 ISBN

A comparison of the core aspects of the ACM/IEEE Computer Science Curriculum 2013 Strawman report with the specified core of CC2001 and CS2008 Review

Linda Marshall
Department of Computer Science
University of Pretoria
Pretoria, South Africa
Email: lmarshall@cs.up.ac.za

***Abstract*—With the changes in the structure of the proposed volume for Computer Science of the ACM/IEEE curricula series for 2013 it is difficult to conceptualise what the extent of these changes are when reading the Strawman report. This paper is an attempt to quantify these changes. It discusses the core curricula as defined in the ACM/IEEE curricula volumes for Computer Science of 2001, 2008 Review and the 2013 Strawman report and shows how a real-world curriculum can be compared to the volumes.**

Both a visual comparison for the curricula volumes and an algorithmic comparison are presented. The visual comparison provides a quick indication that there are differences, while the algorithmic comparison provides a mechanism to quantify the differences and as a side effect the similarities. These comparisons are restricted to the core aspects of the curricula being evaluated and further to this subsections of the core curricula will be used for illustrative purposes in order to contain the complexity of curricula specifications.

I. Introduction

WITH the rate at which technology changes, it is no surprise that the ACM/IEEE Computer Science Curriculum volume needs to change too. Every so often, a more up to date version of the volume is released and the changes between this volume and the previous volume are provided in the curriculum volume report. From the description of the changes, it is very difficult to judge what the impact of these changes would be on a curriculum that has been implemented and which complies with the previous version of the volume. It is the purpose of this paper to illustrate how this kind of information can be represented in an understandable way in order for curriculum developers to easily be able to see what the impact of the changes are on their curricula.

Prior to the ACM/IEEE Computing Curricula of 1991, the ACM and the IEEE-CS worked independently on computing curricula recommendations. In the spring of 1988, the Joint Curriculum Task Force, comprising of the ACM and the IEEE-CS, was formed to develop curriculum recommendations for computing. The outcome of this partnership was the *Computing Curricula 1991* report [1]. Ten years later, the *Computing Curricula 2001* report was released. This was followed by the 2005 overview report which identified different disciplines within computing [2] and the *Computing Curricula 2001* was included in the overview report and renamed *Computer Science 2001* to better reflect the discipline it represented. The latest version of the Computer Science curricula volume is under review and will be released towards the end of 2013.

For the purposes of this study, the curricula volumes for Computer Science of the ACM/IEEE of 2001, 2008 Review and 2013 Strawman are considered. All three volumes use the same basic terminology when describing the structure of curricula and all three focus specifically on computer science as a discipline. The paper focusses on the core requirements of the curriculum volumes, leaving the elective requirements for a later study.

To make the comparison, it is necessary to model the curricula in a uniform manner, this is achieved by modelling the structure of the ACM/IEEE curricula discussed in section II as a graph, refer to section IV. Comprehending the sheer size of the curricula volumes can be daunting and a basic indication of the core aspects for comparison are given in section III. The comparison between the modelled volumes is presented in section V in which a visual comparison is presented before an algorithmic finer granule comparison is given. In order to restrict the complexity of the graph, the results will be discussed on the knowledge area level. Section VII provides a quick overview of how the techniques used to compare the curricula volumes can be used to compare the compliance of a real-world curriculum that is being presented at a university to the volumes. Concluding remarks are presented in section VIII.

II. ACM/IEEE Curriculum Structure

For each discipline specified by the *ACM/IEEE Computing Curricula 2005* [2], the contents of the discipline is defined in terms of Knowledge Areas (KAs), Knowledge Units (KUs) and topics [3]. In some cases sub-topics have been identified, but all these can still be seen as part of the topic sphere. KAs can be shared between disciplines, but it is more likely that a subset of KUs within a KA may be shared. There is a linear progression, and it is possible that a topic may resort under more than one KU and therefore also more than one KA.

The last complete Computer Science Curricula volume was released in 2001 (hereafter CC2001). In 2008 (hereafter CS2008), a review of the 2001 curriculum volume was released in which there was a strong focus on Computer Security [4]. In both the 2001 and 2008 curricula volumes, KUs were identified as being either *core* or *elective* [5], [4]. Core KUs and their respective topics are required in any curriculum that is developed for the discipline. The minimum number of hours that should be spent on the KU is also specified. Elective modules on the other hand are optional to the curriculum being developed. The 2013 strawman volume (hereafter CS2013S) [3] moves the designation of *core* or *elective* to the topic level and distinguishes between two types of core topics . The first is referred to as *core tier 1* (CT1) and the second as *core tier 2* (CT2). As topics are now labeled as being core (either CT1 or CT2) or elective, they are grouped within the KUs. These clustering of topics are allocated the minimum number of hours that should be spent on the topics within the cluster for all core topics. A further requirement in CS2013S is that all topics in CT1 must be included in a curriculum, while at least 90% (with an 80% bare minimum) of CT2 is considered essential to be included in a Computer Science curriculum.

TABLE I
KNOWLEDGE AREAS

KA	CC2001	CS2008	CS2013S
AL*	Algorithms and Complexity		
AR*	Architecture and Organisation		
CN**	Computational Science and numerical methods		Computational Science
DS	Discrete Structures		
GV	Graphics and Visual Computing		Graphics and Visualisation
HC**	Human-Computer Interaction		
IAS			Information Assurance and Security
IM**	Information Management		
IS**	Intelligent Systems		
NC	Net-centric Computing		Networking and Communication
OS*	Operating Systems		
PBD			Platform-Based Development
PD			Parallel and Distributed Computing
PF**	Programming Fundamentals		
PL*	Programming Languages		
SDF			Software Development Fundamentals
SE*	Software Engineering		
SF			Systems Fundamentals
SP*	Social and Professional Issues		Social and Professional Practice

* SA defined in CC1991
** CC1991 SA included in the KA

III. COMPARISON BY NUMBERS

CS2013S suggests an increase in KAs from 14 in CC2001 and CS2008 to 18 [6]. From table I it is evident that there were major changes to some of the KAs. KAs AL, AR, DS, HC, IM, IS, OS, PL and SE remain the same. KAs CN, GV, NC and SP changed focus, but not enough to drop them entirely. The PF KA was dropped in CS2013S and IAS, PBD, PD, SDF and SF were introduced. The six (6) KAs marked with an asterisk (*) map directly to subject areas (SAs) in Computing Curricula 1991 (hereafter CC1991). The other 5 SAs defined in CC1991 have been included in KAs marked with two asterisks (**). The most significant changes between CC1991 and CC2001 are the addition of KAs that focus on Discrete Structures, Computer Graphics and Net-centric Computing.

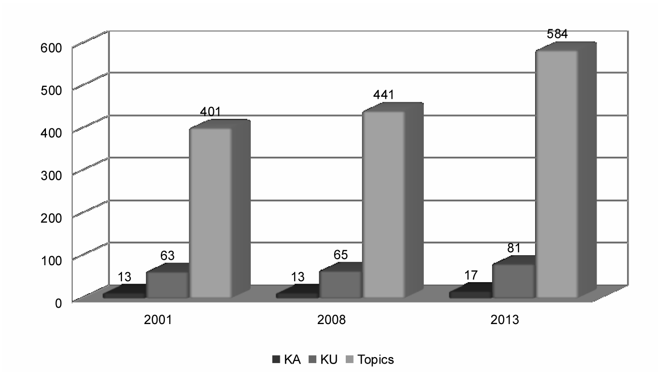

Fig. 1. Categories that represent core content

The number of core topics between the volumes also steadily increased. In figure 1, the number of KAs in which core topics reside, KUs labelled as core in CC2001 and CS2008 and those with core topics in CS2013S, and core topics from CS2013S are given per curriculum volume. From the figure it is evident that there has been a marginal, between 3% and 9% increase, in topics between CC2001 and CS2008. However, between CS2008 and CS2013S, there has been a marked increase in all aspects - 24%, 20% and 24% in KAs, KUs and topics respectively.

When comparing the topics and looking at where they resided over the three curricula volumes, it is interesting to note that only 11% are described using exactly the same phrase, 24% remained constant across two volumes and the remaining 65% are unique to a volume. This fact may be the reason why it is so difficult to compare a curriculum that has been developed and evolved over time to the specification provided by the curricula volumes over the years (refer to section VII for an example).

IV. MODELLING CURRICULA VOLUMES AS GRAPHS

To be able to effectively compare the curricula volumes it is necessary to model them in a uniform manner. The volumes have already defined a linear path between KAs and the topics. Taking this as a basis and including all the KA to topic paths in a single structure will result in a tree structure.

Consequently, with this approach a curriculum volume is modelled by using an onion as the metaphore. In the center is the discipline, in this case Computer Science (CS), the next ring represents the KA's, followed by the KU's and then the topics. As stated in section II, some topics are further broken down into subtopics, but for modeling purposes, these are simply modelled as topics that are related to other topics. There are instances where topics resort under more than one KU which effectively results in a curriculum being modelled as a directed graph with the nodes representing the KA, KU or topic information and the edges the relationships between the nodes.

Once the graph has been defined, the structure and the content of one curriculum volume can be compared with one another.

V. What do the graphs tell us?

It is always revealing when comparing graph structures visually as to what the differences or similarities between the graphs may be. This inspection can be used to guide further investigation into interesting areas in which an algorithmic comparison can provide the details of. The sections that follow illustrate the use of the visual inspection and discusses the results of an algorithmic approach.

A. Visual representation

By visually representing the graphs, it is possible to determine what the similarities and differences between the graphs are. The visualisation *per se* does not provide any concrete information regarding the comparison of the graphs, it does however provide the viewer with an idea of where to look as well as the degree of similarity or difference.

In order to illustrate the significance of a visual representation, a subsection (in terms of KAs) of the graph will be used due to space constraints. Consider figures 2 and 3 which represents the Architecture and Organisation KA and the Human-Computer Interaction KA respectively. The dotted edges show the relationships between the KA node, KU nodes and topic nodes of CC2001. The dashed edges represent CS2008 and the solid edges CS2013S.

Without knowing what each node explicitly represents it is clear from figure 2 that the Architecture and Organisation KA of CS2008 is vastly different from that of CC2001 and CS2013S. Without a doubt the paths from KA to topic defined in CS2013S has significantly reverted back to what it was in CC2001.

From figure 3 it can be seen that one KU of the Human-Computer Interaction KA has stayed relatively stable across all three volumes. A significant KU has been included in CS2013S. Only one topic from the isolated CC2001 dotted branch was left in CS2008 dashed branch. This dashed branch has further been left out of the CS2013S volume, solid line.

B. Algorithmic comparison

With the curriculum volumes modelled as graphs, it is possible to compare the content and structure of the volumes with one another. An overview of the algorithm used to enable this comparison is given by algorithm 1 [7].

The algorithm accepts two graphs as parameters, G_1 and G_2, both representing graphs within the domain $\mathcal{G}$. The

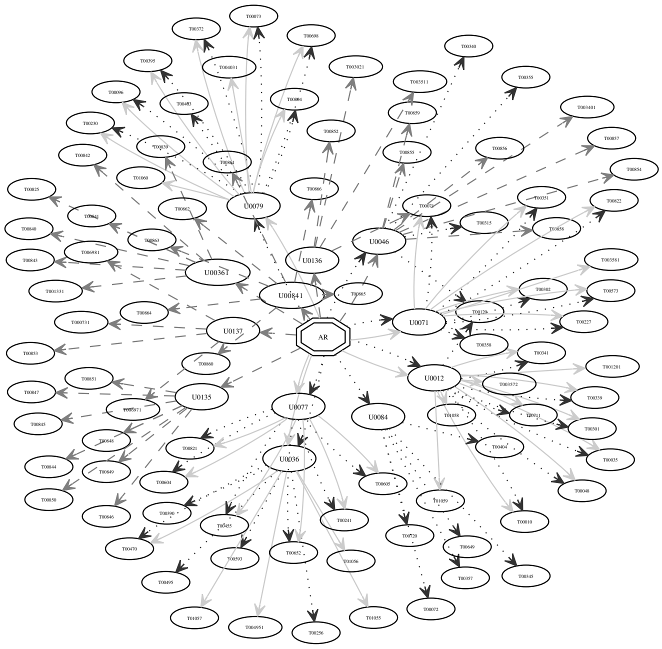

Fig. 2. Representation of the Architecture (AR) KA

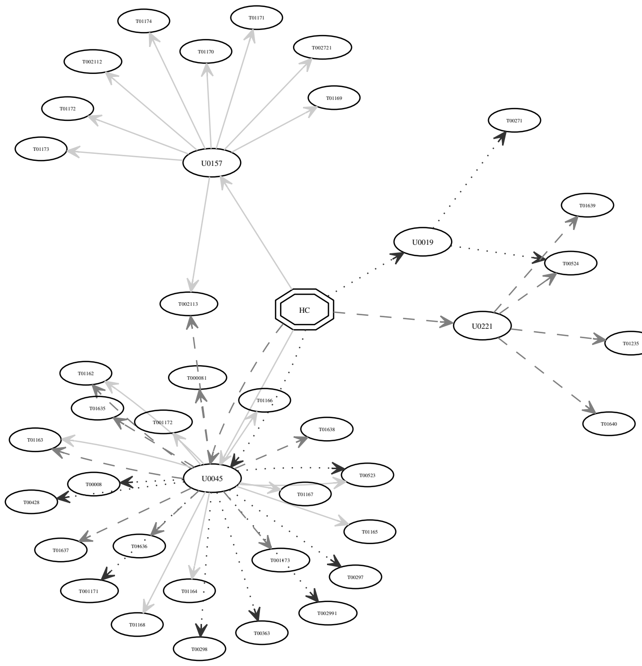

Fig. 3. Representation of the Human-Computer Interaction (HC) KA

Algorithm 1 Algorithm for comparison of two graphs

Require: $G_1, G_2 \in \mathcal{G}$
Ensure: $G_3 \in \mathcal{G}$
  $G_3 = \emptyset$
  for every node pair in G_2 **do**
    determine if there is a path in G_1 for this pair
    if a path is found in G_1 **then**
      add the path to G_3
    end if
  end for
  return G_3

TABLE II
ALGORITHMIC COMPARISON OF GRAPHS

	2001&2008		2001&2013		2008&2013	
	Vertices	*Edges*	*Vertices*	*Edges*	*Vertices*	*Edges*
G_1	476	477	476	477	510	519
G_2	510	519	682	682	682	682
G_3	356	356	191	191	183	184
G_1-G_2	138	248	301	350	344	391
G_1-G_3	120	121	285	286	327	332
G_3-G_2	18	127	16	64	17	59
G_2-G_3	172	290	507	555	516	554

algorithm computes a third graph that is defined by using the nodes in G_2 and the structure of G_1 to guide the process. The resultant graph, G_3, is therefore also in the domain $\mathcal{G}$. It follows that the resultant graph, G_3, is a representation of the second graph in terms of the first.

A comparison of the nodes and vertices of the 3 graphs will give an indication of how compatible the second graph is with the first. In order to achieve this, the third graph is compared with the first. If the first and third graphs are exactly the same, then it can be said the the second graph exactly represents the first. In the event that they differ, the differences can be quantified to give an indication of the degree that they differ and the exact nodes and edges can be investigated.

Table II gives the results for the comparison of the graphs of CC2001, CS2008 and CS2013S for the combinations: CC2001 with CS2008, CC2001 with CS2013S, and CS2008 with CS2013S in terms of edges and vertices as shown in the table columns. The first 3 rows of the table give the edge and vertex count for each of the graphs. The subtraction of the graphs, represented by $A - B$ in the table, provides insight into what the *nett* result would be if all aspects of B are taken out of A. If $A - B = 0$, then the two graphs represented by $A \in \mathcal{G}$ and $B \in \mathcal{G}$ are identical in structure and content.

VI. ALGORITHMIC RESULTS

The algorithmic results are presented in terms of numbers as well as overlap. The numbers (or size) will give an indication of the order of magnitude, while overlap will present a more exact indication of similarity and differences. Each of these aspects are presented individually in the sections that follow.

TABLE III
RELATIVE SIZE COMPARISON OF VERTICES IN G_3

	2001&2008	2001&2013	2008&2013
G_1	75	40	36
G_2	70	28	27

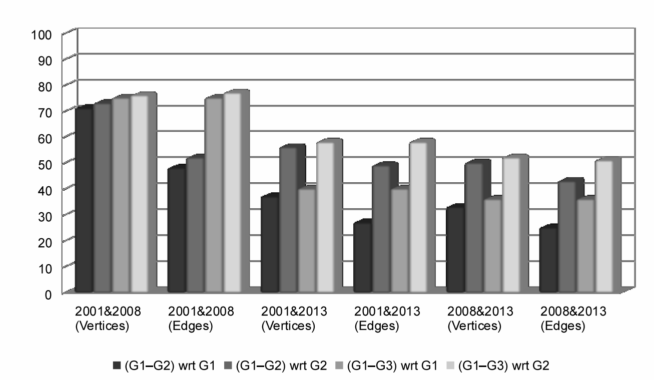

Fig. 4. Overlap between graphs

A. Size

By only considering the number of vertices in the resultant graph G_3 in relation to the number of vertices in the original graphs, G_1 and G_2, it is immediately evident that the core curriculum as defined in CS2013S is vastly different from that of CC2001 and CS2008 (refer to table III). Between CC2001 and CS2008 the relative number of nodes in common between the curricula was 70% or more, indicating some adjustments to topics and KUs as KAs stayed the same. It is interesting that the percentages between CC2001 and CS2013S are marginally higher than those between CS2008 and CS2013S. This can mainly be attributed to KAs reverting back to topic phrasing from CC2001 in CS2013S. The fact remains, that a lot has changed in CS2013S and only close on 40% of both the CC2001 and CS2008 remain in CS2013S. This is line with what was seen in the visual inspection of AR, discussed previously. The percentages of G_3 in relation to G_2 are as expected. G_3 has been moulded to represent G_2's information in terms of G_1 for the purpose of direct comparison. The results for the edges form a similar pattern.

B. Overlap

A more exact measure is to determine the percentage of nodes that overlap between two graphs. The overlap can easily be determine by using the difference $A - B$ as explained in section V-B. The difference determines the nodes in graph A that are not in graph B and therefore $100 - ((A - B)/n\%)$, for $n = A|B$ represents the percentage overlap of $A - B$ with respect to A or B.

From figure 4, it is clear that that there is a greater overlap between CC2001 and CS2008 than either CC2001 or CS2008 with CS2013S. The pattern for both CC2001 and CS2008 when compared with CS2013S has a similar profile, however the values of CS2008 and CS2013S are lower than the CC2001 counterpart. This signifies once again that there is more overlap between CC2001 and CS2013S than with CS2008 and CS2013S. As expected, the values for $G_1 - G_2$ with respect to both G_1 and G_2 is lower than the values of $G_1 - G_3$. It can therefore be concluded that building G_3 using the nodes of G_2 and the structure of G_1 results in a fair comparison of the content of the graphs.

VII. Application to a real-world curriculum

Real-world curriculum specifications do not necessarily follow the same structure of KAs, KUs and topics as presented in the curriculum volumes. Most curricula span over a number of years, in each year the student is required to take a number of modules. Each of these modules are described in terms of descriptions, more often than not comprising of keywords related to the module. These keywords can be mapped back to topics specified in the ACM/IEEE curricula volumes.

By translating the curriculum to a graph and the making use of the algorithm 1, it is possible to compare the curriculum with that specified in the ACM/IEEE curriculum volumes. In order to do this G_1 in the algorithm is the graph representation of the ACM/IEEE curriculum volume and G_2 the representation of the real-world curriculum in terms of years, modules and topics. The resultant graph G_3, which will be referred to as G_4 in this section to avoid confusion with the resultant graph in the previous sections, is the graph of the topic information presented in G_2 moulded in the form of G_1 and therefore making the comparison with the ACM/IEEE curriculum volume possible.

Both a visual comparison of G_4 with respect to G_1 and the algorithmic comparison will be given in the sections that follow. The real-world curriculum, represented by G_2, was developed using the core aspects of CC2001 as a basis.

A. Visual comparison

A visual comparison of G_4, defined in terms of CC2001 (dotted edges), CS2008 (dashed edges) and CS2013S (solid edges), is given in figures 5 and 6 for the KAs AR and HC respectively.

From figure 5 it is evident that CS2013S has reverted back to how AR was defined in CC2001. CS2008 however is minimally represented by the curriculum in the real-world curriculum. Furthermore, two branches, representing KUs in the ACM/IEEE curriculum volumes, of the real-world curriculum that were present in CC2001 have completely disappeared from CS2013S. Further inspection of the KUs reveals that the knowledge unit "Multiprocessing and alternative architectures" (top left branch in the figure) was renamed in CS2008 to "Multiprocessing". In CS2013S it was dropped as a core KU. The KU "Functional Organisation" (bottom left branch in the figure) is defined as an elective in CS2013S. What is interesting is that it is still defined as core in CS2008, but with a redefined topic list. This indicates that the graph model will require a meta-data model that reflects the changes between the naming conventions of the entities in the curricula volumes.

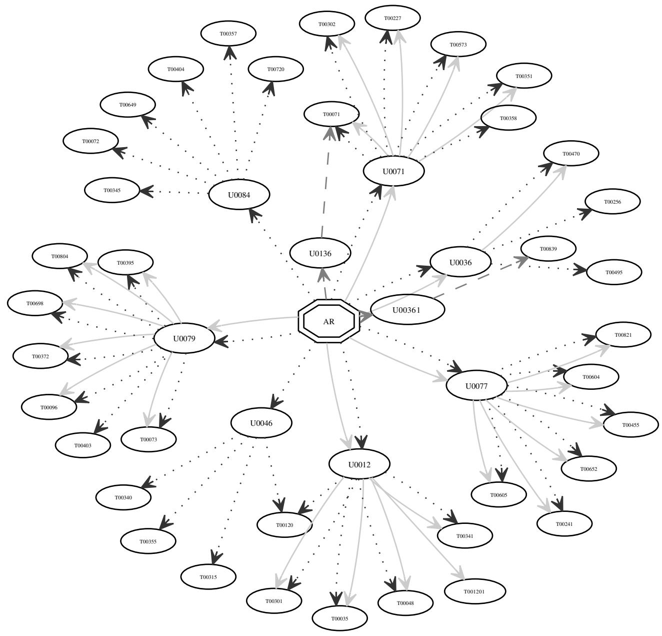

Fig. 5. Real-world AR coverage over the volumes

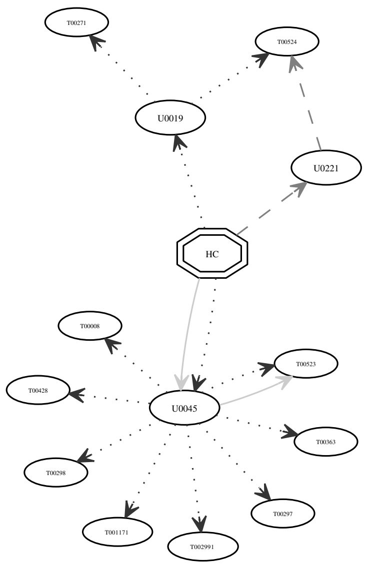

Fig. 6. Real-world HC coverage over the volumes

Figure 6 presents a similar picture for HC as what was seen in AR. The KU "Foundations of human-computer interaction", represented by the branch in the bottom of the figure, is present in both CC2001 and CS2013S. Again it is defined in CS2008, but the topic list associated with it is different from that in CC2001. This KU is represented by the bottom left branch in figure 3, and from this it can be seen that CC2013S make use of both the topics updated in CS2008 and those originally found in CC2001.

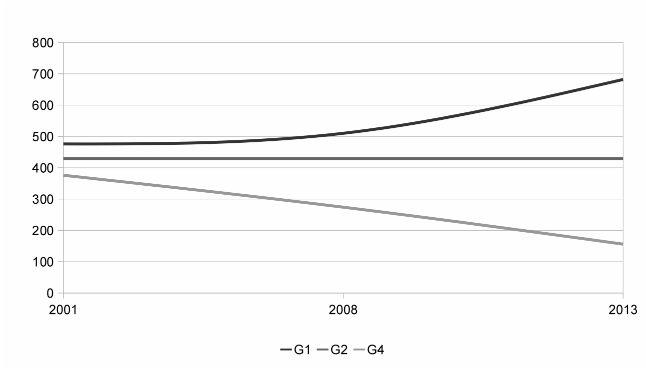

Fig. 7. Comparison of a real-world curriculum

TABLE IV
RELATIVE SIZE COMPARISON OF VERTICES IN $G_1 - G_4$

	2001	2008	2013
$G_1 - G_4$	100	236	526
$G_1 - G_4$ as a % of G_1	21	46	77

B. Algorithmic comparison

By comparing the core aspects, in terms of number of nodes, of the curriculum represented by G_2 and designed using CC2001 as a basis to each of the curricula CC2001, CS2008 and CS2013 (represented by G_1) algorithmically (using algorithm 1), there is a definite trend towards non-compliance. As the number of nodes in the respective curriculum volumes increases, the compliance of the real-world curriculum decreases (represented by G_4). This can be attributed to the changes over the years that have taken place in the curricula volumes. Where the compliance of G_2 to CC2001 was 79%, the compliance to CS2008 drops to 54% and alarmingly to 23% for CS2013. Figure 7 illustrates this trend.

Determining the exact similarities and differences between the graph representations can be done by using the subtraction of graph, $A - B$, values and results for overlap as discussed in section VI-B. $G_1 - G_4$ is the most significant result. It will determine the elements of the graph (vertices and edges) that are in G_1 but not in G_4 resulting in a list that can be used to update the curriculum represented by G_2. For the example, the number of vertices for $G_1 - G_4$ is given in table IV along with the percentage.

VIII. CONCLUSION

The numbers, which are easily extracted from the respective curricula volumes and presented in section III provide the first indication of the changes between CC2001 and CS2008 Review volumes and the proposed CS2013S volume. Further investigation, at a lower level of granularity, highlights more in terms of the differences between the curricula volumes. Both the visual and algorithmic results presented for the core curricula of CC2001, CS2008 Review and CS2013S conclude that the content of a Computer Science core curriculum has changed dramatically in the last 11 years. These changes can be attributed to the use ubiquitous use of computing technologies. By comparing the curricula at a lower level of granularity, that is the exact vertices and edges that differ, than has been presented here, will highlight changes in the focus of the ACM/IEEE curricula volumes. Some of these changes, such as the focus on security on CS2008, have explicitly been recorded in the volumes, however more subtle changes can also be picked up.

Providing exact information will form part of a followup study along with determining the similarities and differences with regards to the elective topics. When capturing the data for the algorithmic comparison, it is evident that some of the previously elective topics have been elevated to core in the CS2013S version. The reverse is also be true, some core topics have been demoted in CS2013S to elective. This was highlighted in the findings given in section VII-A. In order to capture this information it will be necessary to include metadata in the graphs. Examples of this are: whether a topic is a core (including its hours) or elective; the number of hours allocated to core topics; and, the progression between volumes of how KAs, KUs and topics changed and their relationships with each other. This will lead to a process by which the curricula volumes information is refactored and increasing the overlap in terminology increases from 11% to at least 80%. The refactoring process will need to ensure that the representation of the content of the curricula per volume is consistent and that nothing is lost. Once this has been done, it will enable a better comparison between the versions to take place. The refactored curricula specifications can also be compared to the computing ontology being developed by Cassel's team [8].

REFERENCES

[1] ACM/IEEE-CS Joint Curriculum Task Force, "A Summary of the ACM/IEEE-CS Joint Curriculum Task Force Report - Computing Curricula 1991," *Commun. ACM*, vol. 34, no. 6, pp. 68–84, Jun. 1991. [Online]. Available: http://doi.acm.org/10.1145/103701.103710

[2] Joint Task Force for Computing Curricula, "Computing Curricula 2005: The Overview Report covering undergraduate degree programs in Computer Engineering, Computer Science, Information Systems, Information Technology, Software Engineering," 2005.

[3] The Joint Task Force on Computing Curricula Association for Computing Machinery IEEE-Computer Society, "Computer Science Curricula 2013 Strawman Draft," 2012.

[4] ACM/IEEE-Curriculum CS2008 Joint Task Force, "Computer Science Curriculum 2008: An Interim Revision of CS 2001," 2008.

[5] ACM/IEEE-Curriculum 2001 Task Force, "Computing Curricula 2001: Computer Science," 2001.

[6] M. Sahami, S. Roach, E. Cuadros-Vargas, and D. Reed, "Computer science curriculum 2013: reviewing the strawman report from the ACM/IEEE-CS task force," in *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, ser. SIGCSE '12. New York, NY, USA: ACM, 2012, pp. 3–4. [Online]. Available: http://doi.acm.org/10.1145/2157136.2157140

[7] L. Marshall and D. Kourie, "Deriving a digraph isomorphism for digraph compliance measurement," in *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*, ser. SAICSIT '10. New York, NY, USA: ACM, 2010, pp. 160–169. [Online]. Available: http://doi.acm.org/10.1145/1899503.1899521

[8] L. N. Cassel, R. H. Sloan, G. Davies, H. Topi, and A. D. McGettrick, "The computing ontology project: the computing education application," in *SIGCSE*, 2007, pp. 519–520. [Online]. Available: http://doi.acm.org/10.1145/1227310.1227486