other course settings the approach might be applicable. For example, if the course includes an exam in the end, bonus points could be given for the project as an incentive for the students to contribute to it. In whatever shape the approach might be used, it is important to consider *how much material should be prepared and predefined for the students.* As the proposed procedure is complex to be pursued within a student course, the amount of time available and the amount of effort needed by the students to get into the topic, methods, and tools, should be well balanced. Thus, we suggest to prepare introductory material for these parts and ease the students' workload, so that they can focus on understanding and executing adequately the domain exploration and prototype development.

6. Conclusion

In this article, we propose a vivid teaching approach covering a whole design loop for iterative prototyping, which includes clarification of theoretical basics, surveying domain representatives, conceptual modelling of key phenomena, and creation of a running prototype. We integrate executable models as a means of vividness and combine gaining theoretical knowledge with hands-on experiences. As the first test of the approach with one student group was promising, we introduce a generalised representation of the vivid approach procedure that could function as a blueprint for future student projects and give advice on the reuse of the approach. However, the approach need to be tested in a broader range of teaching cases and with different domains in order to be further refined and to be evaluated. It is a long-term objective to continue elaborating practical and theoretical knowledge on vivid approaches to practically useful conceptual modelling and IS generation in teaching and research.

References

[1] G. Sedrakyan, M. Snoeck, S. Poelmans, Assessing the effectiveness of feedback enabled simulation in teaching conceptual modeling, Computers & Education 78 (2014) 367 – 382.

[2] T. Jonsson, H. Enquist, Semantic Consistency in Enterprise Models-Through Seamless Modelling and Execution Support, in: ER Forum, volume 1979, 2017, pp. 343–346.

[3] H. Jaakkola, B. Thalheim, Functions of Models in Scenarios and Their Maturity, in: Information Modelling and Knowledge Bases, volume 321, IOS Press, 2019, pp. 265–284.

[4] D. W. Embley, B. Thalheim (Eds.), Handbook of Conceptual Modeling, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-15865-0.

[5] D. Bogdanova, M. Snoeck, CaMeLOT: An educational framework for conceptual data modelling, Information and Software Technology 110 (2019) 92–107.

[6] S. A. Petersen, J. Krogstie, Experiences from Teaching Enterprise Modelling to Students of Information Systems, in: PoEM Short Papers, volume 1023, 2013, pp. 116–125.

[7] F. Lambusch, M. Fellmann, Towards Context-Aware Assistance for Smart Self-Management of Knowledge Workers, in: Proceedings of the BIR 2018, 2018, pp. 1–12. URL: http://ceur-ws.org/Vol-2218/paper1.pdf.

[8] F. Lambusch, M. Fellmann, M. Poppe, O. Weigelt, S. Hein, Personal Productivity Management in the Digital Age: Measures from Research and Use of Conventional Tools, in: N. Gronau, M. Heine, K. Poustcchi, H. Krasnova (Eds.), WI2020 Zentrale Tracks, GITO Verlag, 2020, pp. 632–647. doi:10.30844/wi_2020_f5-lambusch.

[9] M. Kleinmann, C. J. König, Selbst- und Zeitmanagement, volume Band 38 of *Praxis der Personalpsychologie*, Hogrefe, 2018.

[10] C. C. Manz, H. P. Sims, Self-Management as a Substitute for Leadership: A Social Learning Theory Perspective, Academy of Management Review 5 (1980) 361–367.
[11] A. Graf, Selbstmanagementkompetenz in Organisationen stärken: Leistung, Wohlbefinden und Balance als Herausforderung, Springer Fachmedien Wiesbaden, 2019.
[12] T. Jonsson, H. Enquist, Phenomenological Framework for Model Enabled Enterprise Information Systems, in: New Trends in Databases and Information Systems, volume 1064, Springer, 2019, pp. 176–187. doi:10.1007/978-3-030-30278-8_21.