

# Course Composition based on Semantic Topical Dependency

Javed I. Khan, Yongbin Ma, Manas Hardas  
Networking and Media Communications Research Laboratories  
Department of Computer Science, Kent State University  
233 MSB, Kent, OH 44242  
[javedlyma|mhardas@cs.kent.edu](mailto:javedlyma|mhardas@cs.kent.edu)

**Abstract:** *This paper describes an approach that can facilitate composition of course materials by machine assistance over the web using semantic web technology. A Semantic Topic Graph (STG) that encodes the conceptual knowledge metadata space is used as the front-end to the course resources. STG provides ontologies and properties of the concepts space and course materials are then connected with respect to their relation to this concept space. A course composer system is demonstrated which can intelligently compose various course materials based on pedagogical consideration. Some interesting comparisons of human guided machine-composed courses with real human composed courses in use in US graduate programs are shown.*

**Keywords:** *Semantic Topic Graph, RDF, Ontology, and Composition Algorithm*

## 1 Introduction

Although many courseware materials are now available online, but it seems their reuse has been surprisingly limited. What might be the dilemma? The reason seems to be that it is not easy to pedagogically redesign them. These do not come with the knowledge context. The lofty goal of creating sharable course material may never be successful if the knowledge map or the context behind cannot be shared as well. Interestingly, the recent advances in semantic network now provide a means to do that. In this research we show an approach where courseware materials are organized around a machine computable ontology. We then show a course design system which then can help human designer in designing courses with high level pedagogical guidance. There are recent attempts of online ontological courseware browsing libraries [1, 2, 3]. These keep annotated metadata as another layer to the real contents [1, 2, 3]. These first generation systems are suitable for human browsing but paradoxically are not easily machine computable. Even though they are concept (topic) based- the range of concepts is often very rich and concepts are not ranked in their knowledge domain [4,5]. We base our approach on a modified semantic net with a computable form of expressiveness based on topic dependency and equivalence. We then show its successful use in machine assisted design of courseware and share some composition work done by this system.

## 2 STG General Definition

Semantic Topic Graph is a directed graph  $G = (V, E)$ , where  $V$  is the set of concept nodes,  $E$  is the set of edges between concept nodes. Each concept node represents a

subject specific concept (such as “scheduling”, “thread”, “round robin schedule”, etc.) and acts an index to various course materials.

STG also has a weight structure. For each concept node, node weight is used to represent the relative importance of each concept; for each edge, edge weight is specified to represent the relative importance of the sub-concepts inside a particular concept. The sum of the prerequisite edge weights for a particular concept is always 1. Also they have associated pedagogical variables such as “teaching time”, “depth” etc.

## 3 Algorithmic Pedagogical Composition

The pedagogical process surrounding the design of a course, courseware, or conventional presentations undertaken by a teacher or presenter perhaps can be viewed as a three-step design process.

The first step is concept space selection. This is a global selection process. It can be poised as an optimization problem where each concept has some pedagogical value and also some formative cost. For example, if it is a presentation or course then perhaps there is a time limit or limit on the number of classes within which certain set of concepts have to be covered. If the composition goal is to produce a book or class note then there will be some form of page limit. The concepts have relative merits with respect to the pedagogical objective of the course. Thus we construct the problem such that each design sessions would like to maximize the pedagogical value with respect to a given formative constraint of the presentation. Two issues however, must be noted. First, the concepts have relative merit but not intrinsic merit. The values of the concepts are

also dynamic in the sense that their importance is particular composition is relative to other concepts taught or not taught in the composition. The concepts to be included also must satisfy somewhat complex dependency constraints. Thus the problem cannot be poised as a classical knapsack that picks from an unordered set of objects with fixed value. Below we show a modified Knapsack to accommodate the ordered and dynamic valued objects.

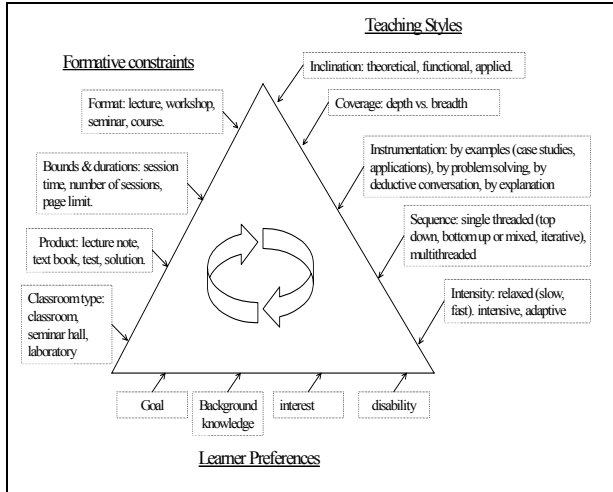


Fig. 1 Influential factors in course composition design

The second process that is also involved is the design of courseware is pedagogical styling. Though, it is a complex and iterative process but, we try to formulate this as a local selection problem to keep it computationally tractable. Each concept can be taught in various ways. Given the global constraints, here for each concept node, based on the style preference particular content is selected based on their style value. For example an applied course would like to include more application examples, than proof of properties, on a given topic. A theoretical focused course will perform the trade-off in the opposite way.

There is yet another process involved in courseware design. This is sequencing. Most learning activity occurs by a systematic presentation of the topics to be covered. The presentation is sequenced into linear thread. Complex teaching encounters allow for multiple teaching threads to continue in parallel. The design requires some topic area, topic bias, and quality goals. Figure-1 shows the influence factors in course composition design.

#### 4 Composition Evaluation

In this section we now define a set of graph theoretic criterion against which we can understand some properties of the composition and perhaps even compare

it against human composition in some pedagogical sense.

Given an input graph  $G(V, E)$ , we can get a sub graph under measurement,  $G_c(V_c, E_c)$ , a reference sub graph  $G_h(V_h, E_h)$  and the intersect graph  $G_\cap = G_c \cap G_h$ . Note:  $G_h$  could be obtained by other ways. We will evaluate course composition with structure similarity and structure deviation.

#### Structure similarity and Deviation

Structure Similarity and Structure Deviation show how much similarity the edges of the two graphs  $G_c$  and  $G_h$  have in common.

**Structure similarity**  $\xi_s$  is defined as the quotient between the weighted degree of connection of the nodes in  $G_\cap$  and the weighted degree of connection of the common nodes in the original graphs  $G_c$  and  $G_h$ .

$$\xi_s = \frac{2 \Xi(G_\cap)}{\Xi(G_\cap, G_c) + \Xi(G_\cap, G_h)} \quad (1)$$

where

$$\Xi(G_\cap) = \sum_{v_i \in G_\cap, v_j \in G_\cap} \omega_{e(v_i, v_j)}(G_\cap) / 2$$

$$\Xi(G_\cap, G_k) = \sum_{v_i \in G_\cap, v_j \in G_k} \omega_{e(v_i, v_j)}(G_k) - \Xi(G_\cap)$$

Where,  $\Xi(G_\cap)$  is the total weight of the edges in the graph  $G_\cap$  and  $\Xi(G_\cap, G_k)$  is the total weight of the edges in  $G_k$  with at least one end vertex is in  $G_\cap$ .

**Structure Deviation**  $\rho_s$  is similar to Node Deviation  $\rho_n$ . It is defined as the weighted number of edges that is not in the intersect graph  $G_\cap$ .  $\rho_s = 1 - \xi_s$ .

There are two kinds of deviations: Inclusion Deviation  $\rho_s(i)$  and Exclusion Deviation  $\rho_s(e)$ . Inclusion Deviation  $\rho_s(i)$  is the weighted number of edges that are in graph  $G_c$ , not in graph  $G_h$ ; Exclusion Deviation  $\rho_s(e)$  is the weighted number of edges that are in graph  $G_h$ , not in graph  $G_c$ .

$$\rho_s = \rho_s(i) + \rho_s(e) \quad (2)$$

$$\rho_s(i) = \frac{\Xi(G_\cap, G_c) - \Xi(G_\cap)}{\Xi(G_\cap, G_c) + \Xi(G_\cap, G_h)}$$

$$\rho_s(e) = \frac{\Xi(G_\cap, G_h) - \Xi(G_\cap)}{\Xi(G_\cap, G_c) + \Xi(G_\cap, G_h)}$$

### 5 Performance Evaluation & Results

#### 5.1 Graph Comparison

We get two kinds of graphs here: reference graphs and computer-selected graph.

## Reference graphs

The reference graphs are extracted from our STG manually with the information downloaded from internet. Two reference graphs will be used here. One is collected from Kent State University (KSU) and the other from The University of Tennessee (UT). Both of these two universities cover the same area “Advanced Algorithms” topic and use the same book Introduction to Algorithms, Cormen, H.T., Leiserson, C.E, Rivest, R.L. Figure 2 illustrates the table of contents respectively. Note: The real node name is represented here only by symbol here in this figure and below.

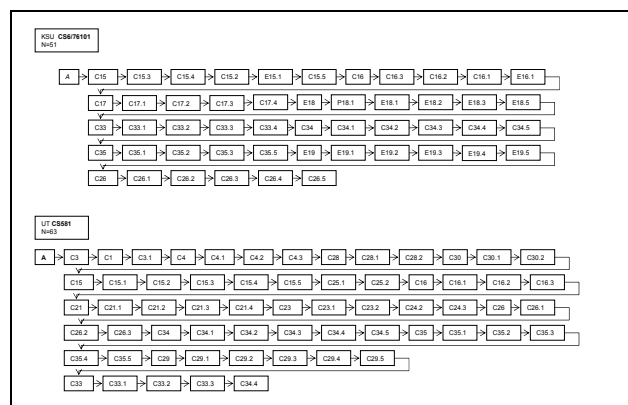


Fig. 2 Table of contents for Advanced Algorithm from KSU at the top part and UT at the bottom part. Total number of nodes is 51 and 63 respectively

## Computer selected graph

Computer selected graph is extracted from STG with our algorithm. After the client goes through whole process and inputs all parameters, the computer will automate to provide a graph for client. Figure 3 shows an example with 50 nodes and time constraints equal to 460 minute at the top part and there is thread linearization result at the bottom part.

Now let us see how a computer generated curriculum (CC) for the same topic of “Advanced Algorithm” compares to these two human generated (reference) curriculums. With two reference graphs (KSU and UT), we compare the organization of the course with respect to the course composer (CC) generated composition with time constraint= 460 minutes, and with unbiased seed topic set of {+’Advanced\_Algorithm’}. CC<sub>1</sub> is shown in figure 3. Table 1 summarizes the comparison result *structure similarity* between them respectively. As can be seen the structure similarity between the KSU and UT is only 66%. The structure similarity between the KSU and CC<sub>1</sub> is 76% and UT and CC<sub>1</sub> is 81%. From the comparison result between KSU and UT, we can get a general range of selection by human in real world.

Different people could have different opinion on the same topic in the same area.  $\xi_s = 0.66$ . The result just only shows the difference between different university and has only subjective meaning.

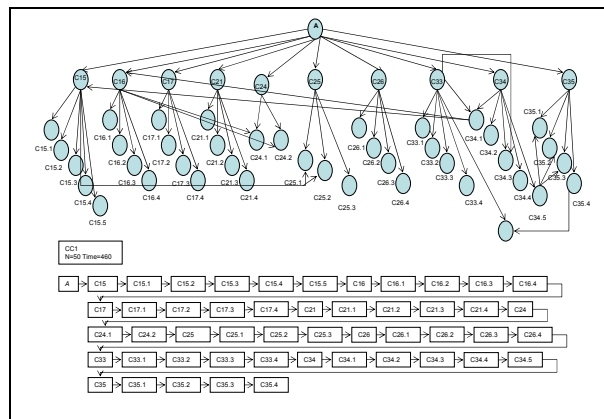


Fig. 3 CC<sub>1</sub> with node number 50 and time constraints=460minutes

## 5.2 Specialized/customized Composition

For particular users, because of their prior knowledge and previous courses covered, some topics can be excluded/Included encroachment from other courses. the weight of the edge to the topic node can be adjusted accordingly, so the course contents can be selected efficiently to meet their needs. With the same tool above, we can indeed analyze the customization performed in both of the universities. KSU has a strong program in Parallel Processing, and there are several subsequent courses that require students to know about parallel algorithms. Other factors are: graph theory has been covered in several other courses and the research area of the teacher is NP. So a customization with initial seed topic = {+’Advanced\_Algorithm’, +’Parallel\_Algorithms’, +’Exact Solutions to NP-Complete Problems’, -’graph theory’, -’disjoint sets’}. ‘+’ means inclusion, ‘-’ means exclusion. This returns the curriculum CC<sub>2</sub> in figure 4. The comparison is shown now in Table 2. CC<sub>2</sub> is similarity to KSU. The structure similarity is 96%.

Now we want to see what happens in UT syllabus? We enter seed topic = {+’Advanced\_Algorithm’, +’linear programming’, +’FFT’, -’Parallel\_Algorithms’, -’Amotized analysis’}. This returns the curriculum CC<sub>3</sub> in figure 5. The comparison results are summarized in Table 3. Now CC<sub>3</sub> is more like UT. The structure similarity between UT and CC<sub>3</sub> is 91%. Inside this table, the comparison between CC<sub>2</sub> and CC<sub>3</sub> is included.

From the result above, we can also tell which edge is most effective. The node to which the edge is connected is a key node. For the topic *Advanced of algorithm*, Key

nodes are *Dynamic programming; Greedy Algorithms; NP-Completeness; Approximation Algorithms; Computational Geometry*

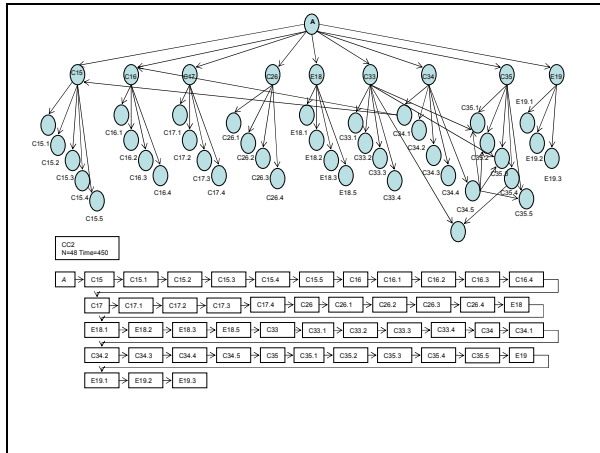


Fig. 4 CC<sub>2</sub> with node number 48

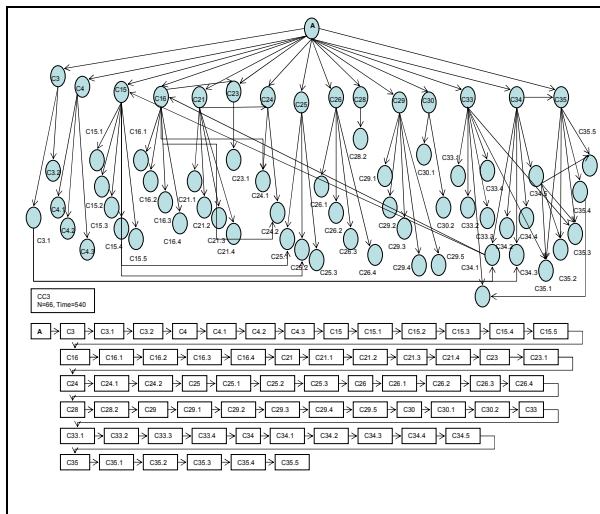


Fig. 5 CC<sub>3</sub> with node number 66

Table 1 Structural Similarity Analysis

	KS ↔ UT	KS ↔ CC <sub>1</sub>	UT ↔ CC <sub>1</sub>
Total Node	36 ↔ 40	40 ↔ 39	46 ↔ 44
Common	29	34	39
$\rho_s(e)$	0.19	0.14	0.11
$\rho_s(i)$	0.15	0.10	0.08
$\xi_{s_3}$	0.66	0.76	0.81

Table 2 Structural Similarity Analysis

	KS ↔ CC <sub>2</sub>	UT ↔ CC <sub>2</sub>
Total	46 ↔ 47	43 ↔ 40
Common	44	31
$\rho_s(e)$	0.01	0.15
$\rho_s(i)$	0.03	0.12
$\xi_{s_2}$	0.96	0.73

Table 3 Structural Similarity Analysis

	KS ↔ CC <sub>3</sub>	UT ↔ CC <sub>3</sub>	CC <sub>2</sub> ↔ CC <sub>3</sub>
Total	36 ↔ 42	63 ↔ 68	48 ↔ 42
Common	30	59	33
$\rho_s(e)$	0.16	0.03	0.16
$\rho_s(i)$	0.10	0.06	0.09
$\xi_{s_3}$	0.74	0.91	0.75

Inside these tables, *Total* means total number of edges in each graph. *Common* stands for common parts between two graphs.

## 6 Conclusions and future works

In this paper we have shown how based on knowledge ontology course can be composed. Previously we have shown how problem complexity can be assessed [6]. However, the actual design of courseware is very difficult and is still beyond the realm of any automatic system. Human considerations are indispensable in the design of learning activity. The suggested tools can only provide some objective assessment about the product of design. In future, computer aided pedagogical design is conceivable which can amplify the design capability of human engineers. It is interesting to note that currently there is plenty of courseware available for digital sharing. Unfortunately, their reuse is very low. The reason is probably the lack of good design help at pedagogical level. Given the inherent autonomous nature of learning activity, a potential future work is to develop a peer-to-peer model of courseware repository integrated with such design tools.

## 7 References

- [1] Farrell, R., Liburd, S.D., Thomas, J.C. Dynamic Assembly of Learning Objects, www 2004, ACM
- [2] Connexions: MathML and Collaborative Curriculum Development in Engineering <http://www.mathmlconference.org/2002/presentations/hendricks/>.
- [3] Dagger, Declan., Wade, Vincent., Conlan, Owen., (2004), Developing Adaptive Pedagogy with the Adaptive Course Construction Toolkit (ACCT), published in the workshop proceedings of the AH'04 conference (online), AH2004
- [4] Muñoz, L.S, José Palazzo Moreira de Oliveira. Adaptive Web-Based Courseware Development Using Metadata Standards and Ontologies. The 16th International Conference on Advanced Information Systems Engineering - CAISE , June 7-11 2004, Riga, Latvia.
- [5] Schraefel, M.C., Shadbolt, N. R., Gibbins, N., Glaser, H., Harris, S., University of Southampton, CS AKTive Space: Representing Computer Science in the Semantic Web, [page 384], WWW2004
- [6] Javed I. Khan, Manas Hardas, Yongbin Ma, A Study of Problem Difficulty Evaluation for Semantic Network Ontology Based Intelligent Courseware Sharing, Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, Compiegne, France, September 2005, pp426-429