Turning the Tables: Authoring as an Asset Rather than a Burden

Toby Dragon
Department of Computer Science, Ithaca College, 953 Danby Rd, Ithaca, NY, U.S.A.

Keywords: Authoring Tool, Concept Map, Domain Model, Intelligent Tutoring System, Factor Analysis.

Abstract: We argue that authoring of Intelligent Tutoring Systems can be beneficial for instructors that choose to author content, rather than a time-consuming burden as it is often seen. In order to make this a reality, the authoring process must be easy to understand, must provide immediate benefit to the instructor doing the authoring, and must allow for incremental development and improvement. We describe a methodology that meets all of these needs using concept maps as a basis for authoring. The methodology creates a basis for intelligent support that helps authors improve their course organization and content as they work on the authoring task. We also present details of the rapid prototype being developed to apply the methodology and the initial experiences from its use.

1 INTRODUCTION

Intelligent Tutoring Systems (ITS) have been demonstrated to be effective, but often these systems are limited to a specific domain and set of content. This is due to the nature of ITS, in that most systems necessitate some type of domain model, a formal structure that the intelligent system can navigate to connect specific assessment data to some greater understanding of the domain. Customizing these domain models to cater to different domains or to different content can be costly and unwieldy. This problem has been considered a major limitation of ITS research, creating concern that although ITS systems are effective, these systems might not scale well. Historically, this concern has been addressed by authoring tools that allow researchers and instructors to create and/or modify the domain model in an intuitive fashion, generally without programming. Even these tools can be expensive in terms of instructor time, although creators of authoring tools offer innovative measures to combat this problem (e.g., Aleven et al., 2006). Many instructors do not have the time or do not feel the need to understand the purpose and internal structure of domain models, and therefore authoring tools can seem impractical.

In this narrow view, we may be considering the wrong client and the wrong customer for our ITS authoring process. In contrast to the over-burdened instructor who is disinterested in this process, many instructors are not only willing to have a hand in the authoring of their course content, they insist upon it. The entire field of Instructional Design (ID) is grounded in this process of creating and honing course content. Instructional designers work with instructors to continually refine their class materials, organization, and content. They seek out opportunity to do this work. Earlier work in relating ID and ITS development discussed the mutual benefits and promise of integrating these activities (Capell and Dannenberg, 1993), but our literature review reveals little beyond these initial attempts. We also see the topic increasing in relevance, as the motivation to define one's own materials and organize custom courses is growing in potential and popularity due to the ever expanding set of mix-and-match materials available. The Open Educational Resources (OER) movement (Atkins, Brown, & Hammond, 2007) exemplifies the wealth of materials available from which instructors can pick and choose.

We argue that this type of instructor would want to author content using an authoring tool, if that authoring tool helps improve their course content in addition to providing a domain model for an ITS. In a foundational review of ITS authoring tools, Murray recognized that one of the main challenges of authoring tool use is the amount of scaffolding the authoring tool can provide during the model creation process (Murray, 2003). We argue that providing intelligent support during the authoring process can offer exactly that type of scaffolding, helping instructors make their course organization and

Dragon, T.
Turning the Tables: Authoring as an Asset Rather than a Burden.
DOI: 10.5220/0006761503080315
In *Proceedings of the 10th International Conference on Computer Supported Education (CSEDU 2018)*, pages 308-315
ISBN: 978-989-758-291-2

content better while simultaneously creating a domain model for an ITS. Specifically, the authoring tool can help instructors formalize their process, give a concrete structure that can be shared, assessed and iterated upon, and give feedback about the content of the produced domain model. This feedback can highlight potential issues with course content and areas ripe for improvement. Additionally, the process would produce an artifact (the domain model) utilized by an ITS to support both instructors and students while running the course.

We present a generic type of domain model that can be used across many domains and a methodology that can be used to create an authoring process including feedback for the author. This methodology provides instructors a flexible, easy-to-understand process which can help them aggregate and organize their course content. This same system can then provide a basis for meaningful ITS feedback to inform and support teachers and students. We have a rapid prototype of this methodology and have begun to employ it in the domain of computer science education. We will refer to this prototype and its use to demonstrate the power of this authoring approach.

2 RELATED WORK

This work combines ideas from a wide variety of specific fields including: ITS authoring tools, instructional design, concept mapping, factor analysis, knowledge factor analysis, and ITS systems that operate with similar types of domain models. We then pull these ideas together into a coherent vision of an authoring methodology. Due to the wide variety of disparate related work, we generally reference related work in the relevant section, rather than grouping it into a unified section.

In terms of directly related work to the larger ideas presented in this paper, we should highlight three particular works. Kumar presents an argument for using the same type of domain model we present, along with arguments about the relevance of this domain model and benefits for the students using the ITS (Kumar, 2006). Cen, Koedinger, and Junker present a similar technique to giving authors feedback through factor analysis (Cen, Koedinger, and Junker, 2006). We see these works as complimentary, since we are synthesizing ideas from both, and because their focus is on the student benefit of a better domain model (which we assume for this paper), while we focus on the potential advantage for the instructor as well. Mitrovic et al. present an authoring tool that uses a similar approach (Mitrovic et al. 2009) and their team demonstrates some positive outcomes for authors (Suraweera, 2004).

3 THE AUTHORING PROCESS

We have several goals to consider when developing the generic approach to domain models and to the authoring process itself. First and foremost, instructors interested in designing their own course content do not want to be constrained by the authoring tool. The process and model should be simple and intuitive enough to be easily understood by instructors. It should also be flexible, and allow for iterative improvement and added refinement/ complexity. For these reasons, the domain model must be easily editable and offer a wide variance in complexity (i.e., domain models should be useful and understandable when simple or complex). We want instructors to create content "their way," and in their own time. We want them to experiment with models and refine them based on experience.

3.1 Concept Map as the Domain Model

To offer this type of intuitive domain model, we rely on concept mapping (Plotnick, 1997). Concept maps are composed of nodes and edges; nodes representing concepts and edges representing relationships between those concepts. The task given to instructors is to identify key concepts to be taught and their interrelations. Each of these maps could provide great detail, or remain fairly high-level. Figures 1 and 2 show two example concept maps for a course at different levels of detail. These maps have been developed using our authoring process, demonstrating the variety of maps that could be used productively.

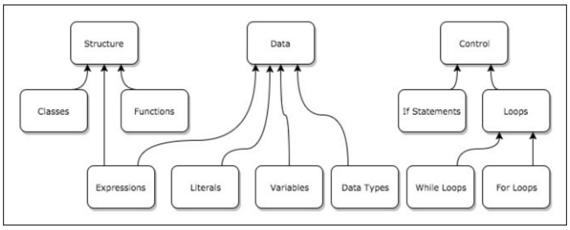

Figure 1: A high-level concept map for an introductory computer science course. This type of map can be produced quickly, and can still be useful to an ITS.

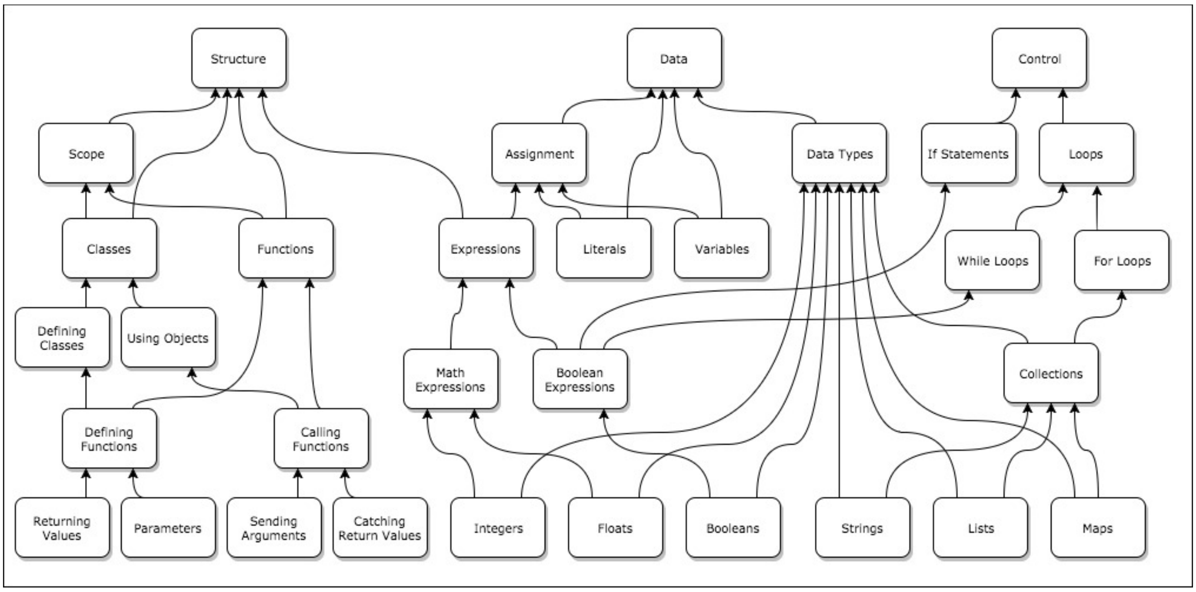

Figure 2: A detailed concept map for the same introductory computer science course. This map was produced through successive revisions beyond the one presented in Figure 1. Revising and adding detail helped the instructor reconsider course content, and the resulting map is more useful as a domain model for an ITS.

Concept mapping has been applied in many domains as an instructional or assessment tool, particularly natural science domains (Novak, 1990). This process has been shown to be useful as a means of instructional design, which is similar to the approach we recommend (Starr & Krajcik, 1990). These types of concept maps have been used as domain models for ITS and the benefits noted (Kumar, 2006).

Development of these concept maps could be considered burdensome, particularly if the benefit were perceived to be only for the ITS and not for the creator of the concept map. We need an authoring process that is immediately rewarding. We describe our proposed process, and detail how the instructor benefits. These benefits include intelligent feedback we can give authors to promote their reflection on, and improvement of, their course organization and content.

3.2 The Authoring Process

We believe that creating these concept maps can be an intuitive, productive authoring process especially if supported by an intelligent system. We now walk through the steps to apply our methodology and prototype in order to demonstrate that such a process could be implemented with current technology. There are four main steps to this process, each one important from an instructional design perspective.

3.2.1 Identifying Concepts

First, the author must collect a set of concepts to be included in the map. We suggest many sources of this type of information that should be considered, including: course syllabi, explicit student learning objectives, table of contents of related textbooks, etc. At least some of these sources should be available to any instructor, and so defining the set of included concepts should be a process to help the instructor decide: from all the possibilities, what are the most important concepts to be the focus of this course?

3.2.2 Relating Concepts

The next task is to relate these concepts to one another. In the concept mapping process, relations can be named freely as one develops their map. Since this creates complexity for the author and challenge of interpretation for the ITS using the domain model, we currently consider only one type of relationship in our concept maps. An edge pointing from one concept to another indicates that one concept "is a part of" the other concept. This relationship is purposely loose, and can be considered for aggregation (e.g., Figure 2 shows that all specific data types such as integers and floats "are a part of" understanding data types) or a prerequisite structure (e.g., Figure 2 shows that you need to understand Boolean expressions to understand if statements). This simplification is for prototyping, and the idea of introducing different types of relationships and the resulting helpfulness to instructors and to the ITS are areas for future research.

For our prototype, we ask instructors to draw these graphs on whiteboard, paper, or with an online diagramming tool (e.g., draw.io[1]). Our research team takes these drawings and converts them into a JSON format representing nodes and links. While we do not currently have an automated process for this, the existence of such online drawing tools and the fact that other ITS have offered the ability to draw such diagrams, e.g., the LASAD system (Loll & Pinkwart, 2013) demonstrates the feasibility of automating this process.

3.2.3 Identifying Resources

Once a tentative concept map has been established, the next task is to identify the resources involved. Resources represent the actual content with which students interact. We refer to three different types of resources: instruction, practice, and assessment. Any given resource can be one, two, or all three of these types. For example, a textbook or YouTube video might be purely instruction. A homework assignment graded and returned to students might be practice and assessment. An online textbook with interactive exercises might be all three. An exam not handed back to students might be solely assessment.

Generally, an already-established course has a set of resources defined. For our prototype, we represent these resources with only labels, URLs, and id strings. There are many sources from which to gather this information. For instruction and practice, instructors can consider their own presentations from class, the textbook if one is used, online resources, etc. For assessment, the spreadsheet used for grading the course has the id information of all assessment as the labels for columns. For this reason, our prototype can accept grading spreadsheets to collect assessment information. Currently the system accepts spreadsheet output from both Sakai[2] and ZyBooks[3]. Future implementations could accept any variety of output from Learning Management Systems (LMS). Another promising source for specific assessment information is concept inventories, which fit well within our system because they aim to assess specific concepts within a domain. These concept inventories are used often in natural sciences (Libarkin, 2008) but are also being applied to other domains (e.g., Almstrum et al., 2006).

3.2.4 Relating Resources

With a tentative concept map and set of resources established, the task is now to relate the resources to the concepts. This is the most crucial and productive part of the process, as the instructor needs to consider specifically how each resource is related to a set of concepts to be taught, and similarly, how each concept should be explicated through a specific set of resources. This part of the process generally causes deep reflection for the instructor, and will undoubtedly cause reconsideration of the structure of the map and the included resources. It can also be the most overwhelming part of the process, as the number of permutations of resources to concepts is not reasonable for a human to consider all at once. We now discuss methods for offering intelligent feedback during the authoring process, with particular attention to this phase of relating resources to concepts that is most challenging, but also most rewarding for authors.

4 FEEDBACK FOR AUTHOR REFLECTION

While the authoring process alone can help instructors consider their course organization and content, intelligent support for this process helps them see immediate benefit. This type of support comes in two forms which represent two aspects of artificial intelligence: theory-driven approaches and data-driven approaches. Considering theory-driven approaches, there are certain rules or constraints about the structure of the concept map indicate the potential need for reflection or improvement. Data-driven approaches allow us to use assessment information from students to validate, question, or explore the map created by instructors, again highlighting areas for reflection.

4.1 Structural Feedback

The first and most obvious form of feedback on the concept mapping task is recognition of missing resources, or missing connections to resources. Specifically, the system can check for concepts that have no assessment or no related materials and alert the instructor. Similarly, there may be assessments or materials that are used in the course but not related to the concept map. These may sound like obvious mistakes, but they are very possible to make due to the number of concepts and resources that might be included. Additionally, the reality of the issues identified and their solutions is much more subtle than might be expected. Consider some example scenarios that have occurred when working with our prototype.

An instructor found that a specific exam question was not connectable to the graph. Upon inspection, it was recognized that the question was added as a check on a semi-related topic that was discussed in class. However, the instructor decided that this topic was not important enough to warrant a node in the map. The instructor now had a few possibilities, each representing a different instructional design choice. They could remove the question, and even possibly the material from class, if it is truly not important. They could make it a part of the map, and possibly relate other resources, if it is deemed important enough. In this specific case, the instructor found upon reflection that this small topic could be presented as an example of a greater concept from the course. This caused the instructor to change the concept map and the resource links. The instructor also slightly changed the way the topic was presented to the class, and the ordering of topics presented. This demonstrates how the automated feedback on this authoring process yielded tangible results on the course organization. However, we should note that no action at all was required. The instructor could have decided to leave the exam question, change nothing else, and the assessment merely wouldn't be related to anything in the concept map.

Another productive example that occurred in use of our prototype was the recognition of large, general assessment without any narrow, specific assessment. This was recognized by the system as a set of low-level concepts that had no related assessment. Upon inspection, the instructor realized that there was a wealth of assessment of the high-level topics that rely on these low level topics. Again the instructor could have left this situation alone if they thought it sufficient, but bringing attention to the content in this way caused reflection. If there is no assessment of these individual, low-level skills, how could the students themselves know whether they had the appropriate building blocks for the high-level concept? When the assessment of high-level concept showed problems, what specifically should the student study? This prompted the instructor to add some short homework assignments that allowed students to address the low-level concepts directly before addressing the high-level concept for which they were necessary.

In these scenarios, we see subtle issues that are easily recognized by an intelligent system during the authoring process and result in productive reflection and course improvement.

4.2 Data-Driven Feedback

Moving beyond the structural feedback, a system can also utilize student assessment data to either validate the domain model or recognize areas for potential improvement. We approach this problem using factor analysis (Thompson, 2004). Factor analysis considers a set of observed variables in an attempt to identify a potentially lower number of unobserved variables called factors. In our case, we consider the grades on assessment resources to be our observed variables, and the unobserved factors to be representative of our concepts, the underlying smaller set of variables that dictate assessment performance. Other ITS researchers have applied factor analysis to learning models (Gong et al., 2001; Cen et al., 2006). Cen et al. take a similar approach to ours, applying factor analysis for reflection on the domain model, but they focus on working with domain experts to produce widely-applicable, static domain models. They highlight the model improvement for the ITS, rather than also considering the direct benefit for instructors and their courses as they instructors develop the models themselves.

We want to deliver this factor analysis to instructors, allowing them to analyze their own concept maps to stimulate their reflection and help them improve their map, thereby improving their content and the resulting domain model. Our prototype system uses the R programming language with the Structural Equation Modeling package[4] to perform these analyses. Two kinds of factor analysis offer different potential uses. Confirmatory analysis can help instructors validate their concept maps or question certain aspects of them. Exploratory analysis offers an overview of the relationship between resources, allowing a semi-automated manner of creating a map and allowing instructors to reconsider their map as a whole.

4.2.1 Confirmatory Factor Analysis

Confirmatory analysis is performed after the concept map has been authored and data is available for the assessment resources. It is important to note that the data need not be collected after the concept map has been established. As long as the instructor has data from the specific assessment used (e.g., exam questions, homework, etc.), they can apply analysis to any configuration of concept map.

Confirmatory factor analysis will take the map and the data set as input, and return a set of weights for each edge in the concept graph. The instructor can then consider those edges to decide if the data is supportive of their concept map. Low weights on edges indicate that the data does not show a strong correlation between concept and resource, even if the instructor considers there to be one.

This analysis can lead to interesting reflection and course improvement for instructors. For example, an instructor might link an assignment to three concepts from the concept map, but after running confirmatory factor analysis, discover that the weight on one edge is very small comparatively. Upon consideration, the instructor sees that, while the assignment does technically relate to the third concept, the task related to that concept is so simple that no one has answered that portion of the assessment incorrectly due to this concept. This causes the instructor to remove that connection from the map. However, upon doing so, the instructor recognizes that the concept now has no assessment. This means that the concept basically never had solid assessment, and therefore the instructor re-thinks the assignment to ensure proper assessment of the concept.

4.2.2 Exploratory Factor Analysis

Exploratory analysis offers a more general type of information to help the instructor reflect. This analysis takes only the assessment data as input, and produces the best-fit set of factors and their relations to the assessment resources. In other words, exploratory analysis creates its own concept graph, one layer deep, showing the underlying factors calculated from the resources.

This information is particularly useful to an instructor that has not yet formed a concept map. These emergent nodes are indications from the data of distinct high-level concepts that independently dictate performance on individual tasks. The instructor can review the grouped assessments to reverse engineer and label the specific factors identified. Similarly, instructors that have concept maps already created can consider these automatically-generated maps to identify the relationship between that and their own creation. This can potentially motivate a major reorganization of the concept map to better fit the reality of the assessment used.

This functionality has only recently been added to the prototype, and so reflection on its usefulness is still forthcoming. Initial attempts intrigued instructors and motivated the team to consider methods for automatically, temporarily simplifying the concept map to make the factor analysis results more useful.

4.2.3 Simplifying for Data-Driven Analysis

From our experience applying these techniques to real data sets, we have found that meaningful results on large, complex graphs such as the one presented in Figure 1 are less likely because our theoretical model is so fine-grained. Instructors are unlikely to have enough assessment items at any given low-level concept or enough data collected to find the statistical validation.

We can still check for meaningful results within the context of these maps by considering larger portions of the concept map. In this way, we simplify the model to check if it can be validated or improved on a more course-grain level. We engage in this process programmatically rather than manually. The system can iteratively remove the lowest nodes in the concept graph, and relate the resources from these lower concepts directly to the higher-level concepts. We can engage in this iterative simplification for as many or as few steps as we deem productive. For example, Figure 3 shows the concept map from Figure 2 with three iterations of simplification.

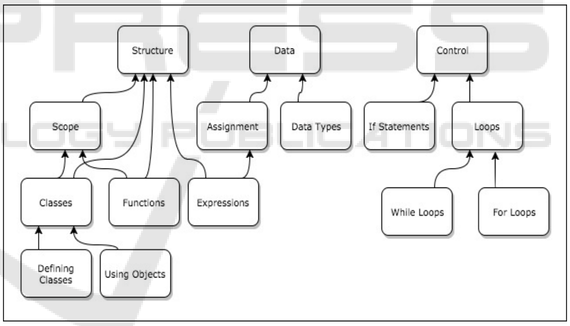

Figure 3: The concept map from Figure 2, with three successive iterations of simplification.

One can see that while not the same, this automatically generated map is quite similar to the early version of a simple map created by the author (Figure 1). With only 15 nodes instead of 32, the data-driven approaches have a much higher chance of finding statistical significance to recognize hidden factors appropriately. We are currently experimenting with methods of simplifying the graph one step at a time and calculating the confirmatory analysis at each stage, potentially automatically recognizing a level of simplification that might be most interesting for authors to consider.

5 USING THE DOMAIN MODEL

The focus of this paper is the methods of ensuring value for the instructors while they author within our system. This will ensure that domain models are authored often and with enthusiasm. We have demonstrated the manner which authoring itself can benefit an instructor interested in improving their course, both in process and through intelligent support. However, we must also note the obvious additional benefit: a domain model that can be shared with students and instructors, and also used by an ITS to provide assessment and feedback for the instructor and for students.

First, this domain model itself is a useful artifact for explicating the instructor's vision of a course. In our experimentation, we have observed instructors using the maps to communicate with other instructors when co-teaching courses. We also note that instructors have used the maps in class to motivate the learning of low-level concepts that enable the learning of high-level concepts. The visualization offers a direct means of communicating the importance of the concepts and their interrelation.

Of course, the initial reason to create a domain model is to perform intelligent analysis and feedback. While the inner-workings of an entire ITS that could use this domain model are outside the scope of this paper, there are many systems to consider that have proven effective using similar types of domain models (e.g., Dragon et al., 2006; Kumar, 2006; Sosnovsky & Brusilovsky, 2015). We now highlight the potential types of assessment and feedback that are clearly enabled through such a domain model, and are available through our system (anonymized citation).

We can calculate estimated knowledge of each concept by using the concept map structure and its relation to assessment scores. This is accomplished by aggregating scores from the related assessments to the concepts, and recursively moving up through the map in a post-order traversal. The result is knowledge estimates at a conceptual level, rather than an assignment level. This has been demonstrated to be successful in other tutoring systems (Kumar, 2006). We present these results directly to students and instructors using a color-coded visualization of the map to present this information as an Open Learner Model (OLM), which has been demonstrated to be effective (Bull & Kay, 2010).

Using these estimates, the system can also use the encoded information about the instruction and practice resources in order to offer suggestions about where students can learn about certain concepts, and where they can find appropriate practice problems.

The last type of intelligent support we offer based on this domain model is group suggestion for temporary in-class collaboration efforts. Using the knowledge estimates for different students, teams can be identified that need to focus on specific concepts, or that might be helpful to one another (anonymized citation).

6 CONCLUSIONS

We argue that the authoring process can be productive and rewarding, rather than burdensome. However, we need to identify the right people to do the job (those that want to engage in ID) and give them tools that empower them to improve their courses while simultaneously producing a useful artifact for ITS. Our proposed solution is a concept mapping process to produce a domain model. This process relies upon feedback from the structure and factor analysis in order to produce tangible benefits for instructors as well as better domain models. If instructors recognize improvement in their class organization and content when engaging in authoring, they are more likely to spend the time necessary to complete the task well.

The described process is also supportive of authors in that it can be completed gradually and iteratively, and it is useful in many contexts even when not complete (in fact, one could argue that the given authoring task is never complete, any more than a course design is complete). Authors working with our prototypes have created concept maps, but not connected any resources (electing to connect the resources during the semester as the resources are used). These instructors could still use the maps as means of communication with other instructors and students. Other instructors chose to implement only portions of their course in the map, selecting the most complex portions that lead to the most confusion and are most in need of reflection in terms of course design. The system is then applicable and fully functional for this portion of the course, and the concept map can grow to include other portions of the course in time.

Finally, we envision this process being used on a grand scale within a department as well. The same techniques discussed at a course level could be applied at the curriculum level, and the related resources and resulting knowledge estimates could give in-depth assessment information to program directors about the successes and challenges for a given program. While we have not yet implemented this within our current work, others have used concept maps as a means of curriculum evaluation (Edmondson, 1995) and we see our work as an extension and potential drastic improvement of those efforts.

Overall, If we can make the authoring process a reasonable investment with a clear benefit, instructional designers will then see the value of authoring beyond the ITS it customizes. This can in turn create an environment where wealth of reliable, validated domain models across different domains are available for use in ITS. With the low barrier to entry, obvious direct benefits, and unlimited ability to refine, we argue that our methodology has the potential to change the future landscape of ITS, creating an abundance of ever-improving domain models.

REFERENCES

Aleven, V., Sewall, J., McLaren, B. M., & Koedinger, K. R. (2006, July). Rapid authoring of intelligent tutors for real-world and experimental use. In *Advanced Learning Technologies, 2006. Sixth International Conference on* (pp. 847-851). IEEE.

Almstrum, V. L., Henderson, P. B., Harvey, V., Heeren, C., Marion, W., Riedesel, C., & Tew, A. E. 2006, June. Concept inventories in computer science for the topic discrete mathematics. In ITiCSE-WGR '06. ACM, New York, NY, USA, 132-145. DOI: 10.1145/1189215.1189182.

Atkins, D. E., Brown, J. S., & Hammond, A. L. (2007). *A review of the open educational resources (OER) movement: Achievements, challenges, and new opportunities* (pp. 1-84). Creative common.

Bull, Susan, and Judy Kay. 2010. "Open learner models." *Advances in intelligent tutoring systems*: 301-322.

Cen, H., Koedinger, K., & Junker, B. (2006, June). Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International Conference on Intelligent Tutoring Systems* (pp. 164-175). Springer, Berlin, Heidelberg.

Capell, P., & Dannenberg, R. B. (1993). Instructional design and intelligent tutoring: Theory and the precision of design. *IJAIED*, *4*(1), 95-121.

Dragon, T., Woolf, B., Marshall, D., & Murray, T. (2006). Coaching within a domain independent inquiry environment. In *Intelligent Tutoring Systems* (pp. 144-153). Springer Berlin/Heidelberg.

Edmondson, K. M. 1995. Concept mapping for the development of medical curricula. Journal of Research in Science Teaching, 32(7), 777-793.

Gong, Y., Beck, J. E., & Heffernan, N. T. (2011). How to construct more accurate student models: Comparing and optimizing knowledge tracing and performance factor analysis. *International Journal of Artificial Intelligence in Education*, *21*(1-2), 27-46.

Kumar, A. N. (2006). Using Enhanced Concept Map for Student Modeling in Programming Tutors. In *FLAIRS Conference* (pp. 527-532).

Libarkin, J. (2008, October). Concept inventories in higher education science. In *National Research Council Promising Practices in Undergraduate STEM Education Workshop 2*.

Loll, F., & Pinkwart, N. (2013). LASAD: Flexible representations for computer-based collaborative argumentation. *International Journal of Human-Computer Studies*, *71*(1), 91-109.

Mitrovic, A. et al. (2009). ASPIRE: An Authoring System and Deployment Environment for Constraint-Based Tutors. *IJAIED*, 19(2), 155-188.

Murray, T. (2003). An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art. In *Authoring tools for advanced technology learning environments* (pp. 491-544). Springer Netherlands.

Novak, J. D. 1990. Concept mapping: A useful tool for science education. *Journal of research in science teaching*, 27(10), 937-949.

Plotnick, E. (1997). Concept Mapping: A Graphical System for Understanding the Relationship between Concepts. ERIC Digest.

Sosnovsky, S., & Brusilovsky, P. 2015. Evaluation of topic-based adaptation and student modeling in QuizGuide. *User Modeling and User-Adapted Interaction*, *25*(4), 371-424.

Starr, M. L., & Krajcik, J. S. 1990. Concept maps as a heuristic for science curriculum development: Toward improvement in process and product. Journal of research in science teaching, 27(10), 987-1000.

Suraweera, P., Mitrovic, A., & Martin, B. (2004a). The Role of Domain Ontology in Knowledge Acquisition for ITSs. In J. Lester, R. M. Vicari. F. Paraguacu (Eds.) *Intelligent Tutoring Systems* 2004 (pp. 207-216). Berlin: Springer.

Thompson, B. (2004). *Exploratory and confirmatory factor analysis: Understanding concepts and applications*. American Psychological Association.

[1] https://www.draw.io/
[2] https://www.sakaiproject.org/
[3] http://www.zybooks.com/
[4] https://cran.r-project.org/web/packages/sem/index.html.