Sven Strickroth

PLATON

Developing a Graphical Lesson Planning System for Prospective Teachers

Suggested citation referring to the original publication:
Education Sciences 9 (2019) 4, 254
DOI https://doi.org/10.3390/educsci9040254
ISSN (online) 2227-7102

education sciences

MDPI

Article

PLATON: Developing a Graphical Lesson Planning System for Prospective Teachers

Sven Strickroth

Department of Computer Science, University of Potsdam, August-Bebel-Str. 89, 14482 Potsdam, Germany; sven.strickroth@uni-potsdam.de; Tel.: +49-331-977-3068

Received: 6 August 2019; Accepted: 2 October 2019; Published: 10 October 2019

Abstract: Lesson planning is both an important and demanding task—especially as part of teacher training. This paper presents the requirements for a lesson planning system and evaluates existing systems regarding these requirements. One major drawback of existing software tools is that most are limited to a text- or form-based representation of the lesson designs. In this article, a new approach with a graphical, time-based representation with (automatic) analyses methods is proposed and the system architecture and domain model are described in detail. The approach is implemented in an interactive, web-based prototype called PLATON, which additionally supports the management of lessons in units as well as the modelling of teacher and student-generated resources. The prototype was evaluated in a study with 61 prospective teachers (bachelor's and master's preservice teachers as well as teacher trainees in post-university teacher training) in Berlin, Germany, with a focus on usability. The results show that this approach proofed usable for lesson planning and offers positive effects for the perception of time and self-reflection.

Keywords: lesson planning; lesson preparation; support system; automatic feedback

1. Introduction

Teaching is a complex and cognitively demanding process [1,2]. At the same time, it is also a very creative task [3,4]. A lesson needs to be carefully prepared in order to achieve an effective, targeted instruction [5–7]. This is particularly the case for prospective teachers [4,8]. Many aspects such as methods, subject content, standards, and the characteristics of the learner group have to be taken into account and reasonably compiled—often at the same time.

Developing lesson plans is presumably the prevalent form of lesson preparation and is often required in teacher education programs [4,8]. A lesson plan does not only depict the intended lesson flow but also serves as a documentation of the decision making process. Furthermore, the planning process should also help with reflecting the design, weighting options and establishing self-confidence for the actual implementation of the lesson [8].

Traditionally, lesson plans were based on handwritten notes. However, lesson plans nowadays are also often created using standard software such as LibreOffice Writer or Microsoft Word. It stands to reason that this complex process of lesson planning can be supported by specialized software systems that do not only facilitate routine tasks, but also stimulate reflection. Therefore, a graphical time-based representation for lesson planning is proposed in this article, based on the principle of direct manipulation [9], because unrealistic time planning is often regarded as one of the main problems among novices [10–12]. The key contributions of this paper are:

- Clear requirements for a lesson planning system which is customizable and independent of planning models/frameworks
- Proposal for a generic data model for lesson plans

- Graphical approach for planning with a focus on time providing different views, analysis functions and automatic feedback
- Evaluation which shows the usefulness and usability of the developed prototype in real world planning scenarios

The remainder of this article is structured as follows: first, general characteristics of lesson plans are outlined. Second, requirements for a lesson planning support system are derived systematically and existing systems are analyzed regarding the requirements. Third, the proposed approach, the data model and architecture of the prototype are described. The article concludes with a presentation of the evaluation and a discussion of the results.

2. Lesson Plans

As already stated in the introduction, developing a lesson plan is considered as a prerequisite for effective, targeted instruction and a lot of details and different aspects have to be considered at the same time. Hence, various (theoretical) models and (non-technical) frameworks were proposed to help teachers plan lessons [6,11,13–18]. These frameworks often introduce a special wording and do not only differentiate the structure of the lesson plan, but also often propose which steps should be taken out in which sequence (at least implicitly). Generally, lesson plans can encompass a single lesson or a sequence of lessons (sometimes also called "unit" or "module"; *unit* is used in the remainder). As already stated, the content also varies, but it often consists of the four following sections (cf. [12] for a broader comparison):

- formal aspects (e.g., name of the teacher, date, school, subject, room),
- description of the context (e.g., teaching intentions, learning outcomes, standards, learner group, previous and future lessons, assessment)
- visualization of the teaching process (e.g., a table containing the teaching steps, used media and the allotted time)
- attachments (e.g., worksheets, blackboard sketches)

Note, however, that not all sections are present in every lesson plan. There is a wide range of the level of details and different naming depending on the country, state, school, and even the preferences of a mentor. In some countries such a plan can be created in approx. 5–25 min [19], approx. 20–50 min [2] or may even last several days for a lesson plan containing very sophisticated explanations (e.g., in Germany for examination lessons). Consequently, a tool supporting lesson planning system must be useable in all such cases and allow planning "traditional" school lessons with all relevant aspects (e.g., worksheets, blackboard drafts).

3. Requirements

In this section, the requirements of a generic lesson planning system are presented. The requirements are split into two parts: the functional requirements and non-functional requirements (cf. [20,21]). The functional requirements (FR) define who uses which functionality (and why) whereas the non-functional requirements (NFR) depict constraints and quality requirements.

The following requirements were drawn from an extensive literature review on lesson planning and existing planning systems as well as interviews conducted with teacher educators, experienced teachers as well as prospective teachers in Berlin, Germany (cf. [12]). The focus lies on prospective teachers as users. There are two main stakeholders: an actively planning (student) teacher and mentors or other (student) teachers such as peers. Other stakeholders, such as administrators, who set up or maintain a system, or researchers, who use a system for investigations, are excluded.

Table 1. Overview of the functional requirements.

No.	Functional Requirement	Short Motivation
FR1	Personal workspace	Lesson planning is a process which is undertaken by (usually) one teacher for a specific learner group. As a draft may include personal information about specific students and, therefore, should not be publicly available (cf. NFR7).
FR2	Adaptability and customizability of the system	A lesson plan can or should include different aspects—depending on the underlying model or specifications of the mentor (cf. Section 2). Moreover, many teachers have a clear opinion on how their drafts and plans should look like. Therefore, a planning tool needs to be customizable in order to fulfill these different requirements.
FR3	Management of units	In practice, classes are not typically planned and conducted in isolated lessons, but in units which consist of multiple connected lessons. Usually not all lessons of a unit are planned at once but iteratively. Especially for more advanced teachers, re-usage of already existing plans plays an important role.
FR4	Planning/ editing of lessons	After the design of the unit, the next step is to plan the associated lessons in more detail. Here, the goals of a lesson are determined before designing the detailed sequence of teaching activities (cf. FR5).
FR5	Planning teaching activities	A lesson is usually structured into several distinct teaching activities or phases. These, in turn, need to be described, e.g., in terms of their title, duration, or social form. Oftentimes, teaching activities are sequenced linearly, however, for internal differentiation and group tasks they can also occur in parallel (or might be modelled as one activity with two nested group tasks). As the lesson plan is steadily reflected on, revised and complemented by the teachers, adjustment procedures need to be designed intuitively.
FR6	Standards/ Curriculum	Teachers have to obey curricular standards while often in need of justification, e.g., by citing those standards.
FR7	Resource management	Resources such as textbooks, worksheets, presentations, or drafts for blackboard drawings are elementary parts of lessons and therefore needs to be taken into account during planning. They are closely linked to the context such as learning group, goals, and current teaching activity. In addition, materials or blackboard drawings may also be relevant at multiple distinct teaching activities or in following lessons and should therefore also be referenceable there. This also helps reflecting on which resources are needed at a certain time and the management of those (in class).
FR8	Anticipated student solutions	Not only do teachers develop and share resources, but also learners are often encouraged to write notes, texts, draw pictures, create posters, or edit worksheets. This might happen directly in class or in the form of homework assignments. These types of resources (learning solutions) are often used as a basis for following lessons (e.g., discussions, presentations or comparisons) and should therefore also be anticipated while planning.
FR9	Providing different views	To promote reflection on the plannings, it is helpful to highlight different aspects and provide dedicated overviews with different levels of abstraction and specialized representations (cf. [22,23]).
FR10	Statistical analyses	Diversity of methods is often seen as a quality feature for high quality teaching [24,25]. To uncover possible imbalances in e.g., social forms, it is helpful to provide a statistical analysis.
FR11	Automatic feedback	Mentors are not always available to give timely feedback. Algorithmically generated feedback can fill this gap by providing hints 24/7 and, thus, stimulate reflection.
FR12	Notes and comments	The planning process is often carried out iteratively and thereby refined several times. Therefore, certain aspects might remain "open" at an early stage but need to be further elaborated at a later time. Comments, however, are not only important in the planning phase but can also be used to reflect the planning and and documenting the experiences following the implementation of a lesson.
FR13	Sharing of plans	Sharing allows a mentor to view the most recent version of the plan. Many teachers find exchanging their designs helpful in order to receive feedback or as a source of inspiration for their own plans (cf. [26,27]).
FR14	Printing	Teachers are often asked to submit their final lesson plan on paper to a supervisor before implementing a lesson plan. In addition, the plan may serve as an orientation during the implementation of the lesson plan.
FR15	Export	One the one hand, more and more classrooms are equipped with computers (for teachers). Therefore, exporting a complete lesson plan that can be used directly in class would be helpful. On the other hand, personal backups need to be possible.

An overview of the functional requirements is presented in Table 1. The requirements are uniquely numbered and referenced using the number in the remainder of this article. Categorization is possible as follows: First, there are quite general requirements for a personal workspace (FR1) and an adaptable system (FR2). Second, there are requirements which concentrate on the planning/management of units (FR3), lessons (FR4), and teaching activities (FR5). Then, there are requirements covering the integration of resources, such as curricular standards (FR6), and the management of teacher developed (FR7) as well as student developed resources (FR8). As the system should not just enable users to plan lessons, it should also provide specific support features such as different views (FR9), statistical analyses (FR10), and automatic feedback (FR11) in order to foster reflection on the plan. Also, the system needs to provide a way to add comments while planning or after conducting a lesson (FR12). Sharing a lesson plan (FR13) with other users should be possible, too. Finally, it is important that the developed lesson plan can be used outside of the system either offline (i.e., by printing it, FR14) or even with other tools (i.e., by exporting it, FR15).

The non-functional requirements are presented in Table 2. Key points for being usable in real world scenarios are on the one hand user-centered, such as a high usability (NFR1) and a neutral naming (NFR6). On the other hand, there are technical aspects such as interoperability (NFR2), performance & scalability (NFR3), and security (NFR7). Additionally, it would be good if a system can be extended easily (NFR4) and can be freely used by third parties (NFR5).

Table 2. Overview of the non-functional requirements.

No.	Non-Functional Requirement	Short Motivation
NFR1	Usability	The user interface (concept, complexity, appearance, ...) and its behavior (expectation conformity, controllability, fault tolerance, ...) have a decisive influence on how a system is perceived and used [28]. This is important as the system should be used by non-tech-savvy users.
NFR2	Interoperability	The software must be usable on various operating systems and platforms (even mobile devices, as those are often used for quick lookups and recalling the planning; cf. [29]). In addition, the system must be usable if users do not have administrator permissions (i.e., in schools).
NFR3	Performance & Scalability	For practical use, especially on mobile devices, it is important that both the initial loading and the starting time as well as the response time, especially for frequently used functions, are as low as possible. Also, scalability plays a special role for (open) field studies or a broad productive use when the system has to stay usable with increasing number of users (e.g., by adding resources).
NFR4	Customizability/ Extendability	Research lives from trying out and evaluating new, innovative ideas. To avoid having to design a completely new system for each approach, it would be easier to adapt an existing system.
NFR5	Openness	To pass on the system to other scientists and to make it more widely applicable in practice, it is necessary not to use any tools, compilers or libraries for which license fees are charged.
NFR6	Neutral naming	As described in Section 2, there are various planning templates/frameworks available. Almost every one of these has its own terminologies, theories or concepts. Special terms should be avoided as they might imply a specific concept (cf. [18]). This is not a theoretical problem and has already been discussed for planning systems in [30,31]. However, the selection of alternative terms should be done with great care so as not to create confusion or other hurdles (see "The Vocabulary Problem", cf. [32]). Even if the system uses neutral terms, it should also be usable with specific didactics using its terminology (see FR2, adaptability and customizability of the system).
NFR7	Security	Lesson plans often include personal and private data, which needs to be protected. This applies in particular if a (research) prototype is to be offered or used productively on the internet.

A compiled list of clear criterions for these requirements can be found in Appendix A.

4. Existing Lesson Planning Tools

Since lesson planning is an important as well as a challenging task, a variety of different tools and approaches have been proposed and developed. The review is, however, limited to tools which were published scientifically, focus on (prospective) teachers as planners, and are used for planning traditional lessons. Simple generators (such as [33]), which consist of a simple HTML form, are not considered here.

For the development of IMS Learning Designs (IMS LD) there are a variety of editors available (cf. [34–36]). Especially the older tools are closely designed to the IMS LD specification and are usable only by Learning Design experts [37]. Usually teachers are no Learning Design experts. More recent Learning Design tools (e.g., [37,38]) and tools for computer-supported collaborative learning (CSCL) scripts (e.g., [39,40]) were also developed for teachers, however, those focus on blended learning or computer-based scenarios and, thus, are not suitable for planning traditional lessons [36,41,42]. Therefore, most of these tools were excluded, too.

Considered here are the following tools: Lesson Planning System (LPS, [43]), Support for Teachers Enhancing Performance in Schools (STEPS, [44]), Instructional Planning Assistant System (IPAS, [45]), TaskStream (TS, [46]), Eduwiki (EW, [26]), Inquiry in Motion Dynamic Lesson Planning Tool (IIM, [47]), Smart Lesson Planning System (SLP, [19]), Learning Activity Management System (LAMS, [48,49]), Phoebe Pedagogic Planner (Ph., [50]), and The Learning Designer (LDSE, [51]) as well as the newer web-based version (LDw, [52]).

There are major differences between (1) the goals which led to the development of the existing lesson planning tools, (2) the context of usage (mostly based on special requirements of the country for which a tool was developed), and (3) the used technologies. The following goals could be identified:

- reduction of the cognitive load, e.g., by providing structures/templates (e.g., IPAS, IIM, SLP)
- integration of helpful resources such as standards (e.g., STEPS, IPAS, Ph.)
- easy re-usage of existing lesson plans (e.g., EW, IPAS, IIM, SLP, LDw)
- support for the development of worksheets (e.g., IPAS)
- collaboration (e.g., EW)
- fostering reflection (e.g., based on visualizations (LAMS, LDSE, LDw) or checklists (IPAS))
- detailed modeling of computer-supported group interactions (e.g., LAMS)
- computer-supported execution of the plans/models (e.g., LAMS)
- provision of activity patterns, which can be directly used (e.g., LDSE)

Table 3 shows a comparison of the requirements of the previous section with the features of existing tools. Conspicuously only very few requirements are fully met. This specifically concerns FR1 (personal workspace) which is at least partially met by all systems. The exceptions are LPS and LDSE, which are, in contrast to the other systems, standalone desktop applications and no multi-user systems. Consequently, these two systems do not support shares (FR13) offered by the other web-based systems.

Table 3. Overview of the fulfillment of the functional requirements of Section 3 (+ = all criterions fulfilled, (+) = some fulfilled, (−) = no criterion fulfilled, but comparable feature available, − = no criterion fulfilled, ? = unknown).

Requirements	LPS	STEPS	IPAS	TS	EW	IIM	SLP	LAMS	Ph.	LDSE	LDw
FR1: Pers. workspace	(+)	+	+	+	+	+	+	+	+	(+)	+
FR2: Customizability	–	–	–	–	(–)	(+)	–	(+)	+	–	–
FR3: Sequences	–	(+)	(+)	(+)	– [a]	– [a]	– [a]	– [a]	(+)	(+)	– [a]
FR4: Lessons	(+)	(+)	(+)	(+)	(–)	(+)	(+)	(+)	(+)	(+)	(+)
FR5: Activity planning	(+)	?	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)	(+)
FR6: Standards	–	(+)	(+)	(+)	–	(+)	(+)	–	–	–	–
FR7: Resource mgmnt.	–	–	(+)	–	(+)	(+)	–	(+)	–	(+)	(+)
FR8: Anticip. solut.	–	–	–	–	–	–	–	–	–	–	–
FR9: Views	–	–	–	(+)	–	–	–	–	–	(+)	(+)
FR10: Statistical eval.	–	–	– [b]	–	–	–	–	–	–	(+)	(+)
FR11: Automatic FB	–	–	–	–	–	–	–	–	–	–	–
FR12: Comments	(+)	?	(+)	–	(+)	–	–	–	–	–	+
FR13: Sharing	–	–	+	?	(+)	+	+	+	+	–	+
FR14: Printing	(+)	(+)	?	?	(+)	(+)	(+)	–	(+)	–	(+)
FR15: Export	?	–	?	?	?	–	?	(–)	(–)	(–)	(–)

[a] No explicit support for units, partly lessons can be categorized; lessons can be cloned. [b] There are statistical evaluations, however, only for the design of curriculumns.

Except for Phoebe and Eduwiki, all tools come with predefined fields that must be used for planning (cf. FR2). The IIM system is even bound to a specific planning model, the 4Ex2-Instructional-Model [17], and thus cannot be used for generic lesson planning. With regard to the input options, many tools rely on simple text input (LPS, IPAS, SLP, LDSE, LDw) and often even do not provide a What-You-See-Is-What-You-Get (WYSIWYG) editor, so that the formatting options are extremely restricted.

It is surprising that only half of the systems allow combining individual lessons into units (FR3). However, this only serves as a structure; there are no ways to explicitly model transitions between lessons of a unit (e.g., homework). In addition, re-ordering or copying lessons is often not possible. The other tools only allow the planning of independent lessons. Cloning lessons is mostly possible within the system. Based on the criteria for selecting the systems in the review, all tools can be used to describe lessons (FR4)—Eduwiki is marked with "(−)" here due to a lack of clarity on whether there is a template provided in the wiki or not. However, no tool provides a way to access objectives of the unit from a lesson without changing views.

Regarding lesson (activity) planning (FR5), it is noteworthy that the majority of the systems is limited to a purely textual description (this can potentially limit the creativity, cf. [4]): Simple input fields are provided either as a large one in which the entire process can be entered textually, or as several small fields; one for each teaching activity/phase (sometimes even with a fixed number of fields, e.g., five in SLP). As already mentioned, WYSIWYG editors are not always available. In addition to Eduwiki IIM, LAMS, LDSE, and LDw stand out: In IIM a lesson is planned in fixed predefined blocks according to the 4Ex2-instructional model. LAMS, LDSE, and LDw all have a graphical representation. LAMS allows modeling a lesson as a sequence of predefined activities similar to Unified Modelling Language (UML) activity diagrams with the aim to directly execute the plan with the aid of computers. A special feature of LAMS is the preview mode, which allows going through the lesson step by step during planning. LDSE and LDw display activities as blocks on a timeline, however, in LDw the durations are not visualized and parallelism is restricted. Groupings (i.e., handling groups of students) are only possible with LAMS and partly with LDSE and LDw. Temporal aspects (such as the duration of individual activities) are not directly visible or require a lot of effort (excluding LDSE). LDSE and LDw, however, do neither allow grouping activities, moving activities to different lessons, nor map the

length of the timeline to the length of a lesson. A characteristic disadvantage of the Learning Design tools is that physical resources cannot be modeled directly, workarounds are required [36].

Half of the tools have integrated educational standards, competences or a framework curriculum (FR6). Linking is, however, limited to worksheets (IPAS) or lessons, but not to a unit.

A sophisticated resource management (FR7) which allows re-using materials is not provided by any system. There are, however, systems such as IIM, LAMS, LDSE, and LDw, which allow attaching resources to individual activities. Nevertheless, especially in the case of Learning Design tools (LAMS, Pheobe, LDSE, LDw), there is the restriction that only digital or predefined resources can be modeled [36]. Likewise, no system explicitly requests the modeling of anticipated learner solutions (FR8). Noteworthy, however, is IIM where learner-created solutions can be uploaded to the platform as examples for other teachers.

Different views (FR9) on the lesson plan can be found only in TaskStream and LDSE/LDw: TaskStream provides learning outcome and competence matrices with which goals can be clearly assigned to individual lessons. LDSE provides three views: (1) one for entering goals, properties and descriptions, (2) a timeline, and (3) an analysis view; LDw combines the views (1) and (2). Other graphical representations, in which resources play a central role (e.g., for modeling a resource flow, cf. [53]) exist, but are not available in tools. Statistical evaluations of lesson plans (FR10) can be found in a basic variant in the analysis view of LDSE, which can evaluate certain given properties of the activities. IPAS provides self-assessment checklists, which turned out to be not very helpful [45]. More in-depth support, such as automatic feedback (FR11), is not implemented by any system so far. An approach tried to give context sensitive advices or raise specific questions while planning in order to stimulate self-reflection [54]. However, no quality improvements were found [54].

A note function is present in four systems (LPS, IPAS, EduWiki, and LDw, FR12), but no highlighting is possible and often notes are always visible to all users.

Although a print function (FR14) is available in the majority of systems, it cannot be customized. An export function (FR15) which contains the entire plan including resources and can also be used without the system, is not implemented by any system.

Overall, comparing the requirements with the features of the systems reveals that no system meets all requirements. Thus, no system can be recommended for classroom planning. Most of the existing systems are limited to purely textual planning, where temporal aspects are not obvious. Some systems have noteworthy approaches, such as the graphical representation of LDSE, various views as in TaskStream, LDSE and LDw, and custom templates as in Phoebe. Nevertheless, important features, such as the management of units (FR3), a sophisticated resource management (FR7), and modeling of anticipated learner solutions (FR8) are not or not fully implemented. In addition, reflections upon different views and evaluations (FR9 and FR10) are not yet fully exploited. The provision of automatic hints (FR11) cannot be found in any single system. Also, there are very few publications which contain requirements, a data model or a detailed system architecture.

5. The PLATON System

The approach presented in this paper addresses the research gap of missing computer-aided

- graphical, time-based planning based on a timeline metaphor,
- specification of extensible structures (including customizable fields),
- explicit modeling of anticipated solutions and expectations on materials created by students,
- connecting individual lessons of a unit, especially by explicitly modeling of transitions between two lessons, and
- provision of different views (e.g., resource or multi-lesson view, fulfilled standards of a unit/lesson) as well as support functions to reflect on the planning (e.g., visualizations, statistical evaluations of the social forms used in a lesson/unit, and automatic feedback).

The main goal of the PLATON system is to stimulate self-reflection and to support a deeper understanding of a planned lesson. This happens on two levels: First, a helpful representation, different views as well as analysis functions are provided, but those depend on meta-cognitive skills of the planners. Second, the system also provides concrete automatic feedback.

As already outlined in Section 4, there are only two approaches for graphical lesson planning and exactly one with a focus on time. However, both do not fulfill the requirements. Generally, graphical representations have already proven to be useful in many ways: reduction of the cognitive load, more systematic task solving, considering different solutions, and representations can make the problem space better or worse accessible to the learner [22,55]. Also, it could be shown that different representations can promote a deeper understanding [23]. However, a graphical representation might also restrict users in case they cannot express their ideas with it [22].

Applying this to lesson planning, the proposed timeline can help to ensure that activities do not unconsciously overlap (graphical constraining) or that a prospective teacher becomes aware about the end of a lesson, which activities take place in parallel, or how long individual activities or phases are relative to each other. Questions such as "Is a lesson fully scheduled?", "Is the plan within the given time feasible?", "How much time is scheduled for an activity—is that realistic?", "How do the lengths of activities relate to each other?" or "Which activities run in parallel?" are very difficult to answer with tables or abstract numbers in textual descriptions. The proposed approach makes it possible to get an overview of the central aspects of the course of a lesson respective a unit quickly and to answer these questions. In particular, parallelism of activities must not be understood as the simultaneous execution of activities with the same start and end times (e.g., group work). Rather, parallel activities with different start times can be used to model internal differentiation (e.g., a task is planned for the entire class, whereby another task is planned in parallel for "better" students, cf. Section 5.4, these are difficult to plan textually). The approach presented here attempts to solve this gap through a timeline-based view of the activities—without restricting the capabilities of (textual) descriptions of the individual activities.

Other aspects of this approach include fine-grained modeling of resources and lessons. While existing systems allow attaching files (such as worksheets, materials, or blackboard drafts) or to insert hyperlinks into drafts, there is no sophisticated resource management. However, these types are always teacher-created resources, and an overview of the reuse (e.g., a particular book is used for several lessons) is not provided in other tools. In particular, no tool can model student-created resources and a resource flow (a resource is created as a homework assignment and serves as an "input" for reviewing homework and/or editing a follow-up task). The ability to model this can lead to a deeper understanding of the individual activity (because when explicitly modelling a student-creating resource, a planner can say exactly what s/he expects as the outcome).

The proposed PLATON system fulfills all the requirements listed in Section 3 and is described in more detail in the following subsections.

5.1. Data Model

Figure 1 depicts the complete data model (i.e., ontology of the domain). The UML class diagram is structured in packages as follows: At the top are the learning `Standards` and `Users`. Below there is the `ContentStructures` package, which contains the building blocks of the lesson plans (units, lessons and activities). At the bottom, there are the packages for `Resources` and `CustomFields`.

First, there are the registered `Users` of the system. For every `User` the unique `username`, a hashed `password`, an `email` and an `isAdmin` flag for granting administrative privileges are stored. Additionally, there is a `User`-specific key-value store (`UserSetting`).

Each unit (`Sequence`) has a unique `ID`, is created/owned by a `User` and has attributes for meta-data (`title`, `subject`, `grade`). Additionally, there are predefined typical attributes, such as the learning group description (`preknowledge`) and the expected learning results (`goal`). To handle other user

specific attributes a `Sequence` can be linked to a (`CustomField`) and to learning standards (`Standards`). Finally, the owning `User` can share a `Sequence` with other `Users`.

A `Sequence` consists of several `Lessons`, individually ordered using the attribute `order`. For each lesson the `length` (in minutes), expected learning results (`goal`), `Homework` (including materials as `Resource`), `comments` as well as anticipated difficulties and solution strategies can be stored (`AnticipatedIssue`).

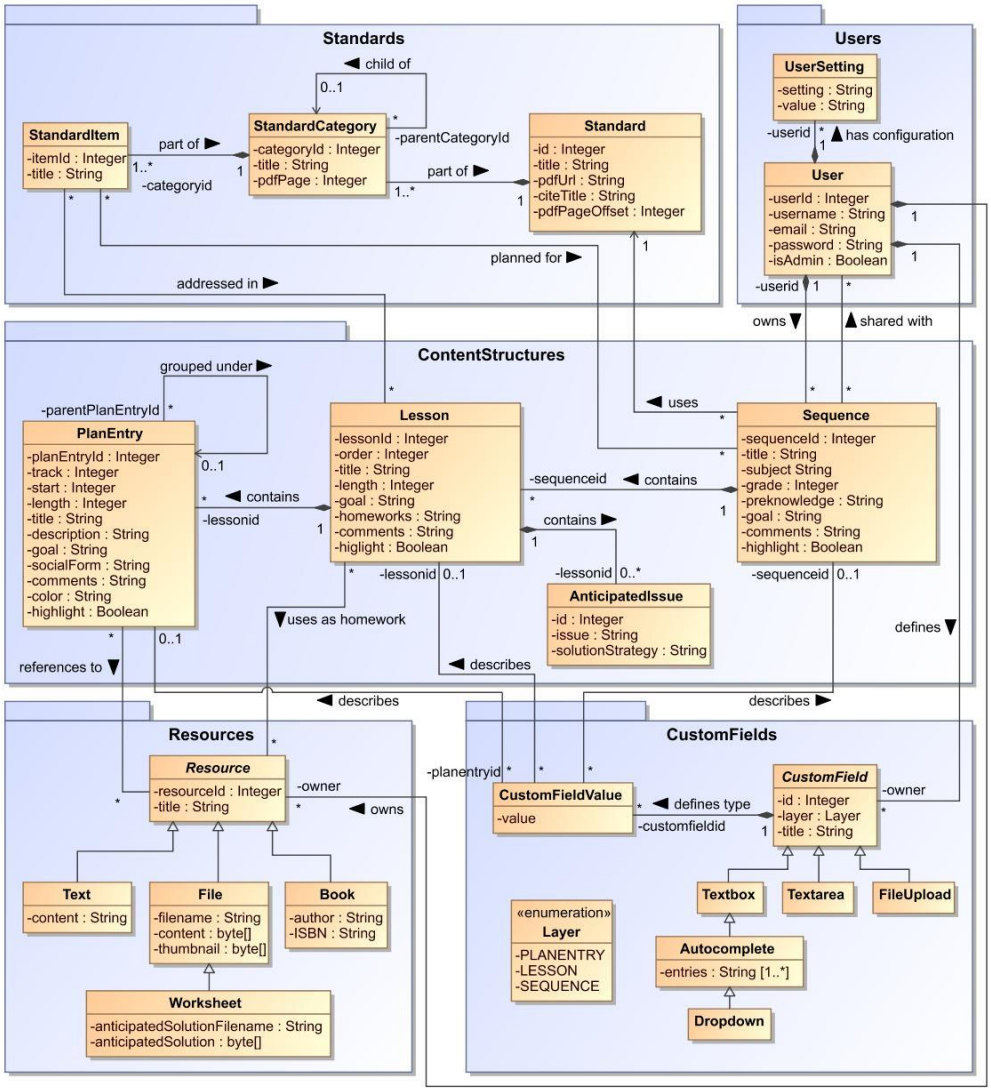


Figure 1. Data model as UML class diagram.

`PlanEntrys` represent most detailed entities, the planned activities (aka. "phases" or "(teaching) steps"). Besides having an `ID`, a `title` and a reference to a `Lesson`, each `PlanEntry` contains information about the `length` (in minutes), the `starting` time (also in minutes) when they are scheduled within the lesson, and a `color`. To model parallel activities, they have a definition of the `track`/row they have on the time stream (0 = first track). Nested activities are indicated with an optional `parent PlanEntry`. Further attributes are the `description`, expected learning results (`goal`), and a `SocialForm`. Similar to `Sequences` and `Lessons`, there is a comment as well as `highlight` flag. In addition, `PlanEntrys` can be linked to `Resources`.

The `Sequences`, `Lessons`, and `PlanEntrys` build a tree with basically three levels. Thus, existing analysis patterns, such as Fowler's plan pattern [56] for project planning which are optimized for critical path analysis and resource allocation, are not applicable.

The classes `Standard`, `StandardCategory`, and `StandardItem` represent educational standards. These three classes are also structured as a tree (cf. Composite Pattern [57]): A `Standard` corresponds to the root, `StandardsCategory` to inner nodes (i.e., chapters and sub-chapters) and the `StandardItem` to the leaves (i.e., specific standards, competencies, ...). A `Standard` has a `title`, a special `citeTitle` that is used for reference lists and can hold a `pdfurl` whereas `pdfPageOffset` can be used for cases where the page numbers of the printed and PDF version differ. Finally, `StandardItems` can be referenced from `Sequences` and `Lessons`.

Another central entity is `Resource`. Each `Resource` has an `ID` as well as a `title`, is owned by a `User` and can be referenced by any number of `Lessons` or `PlanEntrys`. A `Resource` can be one of four types: a textual description (`text`), a book (`book` with an `author` and an `ISBN`), a normal `file` (with `filename`, file `content` and a preview image) or a `worksheet`. The latter is a specialization of a `file`, which may also contain an anticipated solution.

In addition to the above-mentioned typical description attributes for `Sequences`, `Lessons`, and `PlanEntrys`, `Users` can also define custom fields (see FR2). For this purpose, the `CustomFields`, which have an `ID` and a `title`, can be assigned to one of the three planning levels (`Sequence`, `Lesson` or `PlanEntry`) where a field should be displayed. `CustomFields` can occur in one of five versions: (1) single-line input field (as a normal `textbox`, as an `Autocomplete` text field, or as a `dropdown` with strict selection options), (2) a multi-line WYSIWYG editor (`Textarea`), or (3) a `FileUpload`. The contents/values are stored as `CustomFieldValues`. These include the `value`, a reference to the corresponding `CustomField` definition (cf. Item-Description-Pattern [58]) and a reference to either a `Sequence`, a `Lesson` or a `PlanEntry` where the custom field is used.

5.2. Architecture

In this section, the overall architecture of the PLATON system and the client are described.

5.2.1. Overall Architecture

The overall architecture is an adaption of a traditional three-tier architecture (presentation, logic, and data tier, cf. Figure 2). Every tier only communicates with the directly adjacent tiers using well-defined interfaces. This design was chosen based on the non-functional requirements so that on the one hand any of the three tiers can be upgraded or replaced independently. On the other hand, all data is located at a central location and access control can be implemented easily.

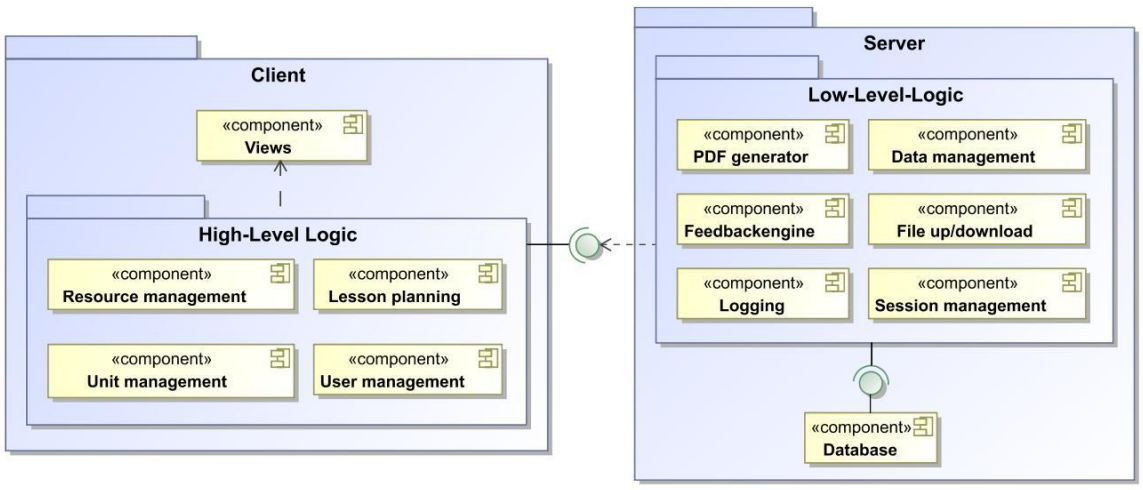


Figure 2. Proposed architecture.

To guarantee a high scalability, significant parts of the logic are processed on the clients (such as smart resp. rich client approaches, cf. NFR3). This distribution reduces the load on the server as well as network latencies and ensures quick response times, because most user interactions can be handled

locally. On the client side, there are the components such as `Lesson planning` or `User management` located which directly relate to the graphical user interface or implement basic high-level functions (cf. Figure 2). The server provides a set of low-level services which are accessed by the client after successful authentication: These services are grouped in components for data management (based on CRUD operations; Create, Read, Update and Delete), the server side part of the feedback engine, a PDF generator, logging, session management (login, logout, ...) and file up/download.

The `data management` component acts as a so-called data mapper [59]. It mediates between the application and the database, capsules the database API, and provides basic methods for data manipulation (cf. service layer [59]). This component can be accesses directly by the low-level components. The high-level components on the client-side employ a remote procedure call mechanism (RPC) in order to interact with the `data management` component. To achieve a loose coupling without distributed objects and to reduce the number of data requests, data transfer objects are used [59].

5.2.2. Client Architecture

The central element of the client architecture is an `Application Controller` [59]. This controller drives the interaction between different components/views. The components use the Passive View pattern, which is based on the common Model-View-Presenter (MVP) pattern for splitting layout, business logic, and data [60]. Main advantage of the MVP pattern compared to the Model-View-Controller pattern is that the view neither does depend on the model nor includes any logic for state transitions. Therefore, the logic and the view can be implemented and tested independently.

To ensure a loose coupling between the components of the client, the components must not need to know about each other. To show the up to date data in different views, an event bus was introduced. This design allows components to subscribe to a set of messages and to react on changes. This way components can communicate even to components which they do not know or which did not exist at the time of their development. This way, components only need to know specific components in case they directly link to them in the view—the transition itself is performed by the application controller.

5.3. Implementation of the PLATON System

Based on the requirements NFR2 and NFR4 the client of the PLATON system was implemented as a rich-internet application (RIA, cf. [61]) using the open source GWT framework (version 2.8, [62]). The GWT framework provides a set of predefined graphical user interface (GUI) elements so-called widgets, layouts, AJAX based type-safe remote-procedure calls over HTTP, and an API for various web technologies. Programming takes place in Java and is then compiled to JavaScript code. The resulting JavaScript code "automagically" supports all widespread browsers. GWT is also extendable by third-party modules. PLATON uses the dual-licensed Sencha GXT framework 4.0 [63], which provides additional LayoutManagers and widgets such as interactive diagrams.

Influenced by the decision to use GWT for the client-side, the server-side was also implemented in Java. This has the advantage that the very same programming language is used as for the client and type safe RPCs (as provided by GWT) can be implemented. Furthermore, the server-side is also platform-independent and only requires a Java Servlet specification 3.0 [64] compatible web server such as the chosen open source Apache Tomcat (version 8, [65]).

The PDF generator component requires a tool for the conversion of HTML documents to PDF. Such a tool is needed because on the client side HTML-based WYSIWYG editors are used and additional (potential lossy) conversions should be avoided. The prototype uses the open source software PhantomJS [66], a scriptable headless WebKit "browser", for this task.

For data persistence the open source relational database management system MariaDB [67] was used for the prototype. The usage of an object-relative mapping (ORM) framework was explicitly ruled out, because their common advantages such as lazy loading and implicit database updates do not come into play for RIAs as the ORM objects can only be used on the server side.

The underlying source code is licensed under AGPLv3 [68] and can be retrieved online [69]. A demo system that can be freely used and tested is also available.

5.4. The Graphical User Interface

After logging in the user is directed to his own personal workspace. Here, the GUI is divided into two parts: On the left side there are the created units resp. uploaded resources shown. It is possible to switch between a planning and resource management view.

The planning view is the default view after login. In this view on the left side there is an overview over all units which are created by the current user or shared with him. New units can be created; existing units can be compared and duplicated. As already described in Section 5.1, PLATON is hierarchically organized whereas the units are the root elements. For each unit it is possible to enter descriptions into predefined textfields (normal text fields, e.g., for the title, or WYSIWYG editors, e.g., for learning goals; the data model in Section 5.1 shows all predefined fields). Additionally, standards can be selected and there is a generic notes field. If users think that there is something missing in the predefined form, they can create their own custom fields to enter other data in a structured way. Within a unit, lessons can be created. Required fields for a lesson are a title and a duration (e.g., 45 min). In case a unit consists of more than one lesson, homework can be entered (by using a simple WYSIWYG textfield for the task and goal descriptions as well as optionally uploaded files).

In PLATON a lesson is visualized as a timeline (incl. explicit indication of the allotted time). This view is the heart of the PLATON system. Different activities (called "actions" in PLATON) of a lesson are represented as boxes on the timeline which can be moved on a grid by simple drag'n'drop (horizontal and vertical). The width of the boxes correspond to their intended/scheduled duration (cf. Figure 3). Each box consists of a header and a body. In the header, the title of an action is shown and the body can be filled dynamically with any other attribute; by default there are small icons displayed which indicate which fields were filled (e.g., a ribbon for learning goals or a paper clip for each linked resource). Detailed properties (cf. data model) for an action can be entered below the timeline (cf. Figure 3). There it is also possible to link resources to an action and to see all linked resources. Actions cannot only be placed sequentially one after another, but also in parallel or nested (moving boxes on top of each other creates a grouping that can be resolved by pulling out a nested box). Nesting is required e.g., for modeling group activities where two subgroups work simultaneously. Parallel activities cannot only be used for group work, but also allow modeling possible internal differentiations. Nesting is limited to one level (cf. Figure 3). An early prototype supported infinite nesting, where nesting automatically created a sub plan. This, however, turned out to be problematic regarding overview and handling. It is most important that a planner gets a quick overview over the whole lesson at once and can make changes easily (e.g., by moving activities around) without having to change the view (to a sub plan).

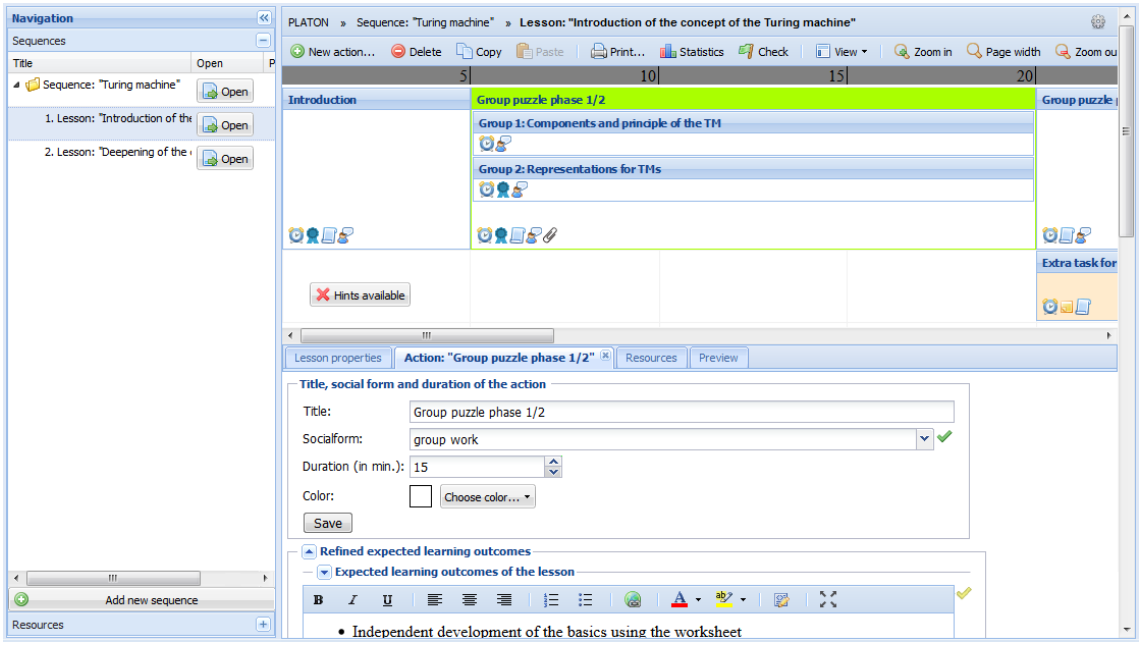


Figure 3. Screenshot of the GUI of the PLATON system.

PLATON supports four types of resources: blackboard sketch/presentation, worksheet, student created, book, and "other". In case of a blackboard sketch and a worksheet, arbitrary files can be uploaded. For books a special form is provided which asks for title, author and optionally an ISBN (a book cover is automatically downloaded). In the other two cases, there are textboxes for the description of the resource or the expected results of the student created resources. Resources can be linked to lessons (as homework resources) or to actions. Multiple links to a resource are possible.

The resource view provides an overview of all resources and allows accessing them independent of any usage. For each resource the title, a preview image and all links to lessons and activities are displayed. Hence, users cannot only see whether a resource is used anywhere or a student-created resource is only used exactly once (i.e., might indicate a missing discussion of the results), but also the context and pedagogic goal in which the resource is used. This is an advantage to paper-based lesson plans, where this information is not directly visible and must be gathered manually.

Additionally PLATON supports several statistical analyses (e.g., used social forms), different views (e.g., resource overview, standards view) and automatic feedback [12,70,71].

6. Evaluation of the PLATON Prototype

The main focus was to evaluate the comprehensibility as well as the usability of the system and the graphical representation in real world planning scenarios within teacher education (in Germany, cf. [72]). The evaluation took place in Berlin, Germany, either at Humboldt-University (called "students" in the remainder) or within the "Vorbereitungsdienst" (i.e., the German post-university teacher training, called "trainees"). In total 61 prospective teachers were involved as participants.

Participant selection and a wide task coverage are crucial for usability tests (cf. [73]). Therefore, two dimensions were considered: The "teaching experience" (i.e., courses at bachelor's and master's levels and on a post-university teacher training) and the "subjects" (in this order: computer science, math, geography, and English) resp. assumed experience with computers of the participants. The rationale behind this fixed order was the assumption that computer science students understand the layered design of the PLATON system more quickly and are more resilient against possible bugs in an early prototype. In the selected courses lesson planning was on the curriculum. The concrete planning tasks were provided by the lecturers. The main research questions in this evaluation were:

- How do the participants rate the graphical representation?

  - Is the graphical representation understandable and meaningful?
  - Is the graphical representation powerful enough to express all possible ideas/plans?
  - Does the graphical representation facilitate reflection?
- Are there fundamental usability issues?
- How do the participants rate the practicability of the tool?
- Which features were used and which ones are of particular importance?

The evaluation was formative and took place in parallel to the development of the system. This way new findings could be directly integrated into the prototype after each intervention. There were two different settings in the study: On the one hand, it were lab studies where the participants synchronously planned a lesson under controlled conditions in a computer pool. During the study, the participants were monitored. On the other hand, it were field studies where the participants got the task for planning a lesson as homework. In both cases, the design of the evaluations followed the typical usability test design [28,74]: Short introduction into the system (5–20 min), working on a given task, then group interviews and individual questionnaires. In all scenarios, the planned lessons were presented and discussed in class. Table 4 gives an overview over the study (parts of the intervention with the bachelor's students have already been published in [70]).

Table 4. Overview over the study consisting of five interventions.

Subject	Computer Science	Computer Science	Math	Geography	English
Experience	Bachelor	Post-University	Master	Master	Bachelor/Master
# participants	10	18	7	11	15
Design	lab	field	lab	lab	field
Duration	120 min.	4–5 weeks	210 min.	150 min.	2 weeks

6.1. General Results

Almost all participants were able to plan lessons using PLATON without a longer introduction to the tool. No fundamental limitations for planning traditional lessons as reported in [36] for Learning Design tools could be observed for PLATON. However, three participants of the English class did not finish a complete lesson. In most cases the planning was done directly in PLATON. Two participants of the English class, however, first planned the whole lesson either on paper or in Microsoft Word and then entered resp. copied the finished plan into PLATON instead of planning directly in PLATON. This behavior of the two students was not expected.

The study showed that PLATON is generally intuitively useable. The tool was often attributed as user-friendly; however, there were also some students who demanded a better navigation in the tool and a longer training period. One participant said that it is possible to use the basic functions of PLATON easily while knowing that "more" is possible; this led to "a quick sense of achievement". An experienced trainee noted, however, that there are many ways for planning a lesson, covering all of them would result in a variety of features he does not need.

In the lab sessions, a few students asked several specific questions on how to use the system without first trying to find the answers themselves. Oftentimes these questions were followed by ideas for new features (which partly were already implemented). From the field study groups there was only one request asking whether it is possible to include the resources in the printout (which was not possible at that time). In the last field study, it was the first and only time that connectivity problems to the PLATON system were reported. As the server is located in the network of the university and no network issues were reported at that time, the connectivity issues must have their origin in the private home networks of the students.

The following subsections first present the qualitative results of the group interviews, followed by the quantitative results of the questionnaires.

6.2. The Graphical Representation

During the interviews, the graphical representation was mostly described as clear. In all settings, it was also reported that PLATON offers a good overview over a lesson. Here, the multi lesson plan was explicitly named as it provides a good overview over the whole unit (e.g., whether it is "diversified enough"). The general structure (units, lessons and teaching activities) was characterized as adequate. There were no issues reported regarding reading, presenting, or discussing a lesson plan directly in PLATON or on the generated printout.

The visualization of the lesson plan was generally assessed positively: Very often "creates clarity", "materials at a glance", "time overview", and "quick overview" were mentioned as an advantage of PLATON. The students also agreed that it was helpful to see how the lesson "fills up" (i.e., see the progress of the planning) and how different teaching activities relate to each other. Repeatedly participants reported that time was more present compared to other ways of planning—one student noted, however, that this is at the expense of a larger display space requirement of the representation. It was explicitly stated that allotting more time than available must be possible with a tool. It was also noted that the overview helped to access whether too much teacher activity was planned. Generally, PLATON has often been described as flexible: (1) It allows planning a lesson at a high level and also very detailed and (2) participants of all experience levels reported that a lesson plan can be easily adapted as well as the integration of other media is uncomplicated. The color-coding and the automatic calculation of the "scheduled" minutes were also found to be very practical by several people. However, an experienced student teacher meant that the created lessons all look quite similar.

All participants rated the selection of the predefined textboxes as sensible. Also, the dropdown boxes with predefined values were considered helpful. The participants also stated that it is good to know, that they are not bound to the predefined ones and can define custom input boxes. However, the custom fields were only used significantly by the Geography and English classes; although all participants knew how to define them. Regarding the size of the textboxes, one participant pointed out that they might be too small by default for a longer argumentation (textboxes can be maximized in PLATON).

There were no reports of aspects or ideas that could not be planned with PLATON. However, the students of the English class missed a way for the explicit juxtaposition of teacher and student activities in the detailed description of a teaching activity (some students created custom fields for this) and an experienced student teacher missed the option to create tables. Both aspects are due to the used HTML editor of the prototype, which has no support for text alignment and tables.

The bachelor's students said in the group interview that the tool provides a guideline that helps structuring the planning process. An experienced career changer in the post-university teacher training with a minor in mathematics saw too much pre-structuring as a problem and asked on the questionnaire whether PLATON was intended to be interdisciplinary. Other trainees attested PLATON a high degree of flexibility with regard to the predefined structure. Participants of all levels reported that the predefined fields in PLATON served as an inspiration (e.g., some students said they often forgot to formulate anticipated issues or to develop an anticipated solution when uploading a worksheet). However, it was also brought up that planning using a specialized tool such as PLATON could lead to less creative lesson plans, because planning might just follow a "checklist".

PLATON was often compared to Microsoft Word resp. LibreOffice Writer by master's students and trainees: One participant reported that the tool is "super structured", but Microsoft Word provides a better overview. Some students said that planning in Word "feels nicer". Other participants preferred PLATON as it provides a better overview over the lesson and the plan is more "compressed". In return, two participants criticized the need to click around a lot in PLATON in order to enter the contents and an experienced trainee noted that most details of the lesson plan are hidden "behind" the boxes and must be explicitly opened to see them. Also, one student of the English class explicitly stated that she would go on planning with pen and paper as this is faster and results in a better outline.

Generally, it was stated that no significant changes were noticed while planning: Some participants (also from beginners in the post-university training) expressed the feeling that they (1) were planning more aspects simultaneously than in their traditional planning and (2) started thinking about specific aspects just because "there is a field for it". When asked about self-reflection, the bachelor's and master's students as well as one experienced trainee stated that they reflected upon the allotted time (as time is more present) and on which aspects to mention in the lesson plan. In particular, a bachelor's student reported that it was the first explicit discussion of the allotted time while planning. On the one hand, an experienced trainee neglected the question whether visualization stimulates reflection or the provision of structures can reduce the cognitive load, because the time schedule is just a trivial task for him and all relevant sections of a lesson plan are already known. Several master's students and a beginning trainee, on the other hand, stated that they were able to concentrate more intensively on planning, because the predefined fields helped them to not to forget an important aspect, to differentiate and to reflect on their lesson plan. Reflection on the teaching/learning contents was denied by all participants.

Asked about the quality of the created lesson plan the participants answered that they think it is better regarding completeness, but worse in detail due to the limited time for planning, compared to lesson plans they usually create. When the lesson plans were discussed in class, the teacher educators said that the results met their expectations and rated the lesson plans as comparable to traditional ones for the specific planning task.

6.3. Practicability

Several participants from all levels of experience reported that they would like to use PLATON in the future. In general, the tool was seen helpful for re-using lesson plans and especially for planning longer units consisting of more than one or two lessons. The trainees in particular reported that the tool demands a detailed planning and stated that they would do this only for teaching visits or example lessons, because of an estimated additional effort that could not be achieved for everyday planning. They also asked how long the system would be available, whether it is possible to share lesson plans, and whether the upcoming new standards are already integrated. Also, several participants (all levels) asked whether they can use PLATON regularly.

Due to the auto formatting of the print view, PLATON was also considered a major time saver. The printout was characterized as comparable in content to "traditional lesson plans", whereby it was seen positive that the lesson plan generated by PLATON additionally contained the timeline view. However, high (especially aesthetic) demands were placed on the printout such as fine-grained control of column/text alignment, page breaks, word breaks, and page orientation. The print dialogue already offers a multitude of variations, but these do not yet seem to be sufficient in particular for some trainees who criticized the generated printout as not yet mature enough in terms of flexibility and "beauty" (note that the printout was not the focus of the investigations).

On the one hand, bachelor's and master's students declare PLATON as particularly helpful in the post-university teacher training when they have to develop lesson plans more regularly. On the other hand, most trainees recommend PLATON particularly for learning lesson planning in university.

There was a controversy about the tool being usable online only. Several participants requested a tool which can be used without an internet connection, because the Internet is not always available (e.g., in schools). Another recurring criticism of the use of PLATON in almost all intervention was the need to use computers. Finally, there were also feature requests: one participant from the English course asked for an automatic updated preview of the print view. During the study, the lesson was planned with PLATON and the print view had to be requested manually; this request is implemented in the latest version. Another desired feature is a spell checker. Also requested from a trainee was a feature for creating a seating plan in PLATON.

6.4. Which Features Were Used and Which Ones Are of Particular Importance?

The most important feature turned out to be the standards integration. This was most widely used and considered the most helpful—especially the overview of the whole sequence (cf. FR6) was often explicitly mentioned. In particular, the master's students considered the integration as a big "relief". Likewise the printout (cf. FR15) was given great importance and was used by most participants multiple times (in particular the automatic formatting, cf. Section 6.3). Resources were added by nearly all participants and linked to specific teaching activities. Color-coding and parallelism of actions were used quite frequently and were characterized as useful for a better overview and planning of internal differentiations. Nesting of actions, however, was used relatively rarely. The comments of the trainees for the statistical function should be emphasized: This feature was praised during the presentation, but rarely used afterwards. The evaluation was viewed critically, as it does not offer any great benefit for an individual lesson and would not make sense if only single lessons of a unit were planned in detail.

Generally, sharing and re-using of lesson plans were seen as a very important features of any lesson planning system (despite these features were not evaluated/used in the studies) and seem to become more important with increasing teaching experience.

6.5. Quantitative Evaluation

In this section, the results of the quantitative questionnaires are presented.

6.5.1. Questionnaire

Starting with the interventions with the math teacher students the following questionnaire was introduced in order to complement the qualitative group interviews ($n = 34$). For each of the following statement the participants were asked to rate it on a Likert-scale with five options ranging from "fully agree" (5) to "fully disagree" (1).

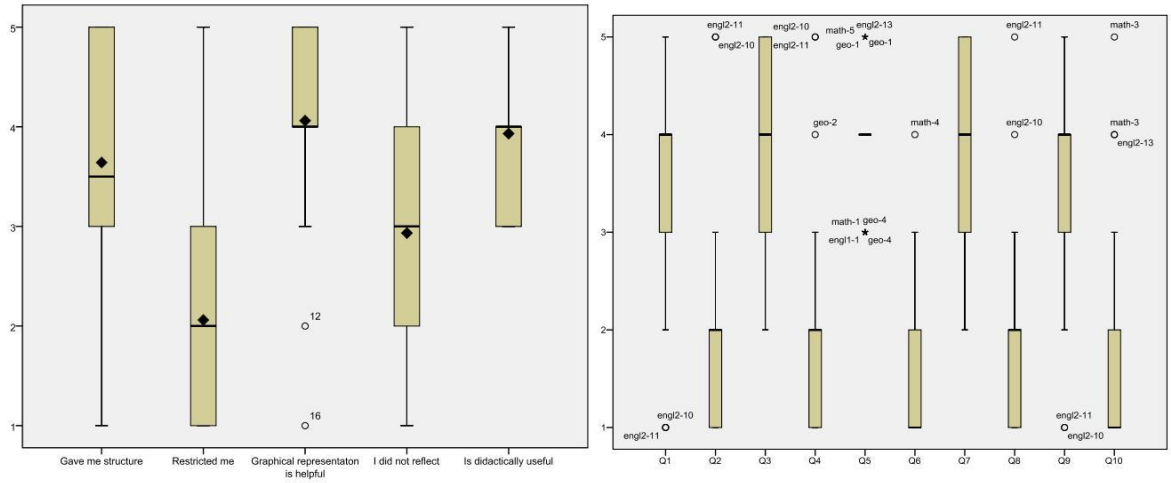
- The tool gave me structure in the planning process.
- The tool has restricted me in my planning process.
- The graphical notation for planning a lesson is helpful.
- I did not reflect on the lesson plan while planning.
- The given structure has been chosen didactically meaningful for lesson planning.

The results are depicted in Figure 4a. The first statement shows a positive tendency towards PLATON providing structure to the participants ($m = 3.5$). There are, however, two persons ("engl2-10" and "engl2-11") who fully disagree, without those two the median is 4. Also, the results indicate that PLATON did not restrict the participants while planning ($m = 2$). Ratings of "fully disagree" were only present by three participants ("engl2-10", "engl2-11" and "math-2"). Noteworthy is, however, that "math-2" rated the first and third question with "fully agree" and has a SUS score of 90 ("excellent", cf. Section 6.5.2). There is a clear tendency ($m = 4$) that the graphical representation was seen as helpful—most participants rated it with "(fully) agree". However, there are two outliners who "(fully) disagree". The statement regarding reflection was answered very heterogeneously: All options from "fully agree" to "fully disagree" were present. The median is $m = 3$ (exactly the middle) which indicates that this question was not easy to answer for the participants after one lesson planning session. Finally, the last question shows that the predefined structure of the PLATON tool seems to be reasonable chosen ($m = 4$).

6.5.2. System Usability Scale

Figure 4b shows the detailed results of the System Usability Scale (SUS) questionnaire [75]. As it was pointed out in [76] that there could be misunderstandings of nonnative English speakers, a translated version of the SUS was used [77]. The overall SUS score was 74 (average; median = 78) and, thus, can be rated as "good" [78]. The boxplot diagram shows that most ratings are relatively

consistent. Except two outliners ("engl2-10" and "engl2-11" with an SUS score of 35 resp. 20), there are no other SUS scores below 57 ("ok"). Approx. 25 % of the participants rated PLATON with a score of at least 85 ("excellent").



(**a**) Boxplot displaying the questionnaire ratings (**b**) Boxplot displaying the System Usability Scale ratings

Figure 4. Evaluation results.

An in-depth analysis of the SUS results shows that the majority of participants can imagine or would like to use PLATON regularly (Q1, $m = 4$) and rate it as easy to use (Q3, $m = 4$) as well as not too complex (Q2, $m = 2$). Help of technical persons (Q4, $m = 2$) to use the system was not required for most participants and there was a clear tendency that most people would learn to use the system very quickly (Q7, $m = 4$). Furthermore, there was a wide consent that the participants did not need to learn a lot of things before they could "get going" with this system (Q10, $m = 1$). Generally, the participants found themselves quite confident while using the system (Q9, $m = 4$) and it was not very "cumbersome" to use (Q8, $m = 2$). Apart from four participants all others found that the different functions where well integrated (Q5, $m = 4$) and even more participants denied the system to be very inconsistent (Q6, $m = 1$).

7. Discussion

In the SUS the majority of students agreed with the statement "I think I would like to use this system regularly" ($m = 4$, out of a total of 34 people 7 fully agreed and 17 almost fully agreed). In addition, many students indicated on questionnaires and in group interviews that they would like to continue using PLATON. Among the prospective teachers in the post-university training, the approval was not quite so high in principle, whereby new lateral entrants tended to be more inclined to imagine a future usage in practice compared to university graduates. It is noteworthy that PLATON was seen by (more experienced) prospective teachers in the post-university training as particularly suitable for learning lesson planning at a university, but by master's students as particularly suitable for the post-university training (presumably due to expected work facilitation). This looks like a contradiction at first. On the one hand, it is questionable to what extent Master's students are able to correctly assess the teacher training. It could also be possible that the use of PLATON in its current form "only" offers a special benefit at certain phases of the training, e.g., for the first learning of lesson planning, in the internship or at the beginning of the teacher training, in order to support prospective teachers in learning or improving their existing skills. On the other hand, it could also be the case that master's students already value PLATON, but estimate that it is even more helpful when they have to plan lessons more regularly. In addition, more experienced trainees are likely more accustomed to their usual form of planning, which could also be an inhibition to engage in a new way of planning.

Statements that should not be underestimated are, for example, Microsoft Word is the "familiar form" or "feels more beautiful" because they indicate a high entry hurdle for the use of a new tool (cf. [26,79]).

The trainees especially noted that the lesson plans developed with PLATON all looked very similar. This was especially because the lesson plans were strongly oriented on the pattern "introduction, elaboration, and evaluation" and only these terms were assigned as titles to the actions. Examples of such strict flows could be found in all interventions. The use of such a strict structure was not expected and could point out that this pattern is so deeply anchored in the minds of the participants and that they transferred the known planning procedure directly to PLATON. Mühlhausen criticizes such as strict structure: "There is no doubt that it makes sense to think about the structure of one's lessons and to get to know and try out different (!) possibilities. But the demand to always structure lessons in the same way according to whatever planning scheme is gross nonsense" (cf. [80] p. 52). Another possible explanation for this could be that some participants did not get involved into the "new" possibilities of PLATON.

The heterogeneous results with regard to flexibility and the predefined structures resp. (custom) fields are conspicuous: The provided fields were considered meaningful, adequate, and flexible enough, but at the same time it was noted by an experienced lateral entrant trainee that many aspects were already "too pre-structured". Nevertheless, a trainee lacked certain input options, and although the definition of custom fields was known, he did not define them because this was considered too time-consuming. This heterogeneity can perhaps be explained by personal preferences developed over time (see above). An indication for this reasoning consists of the statements about the insufficient flexibility and adaptability of the print view, which reflects almost the same structure. The question of the actual effort cannot be answered by the study carried out, but requires a longer-term use and more experience with the tool used.

If one considers the statement that no reduction in cognitive effort can be achieved by predefining fields, since the sections are usually already known, the origin must be taken into account: It comes from an experienced planner. For beginners who have not yet internalized these sections, the specification can certainly be a relief. However, also for an experienced trainee, the offered fields led to consider certain aspects, e.g., the anticipated difficulties. Of course, a differentiation has to be made here, because stupidly filling in text fields with trivial information does not represent any real added value, but only costs time. However, it cannot be assumed from the usage that all text fields were filled with trivial information. Rather, predefined text fields seem to have been used exactly when the planning person had something to describe (e.g., many participants did not enter the expected learning results or did not enter them in all actions).

Some participants replied that PLATON helped them to reflect on their lesson plan. Since all assessments are based exclusively on self-disclosure, however, it cannot be ruled out with certainty that subconscious reflection took place after all. To be able to make more precise statements, a comparison of the number of adjustments to time when using PLATON with traditional planning would be interesting, for example.

There were many different statements on the subject of time expenditure. In some cases PLATON was attributed being a time saver (e.g., through automatic formatting) but also requiring much more time for detailed planning and filling all fields. The following statements could provide a possible explanation: (1) The navigation could be improved and (2) a lot of "clicking back and forth" had to be done in order to enter all aspects in the "right" places. With regard to navigation it can be assumed that not all "abbreviations" are known when PLATON is used for the first time. The use of cumbersome navigation paths supports this assumption: When presenting the plans in class, the sequence level was used instead of the browser navigation to switch back to the last view. Basically, the statement "a lot of clicking around is necessary" must be accepted, since the view has to be changed in order to enter the descriptions of the individual actions. Here a comparison with a text program and the possibly necessary scrolling would be interesting.

The evaluation of the quality of the lesson plans was not the focus of this study and can therefore only be regarded as a trend. However, this circumstance should not diminish the value of the students' statements, since they do not expect any limitations from the use of PLATON, but rather speak of the partly lower quality with regard to the limited planning time and partly also of a small improvement with regard to completeness.

8. Conclusions

In this article, a web-based lesson planning system named PLATON was presented. PLATON is based on a new approach with a graphical, time-based representation and provides different views on the lesson plan, detailed resource management, analysis capabilities, and automatic feedback to promote reflection upon the lesson plans.

The empirical evaluation involving 61 participants (bachelor's, master's and post-university teacher training level as well as different subjects) with a focus on the usability of the approach and the prototype has shown that it is usable for lesson planning and has no fundamental usability flaws. Hence, the PLATON prototype can be used in practice and for further in-depth evaluations. Combined with the most used and most valued features, the compiled functional and non-functional requirements have also proved to be a good foundation. There also seems to be a quite strong preference to the already used tool for planning for more advanced prospective teachers.

Only a tendency to promote reflection could be observed with the help of a graphical, time-based representation with a predefined structure—especially among bachelor's and master's students. Here, further research is necessary. It would also be interesting to investigate what influence a tool has on the completeness and quality of developed lesson plans.

Funding: This research received no external funding.

Acknowledgments: The author acknowledges the support of the Deutsche Forschungsgemeinschaft and Open Access Publishing Fund of University of Potsdam. Additionally, the author would like to thank Anja Penßler-Beyer for proofreading.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A. Requirements

This section provides a list of clear requirements for lesson planning systems. The wording of the criterions is based on [81] whereas "must" requirements build the basis for lesson planning and "should" requirements simplify it.

Appendix A.1. Functional Requirements

The following Table A1 lists the criterions for the functional requirements of Section 3.

Table A1. Criterions for the non-functional requirements.

Functional Requirement	No.	Criterion
Personal workspace (FR1)	FR1.1	The system must provide a personal, protected workspace.
Adaptability and customizability of the system (FR2)	FR2.1	The system must be customizable in terms of input fields, which can be used for planning activities, lessons, and units.
	FR2.2	The system must allow users to format the content in a known way such as common WYSIWYG office tools (e g., enumerations, bold/italic text, . . .).
Management of units (FR3)	FR3.1	The system must allow users to create, delete and duplicate units.
	FR3.2	The system must provide users input fields for typical aspects (e.g., topic, description of the learner group and goals).
	FR3.3	The system must allow creating, deleting, ordering, and duplicating lessons of a unit.
Planning/editing of lessons (FR4)	FR4.1	The system must provide users input fields for typical aspects (e.g., topic, duration of the lesson, goals).
	FR4.2	The system should provide users access to superordinate details (such as goals) of the unit without the need to change the current view.
Planning teaching activities (FR5)	FR5.1	The system must allow users to plan the sequence of teaching activities/phases of a lesson (including differentiations, nesting, parallel group tasks and alternatives).
	FR5.2	The system must provide input fields for the description of typical elements of activities/phases (e.g., title, duration, description) and optional elements (e.g., social form).
	FR5.3	The system should automatically update depending values such as start and end times when modifying the teaching sequence.
	FR5.4	The system must allow teachers to add backup time, i. e. more activities than the in-class time available.
	FR5.5	The system must allow users to move teaching activities into other lessons of the unit.
	FR5.6	The system should allow users to color code teaching activities.
Standards/Curriculum (FR6)	FR6.1	The system should make all relevant standards available directly within the system.
	FR6.2	The system should allow users to select and link standards to units and lessons.
Resource management (FR7)	FR7.1	The system must allow the users to create, view, edit, and delete resources such as books, links and arbitrary files.
	FR7.2	The system must allow resources to be associated with (multiple) activities of a lesson.
Anticipated student solutions (FR8)	FR8.1	The system must provide options for modeling resources created by students. It must be possible to describe these resources (as model solutions or e.g., to indicate a level of expectations).
Providing different views (FR9)	FR9.1	The system should provide different specialized (over)views on the plan, such as all resources linked to a unit or a lesson, a timeline view of resource usage, standards used in the unit.
Statistical analyses (FR10)	FR10.1	The system should provide various statistical evaluations such as an analysis of percentage of social forms used in the planned lessons.
Automatic feedback (FR11)	FR11.1	The system should contain an automated analysis which provides hints on possible errors and optimizations (e.g., a missing/incomplete descriptions or an unrealistic schedule).
Notes and comments (FR12)	FR12.1	The system must allow users to enter notes for units, lessons and teaching activities.
	FR12.2	The system should allow notes to be flagged in a way, so that these are highlighted.

Table A1. *Cont.*

Functional Requirement	No.	Criterion
Sharing of plans (FR13)	FR13.1	The system must allow users to share their plans with other registered users so that these designs can be viewed in the same way as self-developed ones.
Printing (FR14)	FR14.1	The system must allow the lesson plan to be saved as a PDF file or printed on paper.
	FR14.2	The printout must be customizable so that users can influence which aspects are to be displayed (e.g., page orientation and sections to be included in the printout).
Export (FR15)	FR15.1	The system must allow exporting a unit or a single lesson containing all resources in a standardized format such as a ZIP archive that can be used without the planning tool.

Appendix A.2. Non-Functional Requirements

The following Table A2 lists the criterions for the non-functional requirements of Section 3.

Table A2. Criterions for the non-functional requirements.

Non-Functional Requirement	No.	Criterion
Usability (NFR1)	NFR1.1	The system should be based on common user-interface concepts and be as self-explanatory as possible. In particular, for use in timely constrained studies, the system must ensure a short training period (maximum 15 min, self-describing, learning support).
	NFR1.2	The system should at best avoid possible errors by selecting suitable metaphors or visual feedback, or at least rendering them understandable so that they can be easily remedied.
Interoperability (NFR2)	NFR2.1	The system must be platform independent and usable without installation.
Performance & Scalability (NFR3)	NFR3.1	The user interface must respond within a reasonable time (<1 s).
	NFR3.2	The architecture must explicitly support scalability (vertical and/or horizontal).
Customizability/ Extendability (NFR4)	NFR4.1	The architecture of the system should be modular with defined interfaces that extensions and adaptations are possible without significant effort.
Openness (NFR5)	NFR5.1	The system should be fully developed, compiled and usable with open source software.
Neutral naming (NFR6)	NFR6.1	When selecting predefined names or labels, neutral terms should be chosen in order to prevent the association with specific planning models.
Security (NFR7)	NFR7.1	The system must be designed and implemented in such a way as unauthorized access and (typical) attack vectors (e.g., SQL injection, session hijacking, etc.) are prevented.

References

1. Westerman, D.A. Expert and novice teacher decision making. *J. Teach. Educ.* **1991**, *42*, 292–305. [CrossRef]
2. Wild, M. Designing and evaluating an educational performance support system. *Br. J. Educ. Technol.* **2000**, *31*, 5–20. [CrossRef]
3. Wiske, M.S.; Sick, M.; Wirsig, S. New technologies to support teaching for understanding. *Int. J. Educ. Res.* **2001**, *35*, 483–501. [CrossRef]
4. Mutton, T.; Hagger, H.; Burn, K. Learning to plan, planning to learn: The developing expertise of beginning teachers. *Teach. Teach.* **2011**, *17*, 399–416. [CrossRef]
5. Livingston, C.; Borko, H. Expert-novice differences in teaching: A cognitive analysis and implications for teacher education. *J. Teach. Educ.* **1989**, *40*, 36–42. [CrossRef]
6. John, P.D. *Lesson Planning for Teachers*, reprint ed.; Cassell education: London, UK, 1995.

7. Hattie, J. *Lernen Sichtbar Machen. Visible Learning*; Überarb. dt.-sprachige Ausg., 2., korr. Aufl.; Schneider-Verl. Hohengehren: Baltmannsweiler, Germany, 2014.
8. Cameron, L. LAMS: Pre-Service Teachers Update the Old Lesson Plan. In *Proceedings of Society for Information Technology & Teacher Education International Conference 2008*; McFerrin, K., Weber, R., Carlsen, R., Willis, D.A., Eds.; Association for the Advancement of Computing in Education (AACE): Las Vegas, NV, USA, 2008; pp. 2517–2524.
9. Shneiderman, B. Direct Manipulation: A Step Beyond Programming Languages. *IEEE Comput.* **1983**, *16*, 57–69. [CrossRef]
10. Seel, A. Von der Unterrichtsplanung zum konkreten Lehrerhandeln—Eine Untersuchung zum Zusammenhang von Planung und Durchführung von Unterricht bei Hauptschullehrerstudentinnen. *Unterrichtswissenschaft* **1997**, *25*, 257–273.
11. Meyer, H. *Leitfaden zur Unterrichtsvorbereitung*, 7. Auflage ed.; Cornelsen: Berlin, Germany, 2014.
12. Strickroth, S. Unterstützungsmöglichkeiten für die computerbasierte Planung von Unterricht. Ein graphischer, zeitbasierter Ansatz mit automatischem Feedback. Ph.D. Thesis, Humboldt-Universität zu Berlin, Berlin, Germany, 2016. [CrossRef]
13. Tyler, R.W. *Basic Principles of Curriculum and Instruction*; University of Chicago Press: Chicago, IL, USA, 1949.
14. Yinger, R.J. A Study of Teacher Planning. *Elem. Sch. J.* **1980**, *80*, 107–127. [CrossRef]
15. Klafki, W. Didactic analysis as the core of preparation of instruction (Didaktische Analyse als Kern der Unterrichtsvorbereitung). *J. Curric. Stud.* **1995**, *27*, 13–30. [CrossRef]
16. Bybee, R.W.; Taylor, J.A.; Gardner, A.; Van Scotter, P.; Powell, J.C.; Westbrook, A.; Landes, N. *The BSCS 5E Instructional Model: Origins and Effectiveness*; BSCS: Colorado Springs, CO, USA, 2006; Volume 5, pp. 88–98.
17. Marshall, J.C.; Horton, B.; Smart, J. 4Ex2 Instructional Model: Uniting Three Learning Constructs to Improve Praxis in Science and Mathematics Classrooms. *J. Sci. Teach. Educ.* **2009**, *20*, 501–516. [CrossRef]
18. Esslinger-Hinz, I.; Wigbers, M.; Giovannini, N.; Hannig, J.; Herbert, L.; Jäkel, L.; Klingmüller, C.; Lange, B.; Neubrech, N.; Schnepf-Rimsa, E. *Der ausführliche Unterrichtsentwurf*; Beltz: Weinhein und Basel, Germany, 2013.
19. Saad, A.; Chung, P.W.H.; Dawson, C.W. Effectiveness of a case-based system in lesson planning. *J. Comput. Assist. Learn.* **2014**, *30*, 408–424. [CrossRef]
20. Chung, L.; do Prado Leite, J.C.S. On Non-Functional Requirements in Software Engineering. In *Conceptual Modeling: Foundations and Applications*; Lecture Notes in Computer Science; Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5600, pp. 363–379. [CrossRef]
21. Glinz, M. On Non-Functional Requirements. In Proceedings of the 15th IEEE International Requirements Engineering Conference, Delhi, India, 15–19 October 2007; pp. 21–26. [CrossRef]
22. Löhner, S.; van Joolingen, W.R.; Savelsbergh, E.R. The effect of external representation on constructing computer models of complex phenomena. *Instr. Sci.* **2003**, *31*, 395–418. [CrossRef]
23. Ainsworth, S. DeFT: A conceptual framework for considering learning with multiple representations. *Learn. Instr.* **2006**, *16*, 183–198. [CrossRef]
24. Meyer, H. *Was ist guter Unterricht?* Cornelsen Scriptor: Berlin, Germany, 2004.
25. Helmke, A. Was wissen wir über guten Unterricht? Über die Notwendigkeit einer Rückbesinnung auf den Unterricht als dem "Kerngeschäft" der Schule (II. Folge). *Pädagogik* **2006**, *58*, 42–45.
26. Zhou, Y.; Gong, C. Research on Application of Wiki-Based Collaborative Lesson-Preparing. In Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing 2008 (WiCOM '08), Dalian, China, 12–14 October 2008; pp. 1–5. [CrossRef]
27. Ren, Y.; Gong, C. Empirical Study on Application of Wiki Based Collaborative Lesson-Preparing. In Proceedings of the 2011 IEEE International Conference on Information Technology, Computer Engineering and Management Sciences (ICM), Nanjing, Jiangsu, China, 24–25 September 2011; Volume 4, pp. 138–141. [CrossRef]
28. Krug, S. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*, 3rd ed.; New Riders: Berkeley, CA, USA, 2014.

29. Vaughan, N.; Lawrence, K. Investigating the role of mobile devices in a pre-service teacher education program. In *Proceedings of E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2012*; Bastiaens, T., Marks, G., Eds.; Association for the Advancement of Computing in Education (AACE): Montréal, QC, Canada, 2012; pp. 822–830.
30. Park, S.H.; Baek, E.O.; An, J.S. *Usability Evaluation of an Educational Electronic Performance Support System (E-EPSS): Support for Teacher Enhancing Performance in Schools (STEPS)*; Annual Proceedings of Selected Research and Development and Practice Papers Presented at the NationalConvention of the Association for Educational Communicationsand Technology; Association for Educational Communications and Technology: Bloomington, IN, USA, 2001; pp. 560–570.
31. Masterman, E.; Craft, B. Designing and evaluating representations to model pedagogy. *Res. Learn. Technol.* **2013**, *21*. [CrossRef]
32. Furnas, G.W.; Landauer, T.K.; Gomez, L.M.; Dumais, S.T. The Vocabulary Problem in Human-system Communication. *Commun. ACM* **1987**, *30*, 964–971. [CrossRef]
33. Teachnology Incorporated. Lesson Plan Maker. Available online: http://www.teach-nology.com/web_tools/lesson_plan/ (accessed on 8 October 2019).
34. Britain, S. A Review of Learning Design: Concept, Specifications and Tools. A report for the JISC E-Learning Pedagogy Programme. Available online: https://staff.blog.ui.ac.id/harrybs/files/2008/10/learningdesigntoolsfinalreport.pdf (accessed on 8 October 2019).
35. Griffiths, D.; Blat, J. The role of teachers in editing and authoring units of learning using IMS Learning Design. *Adv. Technol. Learn.* **2005**, *2*, 243–251. [CrossRef]
36. Prieto, L.; Dimitriadis, Y.; Craft, B.; Derntl, M.; Émin, V.; Katsamani, M.; Laurillard, D.; Masterman, E.; Retalis, S.; Villasclaras, E. Learning design Rashomon II: Exploring one lesson through multiple tools. *Res. Learn. Technol.* **2013**, *21*. [CrossRef]
37. Katsamani, M.; Retalis, S.; Boloudakis, M. Designing a Moodle course with the CADMOS learning design tool. *Educ. Media Int.* **2012**, *49*, 317–331. [CrossRef]
38. Derntl, M.; Neumann, S.; Oberhuemer, P. Opportunities and Challenges of Formal Instructional Modeling for Web-Based Learning. In *Advances in Web-Based Learning—International Conference on Web-based Learning (ISCL)*; Lecture Notes in Computer Science; Leung, H., Popescu, E., Cao, Y., Lau, R.W.H., Nejdl, W., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 7048, pp. 253–262. [CrossRef]
39. Hernández-Leo, D.; Villasclaras-Fernández, E.D.; Asensio-Pérez, J.I.; Dimitriadis, Y.; Jorrín-Abellán, I.M.; Ruiz-Requies, I.; Rubia-Avi, B. COLLAGE: A collaborative Learning Design editor based on patterns. *J. Educ. Technol. Soc.* **2006**, *9*, 58–71.
40. Villasclaras-Fernández, E.; Hernández-Leo, D.; Asensio-Pérez, J.I.; Dimitriadis, Y. Web Collage: An implementation of support for assessment design in {CSCL} macro-scripts. *Comput. Educ.* **2013**, *67*, 79–97. [CrossRef]
41. Miao, Y.; Hoeksema, K.; Hoppe, H.U.; Harrer, A. CSCL Scripts: Modelling Features and Potential Use. In Proceedings of the 2005 Conference on Computer Support for Collaborative Learning: Learning 2005: The Next 10 Years! International Society of the Learning Sciences (CSCL '05), Taipei, Taiwan, 30 May–4 June 2005; pp. 423–432.
42. Prieto, L.P.; Tchounikine, P.; Asensio-Pérez, J.I.; Sobreira, P.; Dimitriadis, Y. Exploring teachers' perceptions on different {CSCL} script editing tools. *Comput. Educ.* **2014**, *78*, 383–396. [CrossRef]
43. Wild, M. Developing Performance Support Systems for Complex Tasks: Lessons from a Lesson Planning System. Ph.D. Thesis, Edith Cowan University, Faculty of Communication, Health and Science, Perth, Australia, 1998.
44. Northrup, P.T.; Pilcher, J.K. STEPS: An EPSS Tool for Instructional Planning. In Proceedings of the Selected Research and Development Presentations at the National Convention of the Association for Educational Communications and Technology (AECT), St. Louis, MI, USA, 18–22 February 1998; pp. 309–314.
45. Liu, T.C.; Juang, Y.R. IPAS—Teacher's Knowledge Management Platform for Teachers Professional Development. In Proceedings of the International Conference on Engineering Education, Manchester, UK, 18–21 August 2002; pp. 18–22.
46. Hansen, C.C. Technology as an Electronic Mentor: Scaffolding Preservice Teachers in Writing Effective Literacy Lesson Plans. *J. Early Child. Teach. Educ.* **2006**, *27*, 129–148. [CrossRef]

47. Sloop, B.; Horton, B.; Marshall, J.; Higdon, R. Mathematical Inquiry: An Instructional Model and Web-Based Lesson-Planning Tool for Creating, Refining, and Sharing Inquiry-Based Lessons. *MathMate* **2014**, *36*, 28–36.
48. Dalziel, J. Implementing Learning Design: The Learning Activity Management System (LAMS). In Proceedings of the Interact, Integrate, Impact: 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education, Adelaide, Australia, 7–10 December 2003; Crisp, G., Thiele, D., Scholten, I., Barker, S., Baron, J., Eds.; Australasian Society for Computers in Learning in Tertiary Education: Adelaide, Australia, 2003; pp. 593–596.
49. Campbell, C.; Cameron, L. Using Learning Activity Management Systems (LAMS) with pre-service secondary teachers: An authentic task. In Proceedings of the 26th ASCILITE Auckland 2009, Auckland, New Zealan, 6–9 December 2009.
50. Masterman, E.; Manton, M. Teachers' perspectives on digital tools for pedagogic planning and design. *Technol. Pedagog. Educ.* **2011**, *20*, 227–246. [CrossRef]
51. Laurillard, D.; Charlton, P.; Craft, B.; Dimakopoulos, D.; Ljubojevic, D.; Magoulas, G.; Masterman, E.; Pujadas, R.; Whitley, E.; Whittlestone, K. A constructionist learning environment for teachers to model learning designs. *J. Comput. Assist. Learn.* **2013**, *29*, 15–30. [CrossRef]
52. Laurillard, D.; Kennedy, E.; Charlton, P.; Wild, J.; Dimakopoulos, D. Using technology to develop teachers as designers of TEL: Evaluating the learning designer. *Br. J. Educ. Technol.* **2018**, *49*, 1044–1058. [CrossRef]
53. Agostinho, S. The use of a visual learning design representation to document and communicate teaching ideas. In *Proceedings of the 23rd Annual Ascilite Conference: Who's Learning? Whose Technology*? Markauskaite, L., Goodyear, P., Reimann, P., Eds.; Sydney University Press: Sydney, Australia, 2006; pp. 3–7.
54. Baylor, A.L.; Ryu, J. The Effects of Image and Animation in Enhancing Pedagogical Agent Persona. *J. Educ. Comput. Res.* **2003**, *28*, 373–394. [CrossRef]
55. Zhang, J. The Nature of External Representations in Problem Solving. *Cogn. Sci.* **1997**, *21*, 179–217. [CrossRef]
56. Fowler, M. *Analysemuster: Wiederverwendbare Objektmodelle*; Addison Wesley: Bonn, Germany, 1999.
57. Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Entwurfsmuster. Elemente wiederverwendbare objektorientierter Software*; Addison Wesley: München, Germany, 2011.
58. Coad, P. Object-oriented Patterns. *Commun. ACM* **1992**, *35*, 152–159. [CrossRef]
59. Fowler, M. *Patterns of Enterprise Application Architecture*; The Addison-Wesley Signature Series; Addison Wesley: Boston, MA, USA, 2003.
60. Fowler, M. Passive View. Available online: https://www.martinfowler.com/eaaDev/PassiveScreen.html (accessed on 8 October 2019).
61. Fraternali, P.; Rossi, G.; Sánchez-Figueroa, F. Rich Internet Applications. *IEEE Internet Comput.* **2010**, *14*, 9–12. [CrossRef]
62. GWT Project. Available online: http://www.gwtproject.org/ (accessed on 8 October 2019).
63. Idera Incorporated. Sencha GXT. Available online: https://www.sencha.com/products/gxt/ (accessed on 8 October 2019).
64. Mordani, R. Java^{TM} Servlet Specification Version 3.0. 2009. Available online: https://www.jcp.org/en/jsr/detail?id=315 (accessed on 8 October 2019).
65. Apache Foundation. Apache Tomcat. Available online: https://tomcat.apache.org/ (accessed on 8 October 2019).
66. PhantomJS—Scriptable Headless Browser. Available online: https://phantomjs.org/ (accessed on 8 October 2019).
67. MariaDB Foundation. MariaDB.org—Supporting Continuity and Open Collaboration. Available online: https://mariadb.org/ (accessed on 8 October 2019).
68. Free Software Foundation. GNU Affero General Public License, Version 3 (AGPL-3.0). Available online: https://opensource.org/licenses/AGPL-3.0 (accessed on 8 October 2019).
69. Strickroth, S. PLATON—A Time-Based, Graphical Lesson Planning Tool. Available online: https://platon.strickroth.net (accessed on 8 October 2019).
70. Strickroth, S.; Pinkwart, N. Softwaresupport für die graphische, zeitbasierte Planung von Unterrichtseinheiten. In *DeLFI 2014—Die 12. e-Learning Fachtagung Informatik*; Trahasch, S., Plötzner, R., Schneider, G., Sassiat, D., Gayer, C., Wöhrle, N., Eds.; Gesellschaft für Informatik: Bonn, Germany, 2014; pp. 314–319.
71. Strickroth, S.; Pinkwart, N. Planung von Schulunterricht: Automatisches Feedback zur Reflexionsanregung über eigene Unterrichtsentwürfe. In *Tagungsband der 15. e-Learning Fachtagung Informatik (DeLFI)*; GI Lecture Notes in Informatics; Köllen Druck+Verlag GmbH: Bonn, Germany, 2017.

72. Terhart, E. Teacher Induction in Germany: Traditions and Perspectives. In *Professional Inductions of Teachers in Europe and Elsewhere*; Zuljan, M.V., Vogrinc, J., Eds.; Faculty of Education, University of Ljubljana: Ljubljana, Slovenia, 2007; pp. 154–166.
73. Lindgaard, G.; Chattratichart, J. Usability Testing: What Have We Overlooked? In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07), San Jose, CA, USA, 28 April–4 May 2007; ACM: New York, NY, USA, 2007; pp. 1415–1424. [CrossRef]
74. Dillen, A. The evaluation of software usability. In *Encyclopedia of Human Factors and Ergonomics*; Karwowski, W., Ed.; Taylor & Francis: London, UK, 2001; Volume II, pp. 1110–1112.
75. Brooke, J. SUS—A quick and dirty usability scale. In *Usability Evaluation in Industry*; Thomas, P.W.J.B., Weerdmeester, B., McClelland, I.L., Eds.; Taylor & Francis Ltd.: London, UK, 1996; pp. 189–194.
76. Finstad, K. The system usability scale and non-native english speakers. *J. Usability Stud.* **2006**, *1*, 185–188.
77. Lohmann, K. System Usability Scale (SUS)—An Improved German Translation of the Questionnaire. Available online: https://web.archive.org/web/20160330062546/http://minds.coremedia.com/2013/09/18/sus-scale-an-improved-german-translation-questionnaire/ (accessed on 8 October 2019).
78. Bangor, A.; Kortum, P.; Miller, J. Determining what individual SUS scores mean: Adding an adjective rating scale. *J. Usability Stud.* **2009**, *4*, 114–123.
79. Gong, C.; Zhou, Y. The Piloting Researches on Collaborative Lesson-preparing based on Eduwiki Platform. In Proceedings of the IEEE International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008; Volume 5, pp. 113–116. [CrossRef]
80. Mühlhausen, U. *Über Unterrichtsqualität ins Gespräch kommen: Szenarien für eine Virtuelle Hospitation mit multimedialen Unterrichtsdokumenten und Eigenvideos*; Schneider-Verl. Hohengehren: Baltmannsweiler, Germany, 2011.
81. Rupp, C. *Requirements-Engineering und -Management*; Hanser: München, Germany, 2014.