Informing Change: Course Content Analysis and Organization

Linda DuHadway and Thomas C. Henderson
University of Utah

UUCS-15-002

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

27 April 2015

Abstract

This paper introduces a novel, supportive tool for engineering educators while making course adaptations. As pointed out in the 2013 FIE workshop “An Online Revolution in Learning and Teaching,” online learning is likely to impact every department and teacher in some manner. Other innovations impacting engineering educators include active learning, peer instruction, problem-based learning, and just-in-time teaching. When implementing change, educators are expected to present existing course materials in alternative formats. One resultant difficulty is visualizing, understanding, and judging the impact of various alternatives. Learning materials’ organization is often limited by delivery methods such as learning management systems that present material linearly. This project uses text analysis and graph transformation techniques to produce various alternatives allowing educators to envision ways changes can be effectively implemented in their courses. We demonstrate how temporal and topical relations between individual learning items can be extracted from existing courses and used to produce a graph that is an effective representation of the course. From this, graph transformations produce alternative organizations of course material allowing various solutions for educators to consider while redesigning their courses. This form of automated brainstorming stimulates out-of-the-box thinking, often producing options previously not considered.

1 INTRODUCTION

In an effort to meet the changing landscape of education many departments and universities are offering more online courses – a move that is likely to impact every department in some way [9]. This will require more instructors create online courses. Other innovations in instructional strategies are also widely impacting engineering educators [2] including peer instruction, flipped classrooms, problem-based learning, just-in-time teaching, and a variety of active learning strategies. Implementing any of these strategies requires changes to existing courses. Sometimes an educator is so familiar with the current course organization that it becomes a stumbling block for visualizing alternative options.

When anticipating change it is valuable to see how existing learning materials can be organized and used in new ways. The purpose of the ENABLE project is to provide assistance in making informed changes. ENABLE is not an acronym, rather a name that reflects the purpose to enable the implementation of quality educational strategies. The two major contributions of the current ENABLE system are that it:

- gathers information about the existing course and creates a graphical representation of the relations between the learning items, and
- uses text analysis and graph transformation techniques to present alternative arrangements of the learning items.

As an example, consider the data from a sample CS0 course, Foundations of Computer Science, taught at Utah State University. The information about the learning items for this course was gathered from Canvas (a standard learning management system) and a graph was produced representing the current organization of the course; see Figure 1 (upper). This shows all the learning items for the course laid out in order across the days of the semester. Figure 1 (middle) shows the initial course graph constructed directly from the learning materials, and visually exhibiting the relations of interest: *precedes*, *occurs in*, and *includes*. The orange nodes (small, no fill color) represent the learning items. The orange edges between the learning item nodes are the *precedes* relations. The green nodes (larger with solid fill color) represent the topics. The green edges go between the topic nodes and the learning item nodes and represent the *occurs in* relations. The unit relations are expressed visually by locating nodes *included* in a unit near the same vertical location. Figure 1 (lower) shows the class material after text analysis and graph transformation by the ENABLE system. Note that this is one possible reorganization of the course. In this transformation the learning items are organized by topic. There are significantly fewer *precedes* relations which is one of the benefits of the transformation. The *occurs in* relations are all represented. In this particular transformation *includes* relations are not expressed.

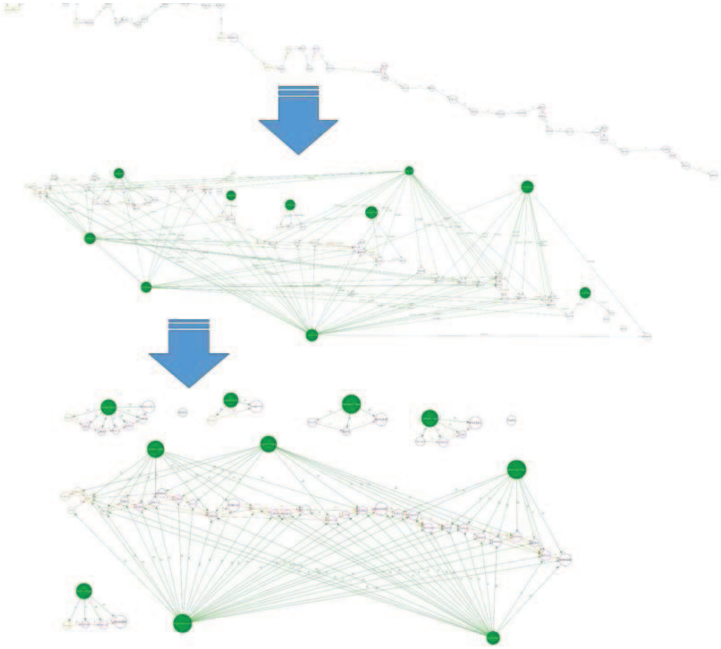

Figure 1: CS0, Foundations of Computer Science Original Course Organization (upper). CSO, Initial Course Graph Relations (middle). CS0, Transformed Course Organization (lower).

2 Discovery of Relations in Course Material

Many types of relations exist between the learning items, but here we focus on three basic ones: temporal, topical, and unit coherence. These describe the chronological order, similarity of topics, and presentation organization of the learning items, respectively.

2.1 Identifying Temporal Relations

Temporal relations express the relation in time between learning items. The word *precedes* is used to express this relation. A learning item (Item A) *precedes* another learning item (Item B) when the due date of Item A is before the due date of Item B. These relations are transitive such that if Item A precedes Item B and Item B precedes Item C, then Item A precedes Item C. When all these relations are included on a directed graph the learning item at location k in the sequence has $k - 1$ in-edges and $n - k$ out-edges, where n is the number of learning items. These graphs are too cluttered to be informative. See Figure 2 (upper).

A simpler view of the temporal relations displays only the *precedes* relations that come immediately before a given node. This graph displays an edge from a node to the node it immediately precedes in time. This reduces the number of edges to n-1 and makes a much more readable graph. Since these relations are transitive, no connection is lost. See Figure 2 (lower). These figures are the graphs produced for the CS0 course.

The meaning of the *precedes* relation is limited. Learning items connected in this way are not necessarily related by topic or grouped in the same unit. Note that *precedes* does not mean it is a prerequisite. This relation expresses nothing more than how learning items are laid out in time in the original course.

By itself, the temporal relation seems trivial, and yet it is the predominant relation presented to students. An educator who has designed and implemented a course is aware of other relations between the learning items such as how they are grouped together to create a unit of learning, how they are related by a single topic or a group of topics, and prerequisite recommendations. Although the educator may consider these other relations more significant, the learning management systems currently available use the temporal relation as the dominant organizational aspect when presenting learning materials. Even when the module tool is used in Canvas to group learning items together into units, the student view presents learning items in a linear format based solely on temporal relations in the assignments page, the gradebook, and the syllabus.

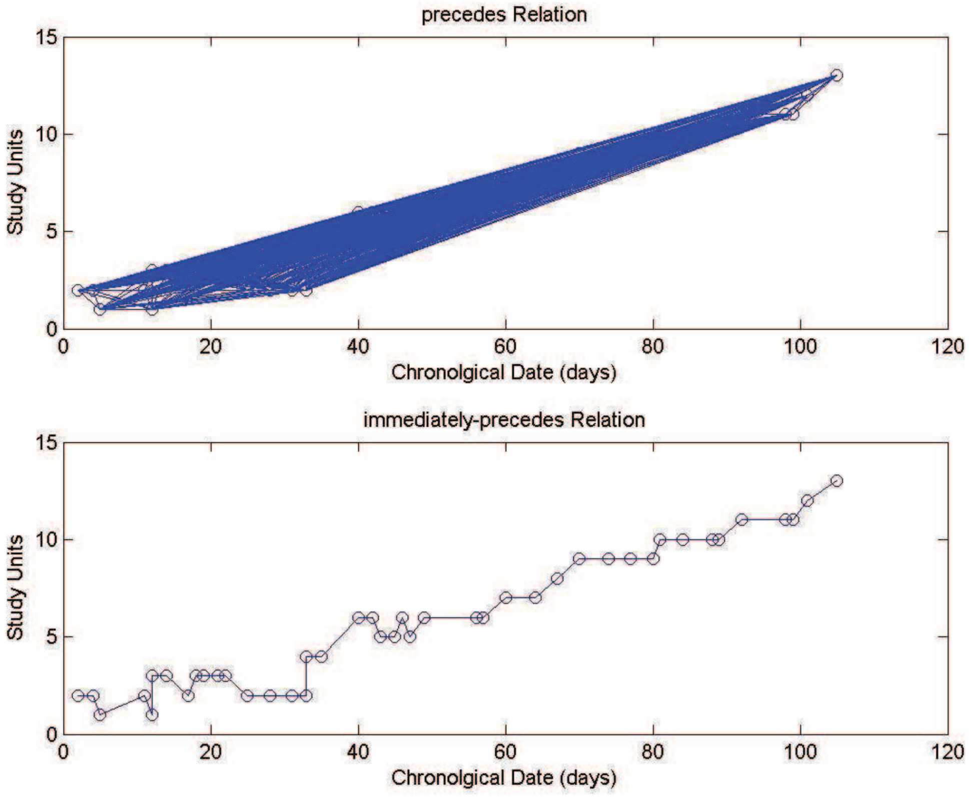

Figure 2: Detailed Graph of *precedes* Relations (upper). Graph of immediately *precedes* Relation (lower).

When combined with topical and unit relations the temporal relations add some information. For example, if there are two assignments that cover the same topic and one precedes the other it is likely that there is a non-commutative relation between the two learning items and it is important that the first is completed before the other.

Not all items in Canvas have due dates. In the sample CS0 course, 8 out of 49 learning items do not have due dates associated with them. These undated learning items include lecture notes, videos, and frequently asked questions. It is likely these items are informational materials that are most beneficial when preceding other items in the same unit or other items of the same topic. As topics extend across a larger time period than units, associating the undated learning items with other items in the unit is preferred. These learning items are dated two days before the first dated item in their unit.

2.2 Topical Relations

To identify the topical relations the text of each learning item is gathered. Canvas provides a title and a text description of each learning item. These become the basis of the text. This text is analyzed to see if there is a link to a file. Canvas has a specific way of referencing files that have been uploaded making it possible to use text parsing and regular expressions to identify these references. Once a filename is found, the file extension is considered. Currently ENABLE adds .txt and .pdf files to the text description. Pdf files are converted to text before being added. Canvas has a category of items identified as quizzes. These contain questions in addition to the text description. For these types of learning items, the questions are added to the text description.

ENABLE uses a .txt file to store a series of topic lists. These lists contain topic words, word groups, and variations. Each line in the file represents a single topic. Individual topic variations are separated by a comma. The original list of topics for the sample CS0 course includes:

- <content>
- <html, structure>
- <attribute, attributes>
- <tag, tags>,
- <element, elements>

- <publishing, host, publish, published>
- <careers, career, careers in cs, cs careers>
- <darpa>
- <history, cs history>,
- <css, style>
- <hardware, system>
- <javascript, script>
- <functions, function>
- <textboxes, textbox>
- <using the web, use the web>.

The use of a list of topic variations allows different versions of the same word such as **publishing**, **publish**, and **published** to be counted as a single topic. Stemming algorithms [11] may be used to accomplish this same grouping. However, this list of topic variations allows entirely different words to be associated with the same topic. For example, the word **host** is included in the list with **publishing**. This allows the instructor a great deal of flexibility in associating a variety of words or word phrases with a single topic.

Using these lists of topic words, a term frequency vector is created for each learning item document. Term frequency (*tf*) is a count of how many times a term occurs in the learning item document [7]. The document in this case is the description of the learning item. This description includes any text available in Canvas or uploaded by the instructor. The frequency count of terms found in a topic list are combined to produce a single *tf* count for each topic.

When computing *tf* for all the terms in a corpus of documents, this process produces high-dimensional, sparse vectors [8]. Techniques such as the application of singular value decomposition (SVD) to a topic similarity matrix (i.e., spectral graph analysis) may allow the reduction of dimension to make computationally intensive text analysis more efficient [3]. In the CS0 example here, the limited number of specific terms found in the topic lists produced *tf* vectors for which no dimensional reduction was possible.

This raw count of how many times a term occurs in a document can be more informative if it is weighted. The weighting approach used by ENABLE is *tf-idf*. *tf-idf* starts with the *tf* and then multiplies it by the inverse of the document frequency. The document frequency (*df*) of a term is the count of how many of the documents in the corpus contain that specific term. If the *df* is high, the term is very common so the fact that it shows up in a document is not as significant as a term that is less common. When the *df* is low, the occurrence of the term in a document is more significant. By multiplying *tf* by the inverse document frequency (*idf*) the resulting value results from a weighting based on the relative frequency of the term in the corpus. The ENABLE system computes *tf-idf* using log weighting of the *tf* count and log inverse frequency weighting on the document counts [7].

$$tfidf_{t,d} = (1 + log(tf_{t,d})))log(\frac{N}{df_t})$$

A Pearson correlation was done between the *tf-idf* values of the topics. For each correlation that was greater than 0.8, the topics were considered for combining. In the sample CS0 course there was a correlation between the *HTML*, *attribute*, *element*, and *tag* topic lists. Combining these was obvious once the correlation pointed them out. These are all parts of the HTML language. The other topics that were highly correlated were *JavaScript*, *functions*, and *textboxes*. Although functions is a topic that exists outside of JavaScript, in the context of this course, functions are only discussed or used in JavaScript. This correlation made the instructor aware that their broader view of the computer science curriculum was reflected in this separation of topics and would best be adapted to fit the content of this specific course. This provided the instructor a fresh perspective informed by feedback from ENABLE.

This illustrates one of the many benefits of gaining another perspective when considering changes to current courses. This process of identifying correlations between topics provided new insights into possible changes to the topic lists. These insights were not recognized when the original topic lists were made. This process led to the reduction of topics from the original fifteen to the following ten:

- <content>
- <html, structure, attribute, attributes, element, elements, tag, tags>
- <publishing, host, publish, published>
- <careers,career,careers in cs,cs careers>
- <darpa>

- <history, cs history>
- <css, style>
- <hardware, system>
- <javascript, script, functions, function, textboxes, textbox>
- <using the web, use the web>.

2.3 Unit Relations

Units are a set of learning items that are grouped together. Unit relations come directly from the modules tool in Canvas. This tool allows an instructor to group learning items into units. Many different groupings are used. Some instructors group the material based on a textbook such as a unit for each chapter. Others use it to organize temporally such as one unit for each week in the course. Another approach is to organize by specific topic coverage. Current grouping in these modules reflects groupings that are in some way meaningful to the instructor. The unit grouping of learning items is used as the y-value in Figure 1 (upper and middle) and Figure 2. This visually shows how learning items are related by unit.

3 Creating the Initial Graph

3.1 Temporal Relations

When graphing the temporal relations the *precedes* relation is used. Nodes represent learning items and edges are the relations between them. This is a directed graph with the arrow of the edge on the node with the later due date, expressing that one node precedes the other node in time. To make the graphs more readable, these edges are labeled with P.

3.2 Topical Relations

To graph the topical relations, a bipartite graph is used with one set of nodes representing the topics and the other set of nodes representing the learning items. This is a directed graph with the arrow of the edge on the learning item nodes expressing that the topic *occurs in* the learning item. For readability, these edges are labeled *OI*.

3.3 Unit Relations

The unit relations represent a grouping of the current organization. This is represented as a bipartite graph with one set of nodes representing the units and the other set of nodes representing the learning items. This is a directed graph with the arrow of the edge on the learning item nodes expressing that the unit *includes* the learning item.

3.4 Combining the Graphs

The temporal and topical relations go well together since the temporal relations are entirely in the set of learning items. This combined graph includes all the topic nodes and learning item nodes with both the *precedes* and the *occurs in* edges included.

The unit relations are loosely expressed by using the unit value to compute the vertical location of the nodes in the starting graph. This provides a visual representation of how the learning items are grouped into units but does not include any edges that connect items in a unit. Figure 1 (middle) shows the graph structure produced for the learning materials in sample CS0 course when these relations are combined.

4 Transforming the Graph

Because ENABLE identifies alternative course structures that maintain the relations between learning items it becomes necessary to transform the graph while still keeping the meaningful relations intact. Graph grammars and graph transformation systems provide a means for doing this. There is much research and many successful applications based on the research in this area [4]. One of the application areas of graph transformation systems is model transformations. This area of model transformation has become important to the field of software engineering [1]. The models used in software engineering have enough similarities to the graphical representation of learning materials to allow model transformation as the graph transformation technique used by ENABLE. These similarities include typed nodes, node attributes, and edges that represent different types of relations.

For graph transformation, ENABLE uses AGG, a development environment for attributed graph transformation [6]. It is based on an algebraic approach to graph transformations. The implementation of this approach closely follows the formal, theoretical foundation of algebraic graph transformation and so provides validation support [5] and sound behavior concerning graph transformation [10]. AGG has non-deterministic rule and match selection but provides control of this with rule layers.

4.1 Defining Semantic Rules

Once the initial course graph is available, it becomes possible to begin a conversion process from a linear (chronological) style class organization to a more non-deterministic, multi-path organization of the learning items more suitable to online delivery. It is now necessary to determine the types of desirable transforms and their meanings. We begin with the consideration of how to eliminate unnecessary *precedes* relations.

We define a restraint as an unnecessary constraint between two items. Thus, restraints are removed in order to open up more possibilities for the relations between learning items. When removing restraints it is important to maintain the integrity of the course representation.

4.1.1 T1: Topic-based *precedes* Elimination Rule

The major restraint is the *precedes* relation. It restricts any change in the order of learning items. However, many of the *precedes* relations are not necessary and can be removed without changing the necessary relations. The first step is to remove unnecessary *precedes* relations. As discussed earlier, *precedes* relations by themselves have little meaning. The fact that one learning item comes before another provides only limited information. Now that the temporal and topical relations have been combined into a single graph the system can identify *precedes* relations that have no topical connections and can be removed.

If A *precedes* B and B *precedes* C and there are no common topics that occur in both A and B, the *precedes* relation from A to B can be removed. When removing this relation it is important to keep the relation that A *precedes* C and B *precedes* C. Note, however, that the net number of *precedes* relations is reduced by 1 as there was an implied $P(A, C)$ before the application of *T1*.

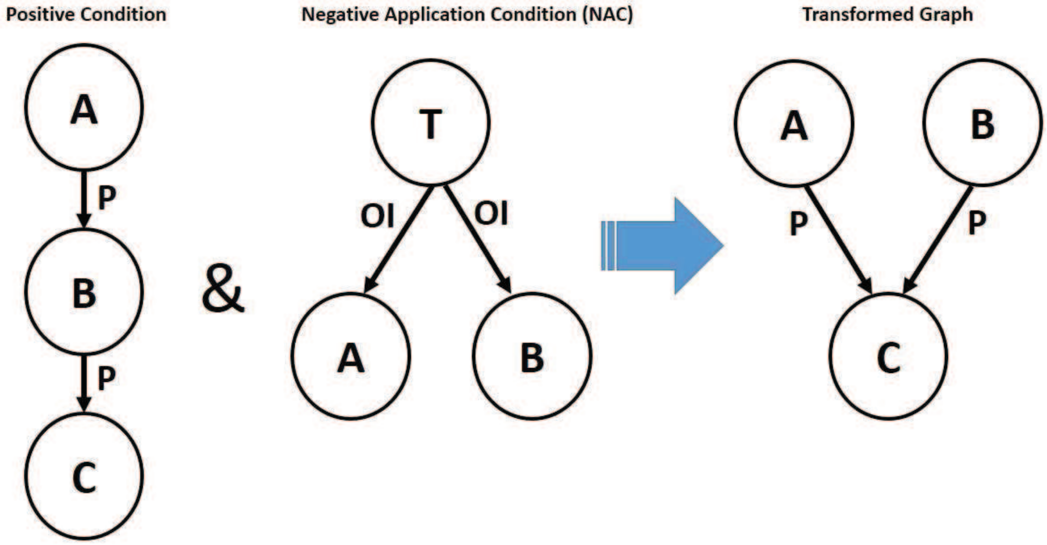

Figure 3: Graph Transformation to Eliminate Unnecessary *precedes*.

More formally, this can be stated as:

if $P(A,B) \wedge P(B,C) \wedge \nexists T \ni (OI(T,A) \wedge OI(T,B))$

then remove $P(A,B)$ and add $P(A,C)$

We call this the *Topic-based precedes Breaking* rule (*T1*). Figure 3 shows a graphical representation of this transform.

4.1.2 T2: Topic-based Exam Splitting Rule

One result of building course organization based on temporal relations is illustrated by exams. Commonly, an exam is written to assess the material that has been covered over a specific period of time such as since the last exam or since the beginning of the semester. This time-based connection is not required for assessment. Therefore it is possible to divide the material assessed in an exam by topic. Separating the temporal grouping inherent in exams provides additional possibilities for change. The split exams rule is applied after the remove *precedes* rule has been applied. Enforcing this rule application order prevents any exams being split when preceding learning items are topically related. AGG allows the user

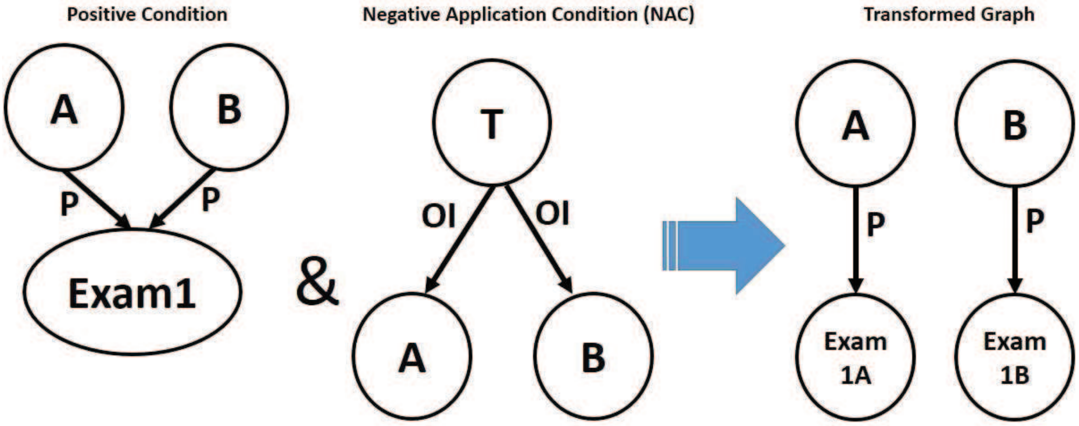

Figure 4: Graph Transformation to Split Unrelated Exam Elements.

to specify which rule layer a specific rule is in. It enforces rule ordering by applying all the rules in one layer before applying rules in the next layer. This, then, is another example of a restraint: when exams tie learning items together that are not related in any other way.

If A *precedes* Exam 1 and B *precedes* Exam 1 and there are no common topics that occur in both A and B, then Exam 1 can be split into two exams, Exam 1A and Exam 1B such that A *precedes* Exam 1A and B *precedes* Exam 1B, and A and B are independent of Exam 1B and Exam 1A, respectively. See Figure 4. This transform must be applied after *T1*.

Currently ENABLE applies both these semantic rules using graph transformations. There are additional meaningful transformations to be explored in the future work of this project, but we give them here to show the power of the approach.

4.1.3 T3: Material Splitting Transform

It may be determined from analysis of student success on homework problems and exams that there is too much material in some learning items. This leads to the transform shown in Figure 5.

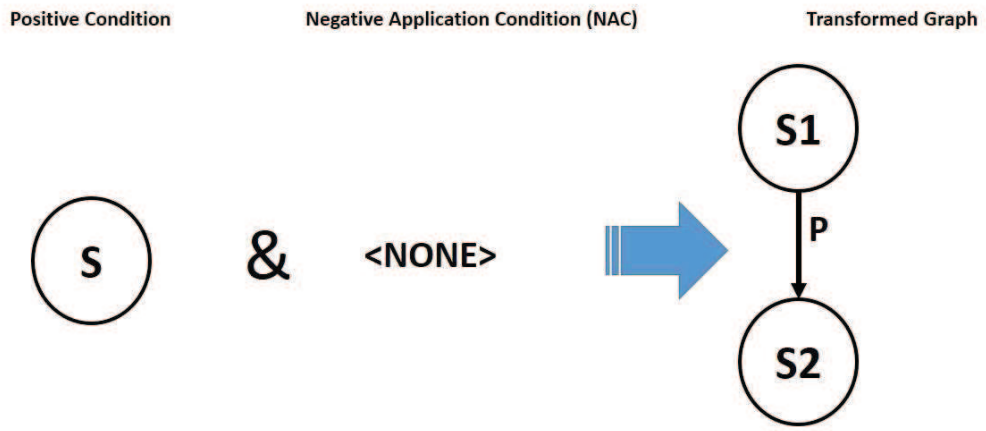

Figure 5: Learning Item Split Due to Too Much Material.

4.1.4 T4: Reduced Pressure Splitting Transform

Another way to reduce the cognitive load for an exam is to split an exam temporally. This leads to the transform shown in Figure 6.

4.1.5 T5: Change of Topic Detection Transform

In a standard classroom setting, a sequence of material on one subject will eventually give way to a change of topic and a new set of materials. We believe that this can be detected in the initial course graph due to the overlap pattern of related topics among the learning items. For example, a learning item that sits at the end of a sequence of topic-related items, and at the start of a distinct topic-related set of items, is most likely a transition item. This leads to the transform shown in Figure 7.

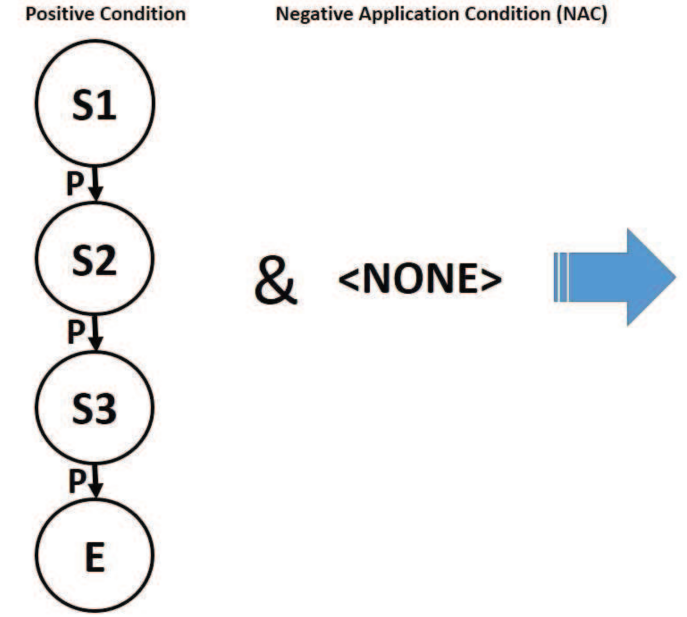
Positive Condition
Negative Application Condition (NAC)
Transformed Graph
S1
P
S2
P
S3
P
E
&
<NONE>
S1
P
P
E1
S2
P
S3
P
E2

Figure 6: Pressure Reducing Exam Split Transform.

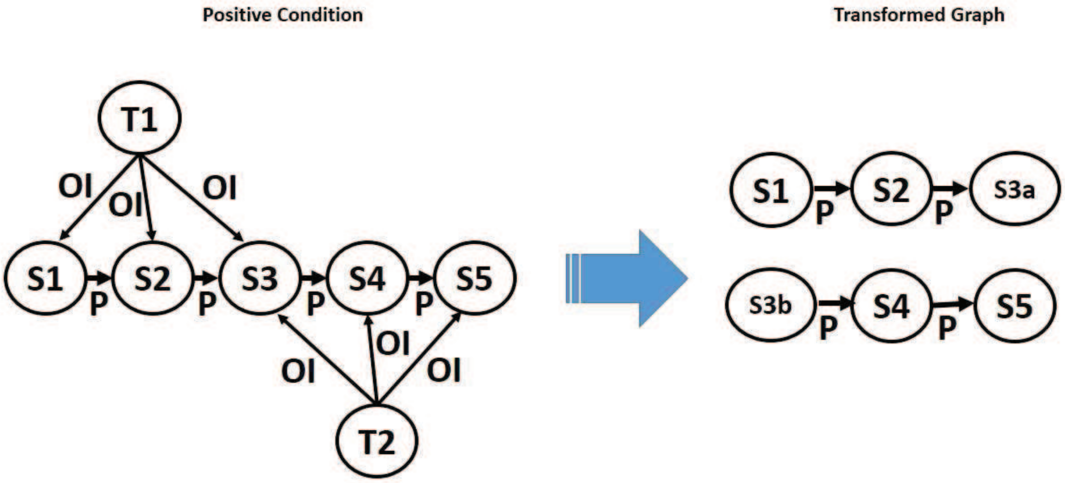
Positive Condition
Transformed Graph
T1
OI
OI
OI
S1
P
S2
P
S3
P
S4
P
S5
OI
OI
OI
T2
S1
P
S2
P
S3a
S3b
P
S4
P
S5

Figure 7: Topic Change Item Split Transform.

4.2 Applying Semantic Rules to the Graph

In AGG, transformations are defined by a rule with three parts (as described in the transformations given above):

- LHS (Left Hand Side): specify the pattern to find in the graph.
- NAC (Negative Application Condition): identify any restrictions to be imposed on the transformation. This part shows results that are not allowed. If the transformation would produce this result, the transformation is not applied.
- RHS (Right Hand Side) demonstrate what the pattern in the graph is to be after the transformation is applied.

These transformations embody the semantic rules listed above.

4.3 Applying T1: Topic-based *precedes* Elimination Rule

After the topic-based *precedes* elimination rule was first applied to the sample CS0 course, the instructor reviewed the resulting graph. Several unexpected findings were encountered.

4.3.1 Learning Items with No Topical Relations

There were five learning items that had no topic relations. Upon examination, two of the items were truly not related to any topic, an assignment in which students were to submit which team they wanted to be on, and the teacher evaluation. Both items were left unchanged with no topical relations.

The other three clearly were related to a specific topic but none of the terms in the topic list were found in the description. This could be remedied by using ENABLE's file upload tool that provides a way to add additional text to the description of a learning item. This also caused the instructor to consider the value of more frequently using the topic terms explicitly in the textual presentation of the learning items.

4.3.2 Learning Items with Meaningless Topical Relations

There were six topical relations that connected learning items to topics mistakenly. In five of these learning items, the topic words occurred but were being used in a more general way. For example, one of the topics is **content**. This is specifically related to selecting content when creating a web site. However, the word **content** was used in its more general way in three of the learning items. In the other case the instructions included a restriction to not use JavaScript which was a future topic. These relations were manually removed.

4.4 Applying T2: Topic-based Exam Splitting Rule

The first time the topic-based exam splitting rule was applied there were fewer exam splits than expected. Upon closer review it was discovered that the exam asked questions about a topic without using a topic word explicitly. This seemed pedagogically sound. For example, one question about computer science history was “Why was the invention of the integrated circuit important?” Although this question does not use the term history, it is clearly assessing the student’s familiarity with the computer science history covered in the course. These missing meaningful relations can included by adding text that includes the missing topic words to the description using ENABLE’s file upload tool. This adds the text to the ENABLE system without altering the exam itself.

There was one case where the review of the exam exposed the possibility of adding a word to the topic list. The word **occupation** was used in the exam that covered careers in computer science. This word was also used in other learning items about the topic. It was determined that adding this word to the topic list would add clarity. Adding the term to the topic list resolved this missing relation.

4.5 The Resulting Graph

Figure 8 shows the result of the application of the transforms *T1* and *T2*. The revised graph affords much greater leeway in the organization, presentation, and order of selection of material for the instructor and the student.

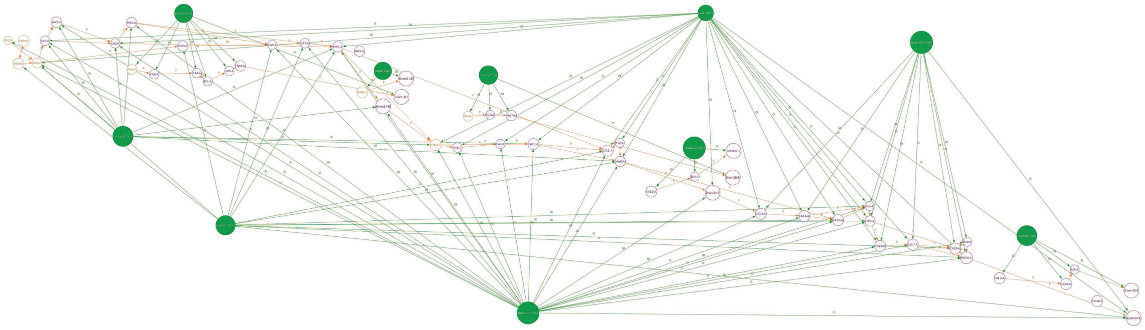

Figure 8: Course Graph After the Application of Transforms *T1* and *T2*.

5 Alternative Arrangements

Once the *T1* and *T2* transforms have been applied, many of the original organizational limitations have been removed. This opens the way for alternative arrangements of the learning items.

5.1 Separating by Topics

The graph in Figure 9 shows the learning items organized by topic. This arrangement separates the learning items in several distinct topic groups. The large group in the middle reflects the interrelated nature of several topics. This provides a visualization of how topics are related and how they might be rearranged. There is no visualization of *includes* relations.

5.2 Adding Unit Clustering

The graph in Figure 10 is clustered by units. The similarity between the graph arranged by topic and this one indicates that the units in the original organization grouped learning items into units by topic. Order of the units is not restricted. There are *precedes* relations between the five units in center of the graph. In the first and second grouping there are edges going in both directions.

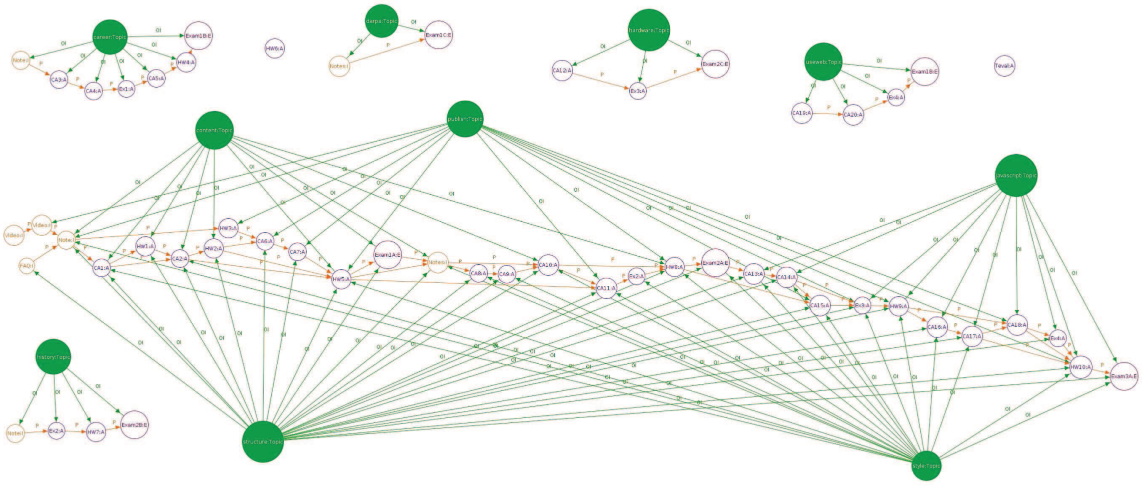
career:Topic
Note:I
CA3:A
CA4:A
Ex1:A
CA5:A
HW4:A
Exam1B:E
HW6:A
darpa:Topic
Notes:I
Exam1C:E
hardware:Topic
CA12:A
Ex3:A
Exam2C:E
useweb:Topic
CA19:A
CA20:A
Ex4:A
Exam1B:E
Teval:A
content:Topic
publish:Topic
javascript:Topic
Video:I
Video:I
FAQ:I
Note:I
CA1:A
HW1:A
CA2:A
HW2:A
HW3:A
CA6:A
CA7:A
HW5:A
Exam1A:E
Notes:I
CA8:A
CA9:A
CA10:A
CA11:A
Ex2:A
HW8:A
Exam2A:E
CA13:A
CA14:A
CA15:A
Ex3:A
HW9:A
CA16:A
CA17:A
CA18:A
Ex4:A
HW10:A
Exam3A:E
history:Topic
Note:I
Ex2:A
HW7:A
Exam2B:E
structure:Topic
style:Topic
OI
P

Figure 9: Course Graph Arranging Learning Items by Topic

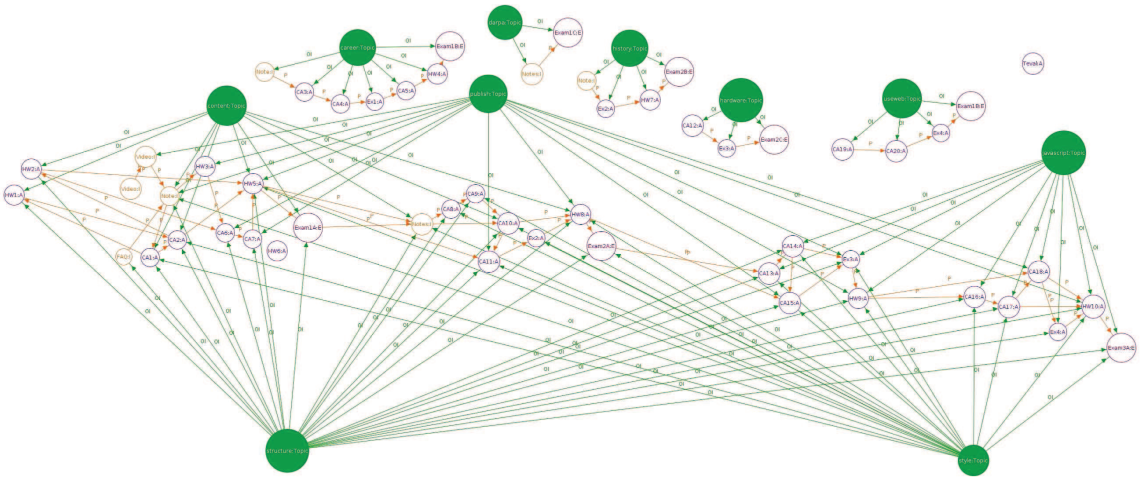
darpa:Topic
Exam1C:E
Notes:I
career:Topic
Exam1B:E
Note:I
CA3:A
CA4:A
Ex1:A
CA5:A
HW4:A
history:Topic
Note:I
Ex2:A
HW7:A
Exam2B:E
Teval:A
publish:Topic
content:Topic
hardware:Topic
CA12:A
Ex3:A
Exam2C:E
useweb:Topic
CA19:A
CA20:A
Ex4:A
Exam1B:E
javascript:Topic
HW2:A
HW1:A
Video:I
Video:I
HW3:A
HW5:A
Note:I
FAQ:I
CA1:A
CA2:A
CA6:A
CA7:A
HW6:A
Exam1A:E
Notes:I
CA8:A
CA9:A
CA10:A
CA11:A
Ex2:A
HW8:A
Exam2A:E
CA14:A
CA13:A
CA15:A
Ex3:A
HW9:A
CA16:A
CA17:A
CA18:A
HW10:A
Ex4:A
Exam3A:E
structure:Topic
style:Topic
OI
P

Figure 10: Course Graph Arranging Learning Items with Clustering by Units.

5.3 Informing change

There is information in these graphs that can be visually retrieved. Consider the following:

- How many topics occur in a specific assignment? This question can be answered by looking at how many *topic* edges come into an assignment. In the sample course, only one topic *occurs in* each of HW3, HW4, and HW7 while four topics *occur in* in each of HW5, HW8, HW9, and HW10.
- What units can be rearranged without interfering with *precedes* relations? The answer to this question can be found by looking at the *precedes* edges between unit clusters. For those units with no *precedes* edges between them the order can be changed without disrupting the temporal order restrictions expressed by these edges.

Having access to this kind of visual information has the potential to provide meaningful insights very quickly.

6 Conclusions and Future Work

We have developed an automated system that constructs an initial course organization graph based on information provided by Canvas, a standard Learning Management System (LMS). A variety of types of material are represented in the nodes of the graph and initially only their chronology is known. A detailed analysis of the materials based on the text contained within each learning item allows a more informed representation which captures the topic relations among the items. A set of graph transformations is then defined which convert the (basically) linear structure of the course to a graph structure which makes evident the dependencies and independencies of the learning items. A specific test case, CS0, was transformed in this way to demonstrate the power of the method.

Information about the existing course was provided by each step in this process. The ENABLE system gathered existing information from what was available about the course in the LMS and represented it in a visual way. This shed light on what students currently have available through their access to the learning materials in the LMS. The most significant finding was how entrenched the *precedes* relation is in the presentation of course material. This relation often adds little meaning to how learning items are related and yet it is the predominant organization strategy used when displaying information to students. When comparing the visual representation of that organization, see Figure 1 (upper) to the

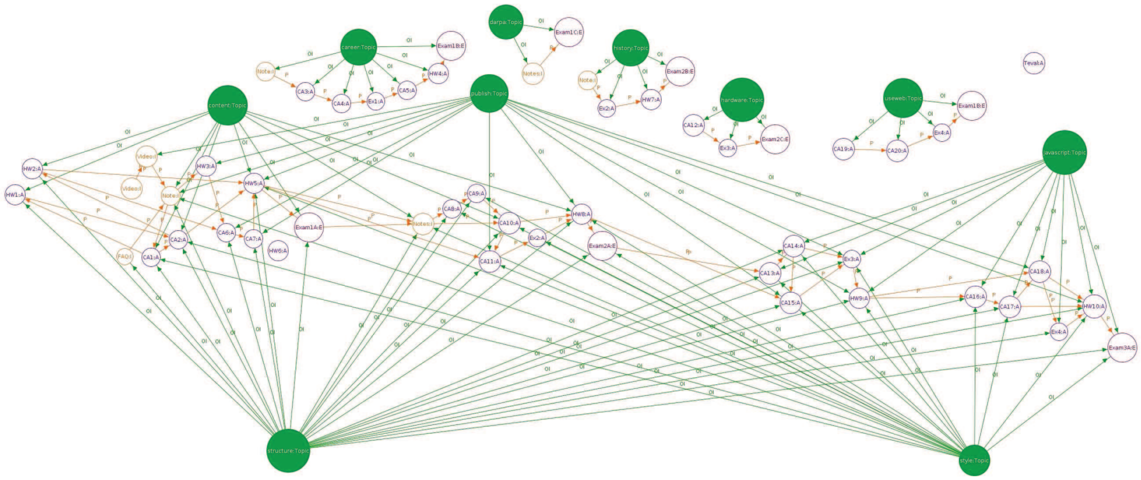
darpa:Topic
Exam1C:E
Notes:I
career:Topic
Exam1B:E
Note:I
CA3:A
CA4:A
Ex1:A
CA5:A
HW4:A
history:Topic
Exam2B:E
Note:I
Ex2:A
HW7:A
hardware:Topic
CA12:A
Ex3:A
Exam2C:E
useweb:Topic
Exam1B:E
CA19:A
CA20:A
Ex4:A
Teval:A
content:Topic
publish:Topic
javascript:Topic
HW2:A
HW1:A
Video:I
Video:I
HW3:A
Note:I
HW5:A
FAQ:I
CA1:A
CA2:A
CA6:A
CA7:A
HW6:A
Exam1A:E
Notes:I
CA8:A
CA9:A
CA10:A
CA11:A
Ex2:A
HW8:A
Exam2A:E
CA14:A
CA13:A
CA15:A
Ex3:A
HW9:A
CA16:A
CA17:A
CA18:A
HW10:A
Ex4:A
Exam3A:E
structure:Topic
style:Topic
OI
P

Figure 11: *precedes* vs. *prerequisite* Relations.

alternative organizations produced by ENABLE, see Figures 9 and 10, it is clear that there is significant room for improvement in how the educational community presents learning material to students. Although this first phase was designed to inform instructors about the many organization options available when making changes, the feeling of the authors is that the effort to develop a graphical, non-linear representation of a course could have significant impact on how the students perceive and interact with course materials.

During the course of this work, we determined that even though the *precedes* relations have been restricted to those that have common topic relations, they still express limited information. We believe that a more informative relation is the *prerequisite* relation that expresses a recommendation that one learning item be completed before another learning item. The *precedes* relation has one learning item directly following another learning item. This limits the connections between learning items and does not allow flexibility in ordering. It is easy to identify cases when this representation is too limited to express how the learning items are actually related. For example, there may be several learning items that are designed to prepare a student to complete a particular homework assignment such as a lecture, a class activity, a video, and a reading assignment. Using *precedes* relations, a graphical representation would look similar to that shown in Figure 11. Representing it this way indicates a specific ordering between the learning items when in fact this ordering is not required. The lecture, video, activity, and reading can be done independently of each other and it is not necessary to complete them in any particular order. A *prerequisite* order would better represent the meaningful relations as shown in Figure 11. Eliminating *precedes* relations in favor of *prerequisite* relations will provide a more accurate representation

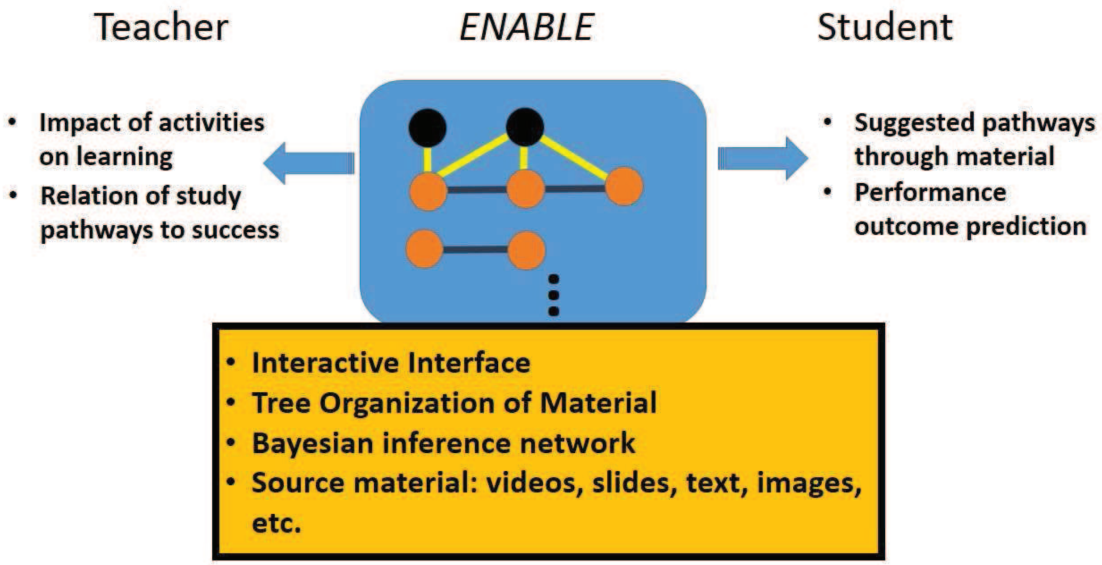
Teacher
ENABLE
Student
• Impact of activities on learning
• Relation of study pathways to success
• Suggested pathways through material
• Performance outcome prediction
• Interactive Interface
• Tree Organization of Material
• Bayesian inference network
• Source material: videos, slides, text, images, etc.

Figure 12: The ENABLE System.

of the course material. This will facilitate flexible organizations that become particularly valuable in asynchronous settings such as online courses, technical training, or competency based learning.

In the future we hope to transform learning outcomes by (1) facilitating deep student learning in science and engineering by providing the student feedback resulting from behavior models based on monitoring paths taken through the on-line course graph and linking that to performance in the class, and (2) providing effective tools for the instructor to monitor the effectiveness of the course material and its organization. The innovative use of a Bayesian inference network, a technology currently applied in many intelligent systems, will be developed and applied in a real-world learning environment to create a predictive computational model for individual learners and educators. Figure 12 shows our view of the final ENABLE system. By identifying operational student learning processes it may be possible to detect how knowledge gaps are a consequence of less successful learning strategies and tactics. Developing learning strategies can be challenging, thus an effective learning environment to support this must be designed and developed. As a next step, we propose to integrate a Bayesian inference network within the ENABLE system to provide synthesized data about learners' activities, behaviors, and performance. The instructor will be enabled to develop customized navigation aids for learners by (1) identifying first-order relationships between different learning behaviors and performance based on Bayesian network data, and (2) developing second-order relationships that reflect the underlying structure and the principles of the cues in first-order relationships.

The ultimate objectives of our research are to:

1. Create a rich graphical user interface that improves both the quality and quantity of student and teacher interaction with the learning material.
    - Conduct user testing at all stages of the system design, development, and testing to identify the usability of the interface and make revisions based on the results.
    - Create a graphical user interface that makes interaction with the ENABLE system available on mobile devices.
2. Create ENABLE from an existing interactive Learning Management System interface to support adaptive self-regulated learning so as to:
    - Identify (automatically) learning strategies (e.g., pathways through the material) that can improve student performance.
    - Provide early and ongoing prediction of student learning success in a course.
3. Provide ENABLE with a Bayesian inference network to support the teacher's effort to personalize self-regulated learning in order to:
    - Provide initial estimates of the value of each learning activity. The system would then provide feedback to the teacher about the actual impact of these activities on student learning success.
    - Make both predictions and learning strategies available to students and teachers.

We believe that the results presented here provide a good first step to the achievement of these goals.

References

[1] J. BÃ©zivin, F. BÃ¼ttner, M. Gogolla, F. Jouault, I. Kurtev, and A. Lindow. Model transformations? Transformation models! In *Model Driven Engineering Languages and Systems*, pages 440–453, 2006.

[2] S. Cutler, M. Borrego, M. Princeand C. Henderson, and J. Froyd. A Comparison of Electrical, Computer, and Chemical Engineering Faculty Progressions through the Innovation-Decision Process. In *Frontiers in Education Conference*, Seattle, Washington, October 2012. IEEE.

[3] I.S Dhillon. Co-clustering Documents and Words Using Bipartite Spectral Graph Partitioning. In *Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 269–274. ACM, August 2001.

[4] H. Ehrig, H. J. Kreowski, U. Montanari, and G. Rozendberg. *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 3. World Scientific Publishing Co., Inc.

[5] G. Taentzer C. Ermel and M. Rudolf. AGG: A Graph Transformation Environment for Modeling and Validation of Software. In *Handbook of Graph Grammars and Computing by Graph Transformation: Applications, Languages, and Tools*, pages 446–453, Heidelberg, Germany, 1999. Springer Berlin Heidelberg.

[6] G. Taentzer C. Ermel and M. Rudolf. The AGG Approach: Language and Tool Environment. In *Handbook of Graph Grammars and Computing by Graph Transformation: Applications, Languages, and Tools*, pages 551–603, London, UK, 1999. World Scientific Publishing Co., Inc.

[7] C.D. Manning, P. Raghavan, and H. SchÃ¼tze. *Introduction to Information Retrieval*. Cambridge, UK.

[8] X.H. Phan, L.M. Nguyen, and S. Horiguchi. Learning to Classify Short and Sparse Text & Web With Hidden Topics From Large-scale Data Collections. In *17th International Conference on World Wide Web*, pages 91–100, Beijing, China, April 2008. ACM.

[9] D. Rover, Y. Astatke, S. Bakshi, and F. Vahid. An Online Revolution in Learning and Teaching. In *Proceedings of the Frontiers in Education Conference*, Oklahoma City, Oklahoma, October 2013. IEEE.

[10] G. Taentzer, K. Ehrig, E. Guerra, J. de Lara, L. Lengyel, T. Levendovszky, U. Prange, and S. Varro-Gyapay. Model Transformation by Graph Transformation: A Comparative Study. In *Workshop Model Transformation in Practice*, Montego Bay, Jamaica, October 2005.

[11] J. Xu and W.B. Croft. Corpus-based Stemming Using Cooccurrence of Word Variants. *ACM Transactions on Information Systems*, 16(1):61–81, 1998.