Concept-Level Design Analytics for Blended Courses

Laia Albó[1[0000-0002-7568-9178]], Jordan Barria-Pineda[2[0000-0002-4961-4818]], Peter Brusilovsky[2[0000-0002-1902-1464]] and Davinia Hernández-Leo[1[0000-0003-0548-7455]]

[1]Universitat Pompeu Fabra, Barcelona, Spain
[2] University of Pittsburgh, Pittsburgh (PA), United States
laia.albo@upf.edu, jab464@pitt.edu, peterb@pitt.edu, davinia.hernandez-leo@upf.edu

Abstract. Although many efforts are being made to provide educators with dashboards and tools to understand student behaviors within specific technological environments (learning analytics), there is a lack of work in supporting educators in making data-informed design decisions when designing a blended course and planning learning activities. In this paper, we introduce concept-level design analytics, a knowledge-based visualization, which uncovers facets of the learning activities that are being authored. The visualization is integrated into a (blended) learning design authoring tool, edCrumble. This new approach is explored in the context of a higher education programming course, where teaching assistants design labs and home practice sessions with online smart learning content on a weekly basis. We performed a within-subjects user study to compare the use of the design tool both with and without the visualization. We studied the differences in terms of cognitive load, design outcomes and user actions within the system to compare both conditions to the objective of evaluating the impact of using design analytics during the decision-making phase of course design.

Keywords: Design Analytics, Blended Learning, Concept-level visualization, Authoring tool, Learning Design, Smart Learning Content

1 Introduction

Learning analytics (LA) has attracted a lot of attention of e-learning researchers and practitioners over the last 10 years. Learning analytics allows instructors to evaluate how students are learning within a learning context, providing them with data-based evidence to improve the overall quality of the learning experience [1]. As the field broadened, it has become customary to recognize different categories of learning analytics and to distinguish each category by its targeted group of users or tasks. This paper focuses on *design analytics*, one of the least explored areas within this broad research field.

We adopt the definition of the term "design analytics" as the "metrics of design decisions and related aspects that characterize learning designs" [2]. A learning design

(LD) is an explicit representation of a lesson plan created by a teacher [3]. Authoring tools can assist teachers in the creation of learning designs, which can lead to computational representations of the elements within a learning design that can be automatically analyzed. Some representations are generic or neutral, which enable only some options for structural analysis of a course (e.g. the number of tasks, time planned for a set of tasks, etc.). Other representations are specific to pedagogical approaches or subject matter concepts and enables a more detailed level of analysis. Analytics of these representations can support teachers' awareness and reflection about the accumulated decisions taken along the learning process to inform pending decisions toward completion of the course designs [2].

This paper explores some approaches for fine-grained design analytics focused on visualizing critical metadata associated with learning content. Our proposed visualization covers various metadata aspects, such as the type of learning content, the nature of knowledge supported, and a list of specific knowledge concepts that a specific fragment of learning content seeks to reinforce. After a brief review of related work (Section 2), we explain what we mean by concept-level design analytics (Section 3) and introduce its implementation in a design tool that supports teachers in selecting the learning content. The design and results of an experimental study as a first exploration of the value of concept-level design analytics are reported in sections 4 and 5.

2 Related work

2.1 Design analytics in learning design environments

The term design analytics, in the cross-road of LD and LA, was coined and defined in the framework proposed by [2]. The framework is built on existing learning design tooling that included features that align with the concept of design analytics. An example of design analytics is provided by Web Collage, which analyzes the accumulated design aspects specified by the teacher when completing a template that is based on a collaborative learning flow pattern [4]. With this analysis, the tool computes and visualizes alerts that point teachers to pending actions needed to complete the design, as required by the design guidelines underpinning the pattern [4].

The idea of learning design analytics can be also observed in the *Activity or Pedagogy Profile* tool, which enables the creation of a bar chart representation to help teachers describe the distribution of tutorials and directed study modules [5]. The profile represents tasks across six activity types of a detailed unit-by-unit or week-by-week analysis. The tool was created to be helpful at different times in the design process, from first ideas to evaluation and review. Moreover, the analytics bar charts can be shared with learners and other stakeholders to express how learners are expected to spend their time, in terms of balance and shape of the expected learning activity.

Another example is the *Learning Design Support Environment* (LDSE or the Learning Designer). The LDSE provides an analysis of the properties of the designs being created by the teacher with the environment as a learning design tool [6]. In particular, it generates charts that visualize the proportion of time that students are expected to spend on the diverse types of tasks that are planned in the design, from “acquisi-

tion" to more active forms of "inquiry, discussion, production and practice". This information serves as feedback to teachers about the nature of the learning experience that the learning design proposes.

The *Educational Design Studio* [7] is a physical environment for multiple designers working in teams that is equipped with wall projectors, whiteboards, a digital tabletop, and other tools. The various displays allow for several representations of the designs being created. The environment collects data from the designs and generates various charts; for example, the proportion of learning tasks distributed in the learning spaces (e.g. tasks occurring at the lecture room, at the lab, or online). This information enhances awareness of the broad view and the progress of their designs while building and editing individual tasks, as well as facilitating comparison between designs.

The concept of design analytics has been more extensively exploited in the edCrumble learning design tool. edCrumble is a pedagogical planner that provides a visual representation of the learning designs, strongly characterized by data analytics, that can facilitate the planning, visualization, understanding and reuse of complex blended learning designs [8]. Specifically, the decision-making that occurs during the design process is supported by design analytics that result from the design of the activities sequenced in a timeline. The design analytics provided include several categories: in-class/out-of-class time analytics, tasks' cognitive process, type of student work, teacher presence, and task evaluation mode. In each category, it is possible to have different visualizations: global time analytics, analytics that depend on the activities' type (in or out-of-class), and analytics that depend on the learning objectives.

In this paper, we present our attempt to further expand the design analytics component of edCrumble in order to support teachers at an extremely fine-grained design level. The new design analytics proposals will account for the metadata from the new integration of smart learning content into the resources' panel.

2.2 Open Learner Modelling and Navigation Support for Smart learning Content

Blended learning approaches usually attempt to focus each of their different learning contexts on the activities that could be performed most efficiently in this context. For example, lecture classroom time could focus on the explanation of complicated topics and discussions and a lab session could focus on solving sample problems where the help of a human teaching assistant might be necessary, while online learning might be devoted to self-study, self-assessment, and practice. As the complexity of learning tools increases, the online component of blended learning is increasingly focused on practicing with so-called *smart learning content* [9]. Each element of this smart content is a relatively complex interactive activity, which engages students in exploration and provides real-time performance feedback. For example, in the area of computer science education, some previously explored types of smart content included interactive animations, worked examples, parameterized semantics questions, Parson's puzzles, and programming problems. As each smart learning content item is relatively complex and advanced, it usually allows a student to practice a number of different course concepts or skills, which could be introduced in different lectures or course

units. This complex nature of smart learning content makes it hard for the student to accurately track progress and to select the most relevant learning content item for further practice.

To improve student knowledge-tracking ability in their work with smart learning content, several researchers suggested concept-level *open learner models (OLM)* [10]. A concept-level OLM recognizes the presence of multiple domain knowledge components (KC), such as concepts and skills, and visualizes student knowledge progress separately for each of these skills. Made popular by the field of intelligent tutoring systems as *skillometers* [11], concept-level OLM has become popular in other types of e-learning systems. A brief review of different concept-level OLM visualizations can be found in [12].

In our own work, we have explored visual interfaces, which combine topic-level open learner modeling with navigation support in order to help learners in selecting most relevant learning content [13]. Most recently, we explored student-focused concept-level knowledge visualization to help students in tracking their knowledge and selecting relevant smart content [14]. In this paper, we attempt to further expand the application area of concept-level knowledge visualization by exploring its value in a different context—helping instructors select learning content in a blended learning context.

3 Concept-Level Design Analytics for Blended Learning

The key idea of concept-level design analytics is to visualize the concept coverage of individual learning activities as well as learning sessions (such as a lecture, a lab, or a home practice) to help instructors in creating balanced learning designs. A learning activity is usually associated with metadata, which describes its type, engaged concepts or learning objectives, expected time to complete, and other aspects. This metadata is critical to create balanced learning designs. For example, learning practice prepared for a specific lecture should offer a balance of examples and problems, rather than over-focus on just one of these types of activities, and should cover all critical concepts introduced during the lecture, rather than over-focusing on some of them. Such a balance is usually hard to achieve without supporting the instructors with appropriate design analytics.

In this section, we present the design of a concept-level *design visualization* component, which extends the design analytics offered to the users of edCrumble. To demonstrate the power of the concept-based approach, we apply it to a relatively challenging design context: developing lab and practice sessions for an introductory programming course that uses several kinds of smart learning content. This context is challenging, since these kinds of smart content are of a different nature (examples vs. problems) and cover different kinds of programming knowledge (program comprehension vs. program construction). Moreover, each content item engages students in practicing a number of different programming concepts.

To support teachers in adapting this complex context, our designed visualization offered a concept-level visualization of a learning session being constructed and al-

lowed teachers to compare different aspects of the constructed session on the concept-level by using a mirrored bar chart visualization (i.e., balance of concepts between problems and examples). Firstly, the bar chart approach for showing the distribution of concepts in a programming domain was defined after a series of user studies described in [14]. Secondly, the mirrored layout was grounded by findings in information visualization research, which show that correlation tasks (i.e. easily detecting if two data distributions were similar or not) are better supported when presented through graphs with a mirrored layout [15], and that the visual system's capability for detecting differences between two regions is more efficient when they are shown as mirror images of each other, as compared to repeated translations of each other [16].

We explain the behavior of this visualization with the following scenario. The process of adding a new activity to a learning session starts with selecting a type of learning activity to add. To support the programming context, six types of smart learning content for introductory programming (Table 1) have been integrated into the resources panel of the design tool (Fig.1. A).

Table 1. Smart learning content integrated into the learning design tool, distinguishing between examples and problems and construction and comprehension types.

ID	Title	Type	Description
WebEx	Annotated Examples	Example Compr	Annotated program examples. Students can click each line of code to see the related explanation for that line [17].
AnimEx	Animated Examples	Example Compr.	Animated program execution examples, which visualize line-by-line execution of a piece of code [18].
PCEX	Program Construction Examples	Example Constr.	Interactive program construction examples. Each example provides a goal that specifies the given example's functionality. User can click on each line of code for getting explanations [19].
PCEXch	Program Construction Challenges	Problem Constr.	Small problems to help students developing program construction skills. Each challenge is a code example with 1-3 removed lines. Students need to drag-drop candidate lines to complete a program to achieve the provided goal [19].
Quizjet	Parameterized Problems	Problem Compr.	Parameterized problems for self-assessment of student knowledge of programming semantics. Students are asked to predict the final value of a program output [20].
PCRS	Programming exercises	Problem Constr.	Coding exercises with automatic assessment. The system asks user to complete a partial code skeleton and then, it checks the submitted answer using a set of tests [21].

By clicking on each resource tab, the system shows a list of the corresponding activities available for this content type. Users can select the preview button to open and try each activity and make an informed decision when selecting the activities for a new session. When an activity is judged as suitable to be used in the design, users can drag and drop the activity's icon to the open session (lecture, lab or practice) in the editor (Fig.1. B). Once an activity has been aggregated into the design, the design analytics

panel (Fig1. C) offers a short animation that allows the user to visualize the activity's contribution in terms of concept-level knowledge coverage (knowledge gained upon its completion).

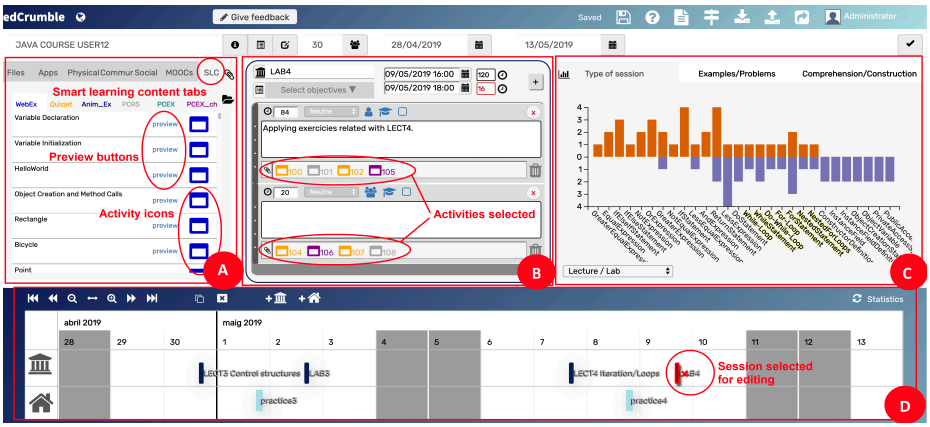

Fig. 1. Screenshot of the learning design tool's editor. (A) Resources panel with the 6 categories of smart learning content; (B) Editor for the selected session in the timeline; (C) Design analytics' visualizations; (D) Timeline with the in-class and out-of-class sessions.

Each bar on the concept-level knowledge visualization chart (Fig1. C) represents a domain concept, and its length represents how frequently the concept will be practiced by the learner when working with the selected session content (which could be also considered to be an estimation of knowledge gained after completing the session). The name of concepts that the instructor should target when designing for a specific lecture (e.g. lecture 4, with its subsequent lab-4 and practice-4 sessions) are highlighted in yellow for facilitating their coverage (see the seven concepts highlighted in Figs. 1 and 2). The concepts shown to the left of the highlighted ones are those targeted by the previous lecture, whereas those placed to the right are the ones which has not yet been introduced past lectures. The system also offers the possibility of previewing the contribution of a candidate activity to the overall design by situating the mouse over it, before dragging and dropping it into the selected session. The system then shows the preview of its contribution to learning different concepts by adding striped-bars to the visualization, as a short animation is shown when bars are added (Fig. 2 left).

In the analytics panel, we can find three tabs that offer different types of concept-level comparisons, depending on the sessions and the activities' types and knowledge. This comparisons help to balance the concept coverage of selected content by content type, session type, or covered knowledge. The first tab 'Type of session' (Fig. 2 left) allows a user to compare the concept-contribution of the activities selected, depending in which type of session they have been placed. It also offers the possibility of switching between three comparisons (Lecture/Lab, Lecture/Practice and Lab/Practice sessions). The second tab 'Examples/Problems' (Fig. 2 right) offers a unique comparison between these two types of activities but gives the option of filtering the results by

visualizing only Lab, Practice, or both. The same applies for the third tab ‘Comprehension/Construction’.

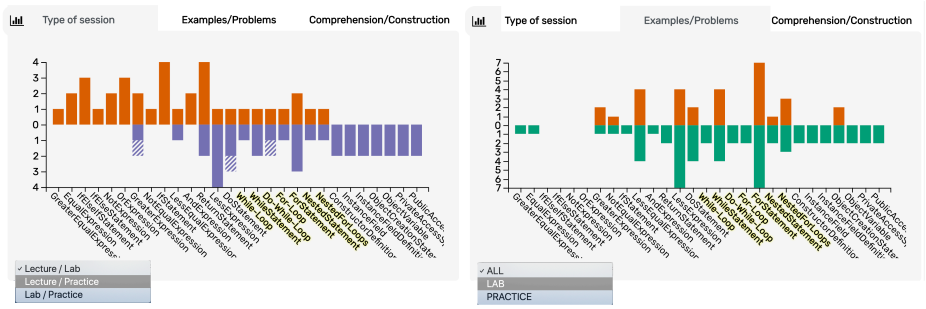

Fig. 2. Design analytics provided in concept-level visualizations. Left: Activity contribution split by the type of session (i.e., lecture on top, lab on the bottom). Right: Activity contribution split by content type (i.e., examples on top, problems on the bottom. Striped bars (left) indicate the preview of the contribution of a possible addition of a new resource.

4 Exploring the Value of Concept-Level Design Analytics

4.1 Participants and Sample

Evaluating a system focused on instructors as users is a known challenge, due to the limited availability of qualified participants. For our study, we recruited a total of 10 domain experts (six female) who were sufficiently qualified as introductory programming instructors. All of the instructors were computer or information science PhD students in a public university. Eligibility criteria required individuals to have knowledge in programming languages and experience as instructors or teaching assistants. Their ages ranged from 24 to 32 (M=28, SE=0.90) and they had between one and 13 years of teaching experience (M=3.50, SE=1.15). The scores (on a six-point scale) of how often their teaching tasks had implied selecting what activities and what type of teaching resources would be used during a course were (M=3.70, SE=0.42; M=3.60, SE=0.48), respectively. The scores (on a five-point scale) related to the instructors’ background knowledge of programming in general, in Java, and interpreting graphs were (M=4.50, SE=0.17; M=4.20, SE=0.20; M=4.20, SE=0.20) respectively. In addition to the 10 instructors, two teaching assistants were recruited as pilot users to test and refine the procedure; however, their work has not been considered in the analysis. All 12 subjects were compensated for their participation in the study.

4.2 Design and Procedure

To assess the value of the design analytics that were provided, we compared the interface without the visualizations (baseline interface) to the one with the visualizations (visualizations interface). Due to the size of our sample, we used a within-subjects design. Instructors were asked to perform two different tasks with the system, and all

of them experienced both treatments. The order of treatments was randomized to control for the effect of ordering (half of the instructors started the study using the baseline interface) and each participant did each task with just one treatment. The tasks were designed within the context of a higher education programming course (JAVA course) of 15 weeks: each week had a lecture and a lab session in class, and practice time at home. Our study was focused on the third and fourth weeks (the editor was prepared with the sessions of these two weeks to allow instructors to design within this framework) and asked instructors perform realistic design tasks to target concepts explained specifically in Lecture 4, which is described as follows. **Task 1:** Design a Lab session for Lecture 4 using eight (problems) activities in total. a) Try to ensure that the practice session covers key concepts introduced during the class (as shown by lecture examples). b) Try to strike a balance between problems that focus on program comprehension and program construction. **Task 2:** Design a Practice session for the Lecture 4 using 20 (examples and problems) activities in total. a) Try to ensure that the practice covers key concepts introduced at the class (as shown by lecture examples). b) Try to ensure a balance of examples and problems. c) Make sure that the student will have a chance to practice both program comprehension and program construction skills. The order of the tasks was not randomized, since we considered the second task to be an extension of the first (albeit with a higher difficulty). Instructors received two training sessions, one about the use of the design tool itself and the other about the use of the visualization. The group that started the study with the baseline interface received the tool training before the first task and the visualization training before the second task, while the group that started with the visualization got both trainings before the first task. During the tasks, instructors had access to help files on the six types of activities with a short description of each one (indicating the categories to which they belonged: examples/problems and construction/comprehension). After each task, we asked instructors to complete a post-task questionnaire. At the end of the study, instructors filled out a final questionnaire.

4.3 Data collection and analysis

We collected the action logs of the instructors while they interacted with the system. Above all, we focused on the actions that took place within the resources panel and the visualizations tabs. Moreover, we also gathered the learning design outcomes generated during the study to assess the instructors' performance of the tasks. After each task, we used the NASA_TLX questionnaire [22] which aimed to measure the instructors' cognitive load of the tasks' performances. We used a paper version of the questionnaire that included both known parts (rating and weights). The final questionnaire asked instructors to provide their feedback about the use of visualizations and the design tool. It had two open questions to ask instructors about their preferences between the two treatments, as well as which interface they found to be more efficient in performing the given tasks and why. The third question asked instructors to order the three type of visualizations by their level of usefulness. Next, 14+5 items were presented to instructors for gathering their feedback about the visualizations and the design tool (all of them were seven-point Likert scale: strongly disagree: 1,

strongly agree: 7). The final open question gave instructors the opportunity to provide general suggestions or comments.

5 Results and Discussion

5.1 Cognitive load

The first result of the NASA_TLX questionnaire indicates that the second task (TLX index of 56.2) presented more difficulties to the instructors than the first task (TLX index of 37.1). This is an expected result that validates the design of the study, which ordered the tasks by its level of difficulty (not randomized). Global TLX indexes indicate that, in both tasks, the perceived workload was higher when instructors do not use visualizations. The perceived mental demand (MD) is always higher when without visualization, and this difference is significant when comparing all tasks' performances together (using the visualization: M=169, SE=36.2; without visualization: M=253, SE=35; $p<0.05$). Significant results were also found for the temporal demand (TD) ($p=0.043$) and frustration (FR) ($p=0.015$) values when performing the first task. Instructors using the visualization felt that more time was needed to perform the task (time was also slightly higher in the second task when using visualizations), whereas those using the baseline interface felt more frustrated.

5.2 Action Analysis

The click data collected when instructors worked on the tasks provided an objective measure of how the two conditions (with and without the visualization) affect the way subjects use the system. Results of the action analysis (Table 2) reveal significant difference between the number of clicks performed for previewing the activities (the number of clicks being significantly higher in the case of not using the visualizations).

Table 2. User actions with the system while performing each task during the two treatments.

		With Visualization	Without Visualization	P
Task	Action	M (SE)	M (SE)	
T 1	Total actions	119.4 (18.16)	136.6 (23.0)	
	Click preview activity	4.2 (2.8)	21.4 (3.04)	
	Add activity	10.2 (0.73)	11.2 (1.69)	
	Delete activity selected	2.2 (0.73)	3.4 (1.75)	
	Time Spent (min)	13.78	11.88	
T 2	Total actions	236.4 (26.28)	211.4 (17.4)	
	Click preview activity	**1.6 (1.03)**	**23.4 (5.3)**	*p=0.03 T-test between-subjects
	Add activity	26.4 (2.79)	23.4 (1.8)	
	Delete activity selected	6.2 (2.96)	4 (1.9)	
	Time Spent (min)	19.14	17.72	

The fact of introducing the visualizations seems to change the behavior of the instructors in selecting the activities. When visual analytics were available, instructors previewed the activities much less frequently (4.2 and 6.2 times on average in tasks 1 and 2, compared with 21.4 and 23.4 in the baseline case). In other words, they decided whether or not to add the activity to the session by previewing the activity's contribution to the concept-level visualization, rather than previewing the activity itself. We can also observe that the time needed to perform the tasks was slightly higher on average in the condition with visualizations; however, this difference was not significant. Thus, the introduction of the visualization did not significantly influence the design time. Actions related to the addition and deletion of activities indicated similar results for both treatments.

5.3 Learning Design Outcomes

The learning designs collected after instructors completed the tasks provide an objective measure of how the two treatments affect the way subjects designed the two sessions (the lab and practice sessions required in the two tasks, respectively). As shown in Table 3, the presence of visualization slightly increased the instructors' ability to focus on the concepts of the target and immediate previous lectures when selecting activities (*onTopicCurrent* and *OnTopicPrevious*). However, the most impressive difference between the conditions was the almost complete disappearance of concepts that had not yet been introduced during the lectures (*outTopic*). The presence of these "future" concepts in practice and lab sessions is undesirable, since the students have not yet been introduced to them; yet instructors frequently miss these unwanted concepts when selecting learning content. As our data shows, the concept-level design analytics helped designers to avoid these future concepts in their design. When instructors used the baseline interface, they introduced, on average, a significantly higher number of future concepts (M=5.6, SE=2.61 in the first task; M=8.2, SE= 5.3 in the second task). When using the visualization, the cases of introducing future concepts practically disappeared (0 in task 1; M=1, SE=.63 in task 2).

Table 3. Learning designs' outcomes. *(p=0.011; p<0.05) T-test between subjects.

		With Visualization	Without Visualization	P
Task	Selected concepts	M (SE)	M (SE)	
T 1	OnTopicCurrent	13 (.84)	10.6 (.60)	
	OnTopicPrevious	10.2 (1.59)	8.2 (1.28)	
	OutTopic (future)	0	5.6 (2.61)	*
T 2	OnTopicCurrent	29.2 (1.39)	28.8 (1.90)	
	OnTopicPrevious	28 (5.06)	21 (2.12)	
	OutTopic (future)	1 (.63)	8.2 (5.3)	

Consider the distribution of the concepts' coverage from the learning design outcomes. Figure 3 shows how many times concepts have been practiced in the designed

sessions, on average, depending on the tasks and the treatments. Results show that using the visualization approach may have a positive impact on concept-level balance when it is necessary to select just a few activities (task 1), as the educator needs to be more precise when selecting the best ones for their class. However, when the instructor can select a higher number of activities (task 2), the probability of covering the necessary concepts by chance is higher and the presence of visualizations has a lower impact on improving the concept-level balance. However, the selection of a higher number of activities in the second task without using the visualizations led users to introduce a higher number of future concepts. When using the visualizations, in both cases, the number of future concepts selected was reduced drastically.

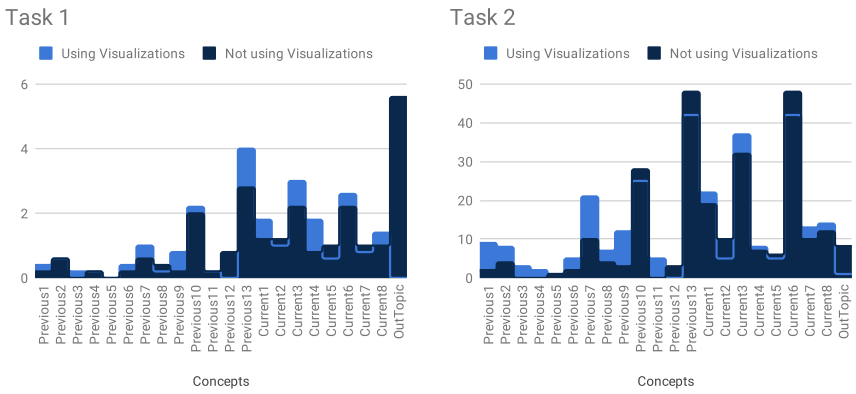

Fig. 3. Mean of the number of times that a concept is practiced during Task 1 (left) and Task 2 (right) (extracted from the learning designs outcomes) depending on the learning designs' conditions (either using or not using the visualizations). Activities can practice a concept more than once, and more than one concept at the same time. Note that there are 13 previous concepts, 8 current concepts, and a counter for future concepts.

Figure 4 presents the balance of concepts from the design outcomes, depending on the characteristics of the smart learning content. Contrary to expectations, the difference for the balance of example versus problem activities between using or not using visualizations is very low; and this balance is also very low in the case of balancing comprehension versus construction activities. We can observe only a moderate improvement of the balance and coverage of the previous concepts in both graphs when using visualizations, as well as a reduction of future concepts, as we discussed above. These results are not entirely surprising. Being domain experts, the instructors were able to understand the type and the most essential concepts of each activity by carefully reviewing its content and were sufficiently successful in balancing the number of activities added to the design (as tasks were requiring). As the log data shows, by previewing the activities, the instructors were able to achieve a reasonably balance, however, for the price of higher load. With the visualization, however, the instructors were able to reach a slightly better balance by using visual previews rather than content previews and with lower load.

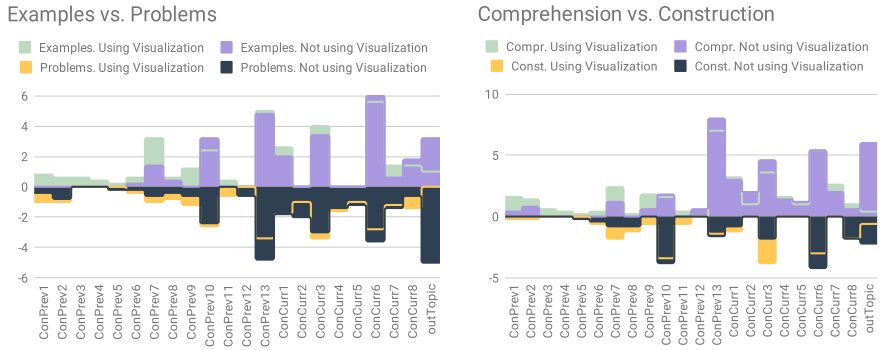

Fig. 4. Mean of the number of times that a concept is practiced during Task 2 (extracted from the learning designs' outcomes), depending on the learning designs' conditions (using or not using the visualizations). Comparison between example activities versus problem activities (left), and comprehension versus construction (right).

5.4 User Feedback Analysis

In the final questionnaire, all 10 instructors stated that they preferred to use the interface with the visualization, and that this condition allowed them to more effectively design their sessions. The visualizations were easy to understand and were useful in deciding which activity to choose; they helped instructors to check whether they were doing well enough in designing the course, as well as thinking about how knowledge was balanced. Regarding their preference about the three visualizations' tabs, six out of ten found the 'Type of session' comparison to be more useful. However, two instructors indicated the 'Examples vs. Problems' comparison as their preferred option, and two other instructors selected the 'Construction vs. Comprehension' comparison as their favorite. We can conclude that all three comparisons were meaningful for the instructors in order to create their course designs.

6 Conclusions

This paper explores some approaches for fine-grained level design analytics focused in visualizing critical metadata associated with smart learning content. Among metadata aspects covered by our visualization are the type of learning content, the nature of knowledge supported by it, and the list of specific knowledge concepts that a specific fragment of learning content allows students to practice. The visualization has been integrated into a (blended) learning design authoring tool. We expected that the concept-level design analytics would help instructors in selecting the most appropriate learning content and would result in designing more balanced learning sessions. We performed a within-subjects user study contrasting conditions both with and without the visualization. Our results indicate that the use of concept-level design analytics may reduce the cognitive load of design tasks, especially in terms of mental de-

mand. We also demonstrated that the use of design analytics has facilitated the selection of the most suitable activities without significantly affecting the overall design time. Interestingly, the presence of the visualizations has changed the behavior of instructors in the process of selecting the activities, by just previewing their contribution to the visualization without looking deeper within their content. When examining the learning outcomes, the most impressive result was an almost complete disappearance of future concepts from sessions designed with the help of visualization. Selecting content that requires future concepts is usually a design error, and the presence of the concept-level design analytics helped users to avoid these errors. Beyond that, the differences in concept balance between the conditions were small. In addition, our results hint that the visualization may have a higher impact on the concept-level balance when it is necessary to select just a few activities, as the instructor needs to be more precise selecting the best ones. On the contrary, when the instructor can select a higher number of activities, the probability of covering the concepts by chance is higher and the visualizations have a smaller impact on improving the overall balance among concept levels.

Although our results indicate that the use of design analytics improves the overall learning design quality, our study has some limitations. Most importantly, the number of subjects was too small to draw a general conclusion, which is, however, typical for studies focused on instructor-level users. Future research will be necessary to explore and evaluate the use of concept-level design analytics with a larger sample in other educational contexts and in comparing different types of visualizations. Moreover, further research may explore the connection of design analytics with learning analytics extracted from the existing smart learning content.

Acknowledgements. The authors would like to thank all the instructors who participated in the study. This work is a result of a collaboration within a mobility grant for research funded by the SEBAP, *Societat Econòmica Barcelonesa d'Amics del País*. This work has also been partially funded by NSF DRL 1740775, "la Caixa Foundation" (CoT project, 100010434) and FEDER, the National Research Agency of the Spanish Ministry of Science, Innovations and Universities MDM-2015-0502, TIN2014-53199-C3-3-R, TIN2017-85179-C3-3-R. DHL is a Serra Húnter Fellow.

References

1. Lockyer, L., Dawson, S.: Learning designs and learning analytics. In: Proceedings of the 1st International Conference on Learning Analytics and Knowledge LAK '11, 153 (2011).
2. Hernández-Leo, D., Martinez-Maldonado, R., Pardo, A., Muñoz-Cristóbal, J. A., Rodríguez-Triana, M. J.: Analytics for learning design: A layered framework and tools. British Journal of Educational Technology 51(1), 139-152 (2019).
3. Persico, D., Pozzi, F., Anastopoulou, S., Conole, G., Craft, B., Dimitriadis, Y., Hernández-Leo, D., Kali, Y., Mor, Y., Pérez.Sanagustín, M., Walmsley, H.: Learning design Rashomon I – supporting the design of one lesson through different approaches. Journal of Research in Learning Technologies 21 (2013).
4. Villasclaras-Fernández, E. D., Hernández-Leo, D., Asensio-Pérez, J. I., Dimitriadis, Y.: Web Collage: an implementation of support for assessment design in CSCL macro-scripts. Computers & Education 67, 79-97 (2013).

5. Cross, S., Galley, R., Brasher, A., Weller, M.: OULDI-JISC project evaluation report: The impact of new curriculum design tools and approaches on institutional process and design cultures. OULDI Project, at http://oro.open.ac.uk/34140/ (2012).
6. Laurillard, D., Charlton, P., Craft, B., Dimakopoulos, D., Ljubojevic, D., Magoulas, G., ... & Whittlestone, K.: A constructionist learning environment for teachers to model learning designs. Journal of Computer Assisted Learning 29(1), 15-30 (2013).
7. Martinez-Maldonado, R., Goodyear, P., Carvalho, L., Thompson, K., Hernandez-Leo, D., Dimitriadis, Y., Prieto, L., Wardak, D.: Supporting collaborative design activity in a multi-user digital design ecology. Computers in Human Behavior 71, 327-342 (2017).
8. Albó, L., Hernández-Leo, D.: edCrumble: designing for learning with data analytics. In: Lifelong Technology-Enhanced Learning. EC-TEL 2018. Lecture Notes in Computer Science, vol 11082., pp. 605–608 (2018).
9. Brusilovsky, P., S. Edwards, A. Kumar, L. Malmi, L. Benotti, D. Buck, P. Ihantola, R. Prince, Sirkiä, T., Sosnovsky, S., Urquiza, J., Vihavainen, A, Wollowski, M.: Increasing Adoption of Smart Learning Content for Computer Science Education. Working Group Reports of the 2014 on Innovation and Technology in Computer Science Education Conference, Uppsala, Sweden, ACM (2014).
10. Bull, S., Kay, J.: Student Models That Invite the Learner In: The SMILI: Open Learner Modelling Framework. International Journal of Artificial Intelligence in Education 17(2), 89-120 (2007).
11. Corbett, A., McLaughlin, M., Scarpinatto, C.: Modeling Student Knowledge: Cognitive Tutors in High School and College. User Modeling and User-Adapted Interaction 10(2-3): 81-108 (2000).
12. Bull, S., Brusilovsky, P. Guerra, J.: Which Learning Visualisations to Offer Students? In: 13th European Conference on Technology Enhanced Learning, EC-TEL 2018, Leeds, UK, Springer (2018).
13. Sosnovsky, S.,Brusilovsky, P.: Evaluation of Topic-based Adaptation & Student Modeling in QuizGuide. User Modeling and User-Adapted Interaction 25(4), 371-424 (2015).
14. Guerra, J., C. Schunn, S. Bull, J. Barria-Pineda and P. Brusilovsky: Navigation support in complex open learner models: assessing visual design alternatives. New Review of Hypermedia and Multimedia 24(3): 160-192 (2018).
15. Ondov, B.D., Jardine, N., Elmqvist, N., Franconeri, S.L. Face to face: Evaluating visual comparison. IEEE TVCG 25(2), 861–871 (2018).
16. Treder, M.S.: Behind the looking-glass: A review on human symmetry perception. Symmetry 2(3), 1510–1543 (2010).
17. Brusilovsky, P.: WebEx: Learning from Examples in a Programming Course. In: WebNet (1), pp. 124-129 (2001).
18. Hosseini, R., Sirkiä, T., Guerra, J., Brusilovsky, P., Malmi, L.: Animated Examples as Practice Content in a Java Programming Course. In: Proceedings of the 47th ACM Technical Symposium on Computing Science Education - SIGCSE '16, 540–545 (2016).
19. Hosseini, R., Akhuseyinoglu, K., Petersen, A., Schunn, C. D., Brusilovsky, P.: PCEX: Interactive Program Construction Examples for Learning Programming. In: Proceedings of the 18th Koli Calling International Conference on Computing Education Research, Koli, Finland, ACM (2018).
20. Hsiao, I-H., Sosnovsky, S., Brusilovsky, P.: Guiding Students to the Right Questions: Adaptive Navigation Support in an E-learning System for Java Programming, Journal of Computer Assisted Learning, Volume 26 Issue 4, Pages 270 - 283 (2010).
21. Zingaro, D., Cherenkova, Y., Karpova, O., Petersen, A.: Facilitating code-writing in PI classes. In: The 44th ACM Technical Symposium on Computer Science Education, SIGCSE '13, Denver, CO, USA, March 6-9, 585-590 (2013).
22. Hart, S. G.: Nasa-Task Load Index (NASA-TLX): 20 Years Later. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 50(9), 904–908 (2012).