

A Visual Analytics Approach to Gain insights into the Structure of Computing Curricula

Musa Jafar

musa.jafar@manhattan.edu

Computer Information Systems & Business Analytics

Manhattan College

Riverdale, New York 10471, USA

Leslie J Waguespack

lwaguespack@bentley.edu

Computer Information Systems

Bentley University

Waltham, Massachusetts 02452, USA

Jeffrey S. Babb

jbabb@wtamu.edu

Computer Information and Decision Management

West Texas A&M University

Canyon, Texas 79016, USA

Abstract

In their effort to keep the computing curriculum relevant, competitive, and reflective of market needs and provide clear benchmarks for accreditation, the computing societies propose curriculum guidelines for degree granting programs to define and promulgate their disciplinary vision. It is difficult to develop a comprehensive understanding of the scope of computing curricula or the orientation of any particular curriculum within that scope due to the curricular complexity (i.e. Bodies of Knowledge, Areas of Knowledge, taxonomy of topics, levels of coverage, learning outcomes, professional competencies). Visual analytics tools (tableau, ggplot, d3js, etc.) as well as visual analytics methods (tree-maps, heat-maps, network graphs, text-analytics, bar-charts, etc.) can offer an interactive, holistic representation to aid in analyzing the structure/content of a computing curriculum, assessing proposed changes, and benchmarking an individual offering in a given academic institution against published guidelines. As a sample data set we explore the results of a research project based upon U.S. job postings during the period of 1999 through 2012. Our purpose is to demonstrate the potential of visualization techniques to reveal and analyze the evolving demand for different computing skill areas and desired depth of competency.

Keywords: computing curricula, visualization, curriculum representation, curriculum analysis / review

1. INTRODUCTION

There is a long tradition among computing societies defining and promulgating their disciplinary

identities through the publication of curricular guidelines that outline the knowledge, skills, competencies, and standards of professional conduct.

Each of these curricular efforts has involved the painstaking effort of identifying the elemental structure of each of these curricular edifices: the proposed learning outcomes, suggested pedagogically directed modularization and sequencing of content, and a vision of proficiency that meets the career aspirations of a slice of professional practice. In addition, each guideline also offers a range of suggested alternative institutional approaches for locally adapting to the geographic, cultural, and workforce context of the employment market and their graduates.

The depth and range of curricular aspects and issues make its consumption, interpretation, comparison, and analysis a daunting intellectual task. In this paper, we explore the opportunities and potential benefits of applying visual analytics tools and technology to the tasks of design, evaluation, comparison, analysis and discovery as well as the evolution of models of computing curriculum.

We begin by reviewing the basic elements and structure used to portray curricula in the stream of guidelines published over the years in the disciplines of computing. In the sections that follow we have chosen a demonstration dataset originally compiled by a research team exploring the evolution of industry emphasis in professional competencies as they relate to computing education knowledge areas. Although this dataset is not a curriculum in itself, it possesses many of the dimensions that require comparison and analysis and provides a simplified domain of data to facilitate the visualization examples included. We will apply various tools and systems of property visualization using the example dataset to review the cognitive implications and range of options. Finally, we will outline an approach coordinating an array of visualization tools with a unified user interface, a “dashboard,” to gather task inventory information and usability assessments to advance to the next stage of development.

2. FIFTY YEARS OF CURRICULAR EFFORTS

A National Science Foundation (NSF) grant funded a committee *on Computer Education for Management* as early as the 1964. That work, “Curriculum Development in Management Information Systems Education in Colleges and Universities,” appeared in November of 1965 (ACM, 1965). The first curriculum for CS, “Recommendation for Academic Programs in Computer Science,” (ACM, 1968) was followed shortly by

“Curriculum Recommendations for Graduate Professional Programs in Information Systems,” (Ashenurst, 1972), and then, undergraduate IS, (Couger, 1973). The Association for Computing Machinery (ACM) was joined in the 1980’s by the Institute for Electrical and Electronic Engineers (IEEE-CS) in a joint committee to develop computing curriculum guidelines for undergraduate degrees in computer science and computer engineering (Tucker, 1991). The Data Processing Management Association (DPMA) [later renamed the Association of Information Technology Professionals (AITP)] published guidelines focused on undergraduate Information Systems (IS) education in 1981 and 1986 (DPMA, 1981, 1986). ACM, IEEE and AITP were joined by the Association of Information Systems (AIS) to publish CC2005 that overviewed computer science, computer engineering, information systems, software engineering and information technology (Shackelford, McGettrick, Sloan, Topi, Davies, Kamali, Cross, Impagliazzo, LeBlanc, & Lunt, 2006). Subsequently there have emerged IS2010 and IT2017 is in final draft stage (Topi, Valacich, Wright, Kaiser, Nunamaker, Sipior, & Vreede, 2010; Savin, Alrumaih, Impagliazzo, Lunt, Zhang, Byers, Newhouse, Paterson, Peltsverger, Tang, van der Veer, & Viola, 2017). Table 1 summarizes the major curriculum projects and publications since 1965.

Because of the interdependent nature and complexity of the computing curricula (e.g. Bodies of Knowledge, Areas of Knowledge, taxonomy of topics, levels of coverage, learning outcomes, professional competencies) or as a matter of fact any curriculum, a comprehensive perspective and understanding is a challenge.

3. WHY INTERACTIVE DATA VISUALIZATION?

Interactive data visualization is an instrument for reasoning about data. Visual analytics tools can be used to provide an interactive, holistic view to analyze the structure of a computing curriculum, visualize and assess prospective changes, analyze an area of knowledge as well as benchmark an individual curriculum in a given academic institution against some published guideline or standards.

At its root data visualization is simply communication! While we’re used to thinking of communication in terms of speech or print, these media actually reside in a very restricted channel of idea conveyance. The old adage, “a picture is worth a thousand words” is indeed grounded in truth. A

good data visualization leaves a lasting mental model of a fact, trend or process (Romer, 2015). Shneiderman’s Mantras (1996) are well established guidelines for presentation and interaction of data on visual displays. Edward Tufte (1990, 2006) espoused graphical design principles that exploit the cognitive facets of perception: *show* the data, induce the viewer to think about the substance of the data, utilize the *thinking eye* by letting the eye compare the data visually, let the *seeing* brain discern insights by presenting varying levels of details through rolling-up and drilling-down into the data.

| Year | Contributers | | | | Publication |
|------|---------------|-----|------|-----|---|
| | DPMA/ AITP | ACM | IEEE | AIS | |
| 1968 | | X | | | Computer Science (ACM, 1968) |
| 1972 | | X | | | Computer Education for Management (Ashenurst, 1972). |
| 1973 | | X | | | Undergraduate Programs in Information Systems (Couger, 1973) |
| 1981 | X | | | | DPMA Model Curriculum (DPMA, 1981) |
| 1982 | X | | | | IS Curriculum Recommendations for the 80s (Nunamaker, 1982). |
| 1986 | X | | | | DPMA Model Curriculum (DPMA, 1986) |
| 1990 | X | | | | IS'90 (Longenecker, 1991) |
| 1991 | | X | X | | Computing Curricula '91 (Tucker, 1991) |
| 1995 | X | X | | X | IS'95 (Couger, 1995) |
| 1997 | X | X | | X | IS'95 (Couger, 1997) |
| 2002 | X | X | | X | IS 2002 (Gorgone, 2003) |
| 2004 | X | | | | Post Secondary Education Programs in Data Resource Management (Henderson, 2004) |
| 2005 | | X | X | X | CC2005 (Shackelford, 2006) |
| 2006 | | X | | X | MSIS 2006 (Gorgone, 2016) |
| 2008 | | X | X | | CS2008 (Cassel, 2008) |
| 2008 | | X | X | | IT 2008 (IEEE/ACM, 2008) |
| 2010 | | X | | X | IS2010 Topi, 2010 |
| 2013 | | X | X | | CS2013 (ACM, 2013) |
| 2016 | | X | | X | MSIS 2016 (ACM, 2016) |
| 2017 | | X | X | | IT2017 (Final Draft) (ACM/IEEE, 2017) |

Table 1. Published Curriculum Reports

The human brain is capable of receiving stimuli from virtually every organ in the body simultaneously. Much of that sensory data never reaches the domain of conscious attention. But, none-the-less, all that data is available for consideration and forms a “wave” of presence for the observer. One of the advantages of interactively visualized information is the circumvention of the routine channels of uttered speech and written prose in language, style, inflections, and emphasis. In

fact, visual communication allows the transmission of concepts and ideas that have no equivalent in spoken or written language.

When ideas are communicated through speech or prose, the author must transliterate their mental model into a verbal or written model of language. Upon receipt the hearer or reader must transliterate (again) the spoken or written words through their language model to finally reach a mental model of the concept of their own. In either case, the concept is transmitted in a serial, sequential expression of data elements where the concept is reassembled in this same serial manner. If the message is sufficiently long, the mental skill and effort required to maintain the series of elements needed to correlate the early elements with those arriving some few words (or many words) later is significant and the sophistication of the analysis of the whole message may be compromised by the commensurate effort required.

Visualized concepts have the benefit of being represented in a significantly more parallel, all-at-once presentation mode. Although there are definite elements of syntax and semantics in graphical representation, the observer’s experience of the visualization is unencumbered by the serial nature of speech or prose. Indeed, the observer is freer to attempt numerous “explanations” of the observation; and in the case of visualized data, engage explanations outside the experience of the visual artifact’s creators(s). The observer has the freedom to attempt any number of interpretations of the visualized data: some correlated to the context of experience or setting, but others triggered by association with tacit knowledge beyond verbal explanation (Polanyi, 1969). Tacit knowledge is knowing that accumulates through lived experience rather than formal education. It manifests as reflexive rather than calculated. For this simple reason, data visualization is a potent tool for exploring the structure and patterns of phenomena where “hidden” inferences or recurrent syndromes disclose here-to-fore overlooked realities.

Modes of Visual Intimation – Suggestive Perspective

The characteristics of the data and the questions asked of it intimate the type of visualization. Demographic data may have a latitude and a longitude, however that should not be the only dimension, that is where various modes of visual intimation, visual metaphors as it were, provide invitation to particular perspectives. These modes can provide an interactive framework that facili-

tates the visual analysis for the purpose of understanding, evaluating, reorganizing and comparing. Some examples are:

- Textual: interactive Word Clouds allow us to take a first look at the open-ended answers and comments of an article or a survey before reading it.
- Network graphs allow us to explore the relationships and associations between linked elements.
- Bar charts and heat maps allows us to compare ordinal and nominal data beyond the listing of facts.
- Multi-Faceted (Year, Discipline, etc.) expose juxtaposition and distance of graphical elements that can represent scalar and/or relative relationships.
- Overlap of topic coverage such as Venn diagrams connote inter-connectedness and whole-part.
- Dependency among topics can represent precedence, necessity or sufficiency among aspects.
- Hierarchical data can be visualized using tree-maps.
- Geographic maps allow us to analyze data that has a geolocation element (address).

If some or all of these visualizations can be assembled as an interactive framework, they can facilitate the visual analysis for the purpose of understanding, evaluating, benchmarking and comparing. They may also improve the analysis of the evolutionary progression of the learner (entry, exit, depth) and the traversal of bodies of knowledge.

4. ONTOLOGIZING CURRICULUM ELEMENTS

Aside from a purely artistic rendering of information primarily intended to evoke emotion rather than perspective or understanding, analytical visualization focuses on casting data in a dimensional space that exposes conceptual characteristics. The following is a possible conceptual structure for such an ontology:

- 1) Visualizing the landscape of computing education
 - a) Visualizing Curriculum
 - i) Characteristics
 - (1) Content relationships
 - (a) Interdependency
 - (i) Dependence
 - (ii) Precedence
 - (iii) Priority
 - (b) Interrelationship

- (i) Underlying theory
 - (ii) Coincident application
 - (iii) Coincident tradeoffs
 - (2) Relative desirable segment competency
 - ii) Individually
 - iii) Comparatively (between/among curricula)
 - (1) Distinguishing differences
 - (2) Coincident similarities
 - (3) Relative desirable competency
 - (4) Segment proportions
- b) Representing curricular characteristics with visual metaphors

An Ontology of Curriculum Elements

In the ACM CS-2013, Computer Science Curricula 2013, the elemental structure and relationships of curricular content taxonomy follow the KA-KULO model (Knowledge area, Knowledge unit, Learning objective). Figure 1 is a UML diagram depiction of the taxonomy of terms depicting the curricular elements, associations, and dependencies as they are described in the ACM CS-2013 report.

ACM CS-2013 describes the curricular elements thusly:

- **Body of Knowledge:** The outline of Topics that should appear in an undergraduate computing Curricula (ACM, 2013, p. 13); a specification of the content to be covered
- **Knowledge Areas:** The Body of Knowledge organized into a set of Knowledge Areas (KAs), corresponding to topical areas of study in Computing. Examples include Information Management, Programming Languages, Social Issues and Professional Practices, Information Assurance and Security, etc. (ACM, 2013, p. 14)
- **Level of Coverage:** On a scale, a judgment to the depth of coverage of the topic at the undergraduate level intended to achieve a level of mastery in a particular learning outcome
- **Hours:** Hours spent on the topic in the classroom
- **Curriculum:** is the implementation of the Body of Knowledge Specification
- **Core Tier-1 Topic:** Topics with widespread consensus for inclusion in every program topic should be a required part of every Computer Science curriculum

- **Core Tier-2 Topic:** Topics that are generally essential in an undergraduate degree
- **Elective Topic:** Covers Core Topics to deepen the understanding in multiple areas
- **Course:** Incorporates topics from multiple knowledge areas

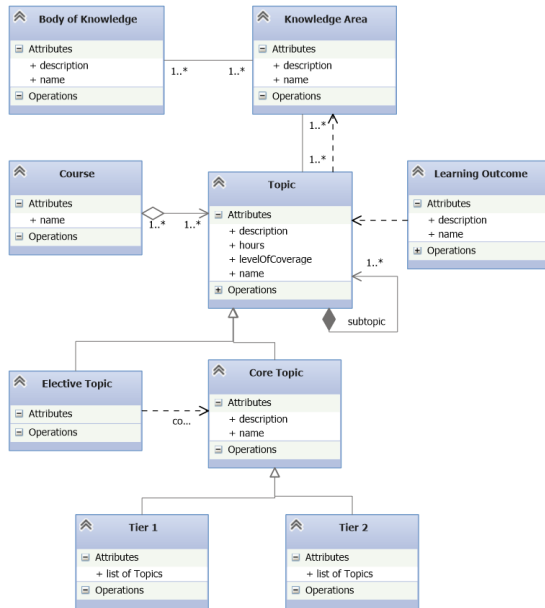


Figure 1 - ACM CS-2013 Curriculum Structure

The Report details 18 Bodies of Knowledge (Algorithms and Complexity, ..., Information Management, ..., Social Issues and Professional Practice), Information Management details 12 knowledge areas (Information Management Concepts, ..., Data Modeling, ..., Multimedia Systems), Data Modeling details 4 Core-Tier1 Topics with 6 learning outcomes and 4 Core-Tier2 Topics with 7 learning outcomes. Appendix-A is an example realization of the Bodies of Knowledge, Knowledge Areas, topics learning and outcomes.

MSIS 2016 and IT 2017 have incorporated an alternative expanding the representation of assessing the productivity of curricula in the form of competencies (CCSL, 2013). (See Figure 2.)

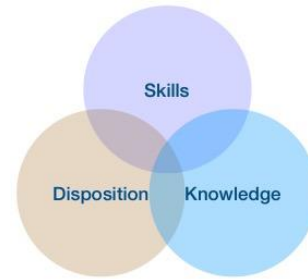


Figure 2 - Competence = Knowledge + Skills + Disposition

Competencies are composed of three facets:

- **Knowledge** - mastery of rigorous content knowledge across multiple disciplines and the facile application or transfer of what has been learned,
- **Skills** - the strategies that students need to engage in higher-order thinking, meaningful interaction with the world around them, and future planning; and
- **Dispositions** - mindsets (sometimes referred to as behaviors, capacities, or habits of mind) that are closely associated with success in college and career.

The addition of these taxonomic aspects of learning to the KA-KU-LO model represents a significant increase in complexity and challenge to analyzing and comprehending a curriculum model. Competencies introduces another dimension in a system of knowing, learning, and practice.

5. VISUALIZING THE JOB POSTING DATA

To demonstrate the power of visual analytics as it applies to the computing curriculum, we borrow the survey data from Longenecker, Feinstein and Clark's 2012 research they abstracted as follows:

"This article presents the results of research to explore the nature of changes in skills over a fifty-year period spanning the life of Information Systems model curricula. Work begun in 1999 was expanded both backwards in time, as well as forwards to 2012 to define skills relevant to Information Systems curricula. The work in 1999 was based on job ads from 17 major national newspapers. The ~3000 ads enabled generation of 37 skills and defined major areas of skills: software development, web development, database, operating systems and telecommunications, strategic organizational development, inter-

personal and team skills, and project management. During the development of this research a ninth skill area was added: information and security assurance. The original 37 skills had been expanded to 69 skills, and within this effort, 69 additional skills were added. Analysis of the skills as of today suggested elimination of retired (24) and too new (13) skills. Of the remaining skills, a set (35) of skills was common to all curricula, a large set of current skills (64) was abandoned by IS 2010 which added new skills (2). Deletion of programming as a requirement of IS 2010 accounts for a significant proportion of deletions.” (Longenecker, 2012)

In their paper, the authors used a three-level skill hierarchy: *Parent-Skill* → *Child-Skill* → *Grand-Child-Skill* → *Skill-Depth*. They gathered *Skill-Depth* data across the 1973, 1981, 1986, 1990, 2002, 2010 Model Curricula as depicted in Table 2. Appendix 2 of their paper contains the full details of their survey. They also used word (phrases) to describe the keywords of a skill that we will also use to demonstrate text analysis through visualization.

| Longenecker’s Mapping | | | | |
|-----------------------|----------------------|---------------------------|-------------|------|
| Parent Skill | Child Skill | Grand Child | Skill Depth | Year |
| IT | Software Development | Low Level Data Structures | 2 | Y73 |
| IT | Software Development | Low Level Data Structures | 3 | Y81 |
| IT | Software Development | Low Level Data Structures | 1 | Y86 |
| IT | Software Development | Low Level Data Structures | 3 | Y90 |
| IT | Software Development | Low Level Data Structures | 2 | Y02 |
| IT | Software Development | Low Level Data Structures | 2 | Y10 |
| ... | ... | ... | ... | ... |

Table 2. Longenecker Skill Structure

We mapped their hierarchy to the *Body-of-Knowledge* → *Knowledge-Area* → *Topic* → *Level-of-Coverage* consistent with the CS-2013 Curriculum guidelines to create a dataset for the visualization demonstrations that follow (See examples in Table 3):

| ACM CS-2013 | | | | |
|-------------------|----------------------|---------------------------|-------------------|------|
| Body of Knowledge | Knowledge Area | topic | Level of Coverage | Year |
| IT | Software Development | Low Level Data Structures | 2 | Y73 |
| IT | Software Development | Low Level Data Structures | 3 | Y81 |
| IT | Software Development | Low Level Data Structures | 1 | Y86 |
| IT | Software Development | Low Level Data Structures | 3 | Y90 |
| IT | Software Development | Low Level Data Structures | 2 | Y02 |
| IT | Software Development | Low Level Data Structures | 2 | Y10 |

Table 3. ACM CS-2013 Curriculum Structure

Example Visual Analytics

For this paper, we use *Tableau™*, R and *ggplot*, word cloud, text mining(tm) and n-gram tokenizer packages. However, going forward in this

project, we envision prototyping curriculum analysis tools using D3JS, Javascript and SVG in a web-browser environment. These tools provide the ability to animate graphic elements to either drill-down to greater detail, recall descriptive text, or perform a reorganization of graphic elements along a property dimension.

To analyze the depth of coverage across multiple years, we can produce heat maps where we can filter by combinations of year, Body of Knowledge and level of coverage. The visualization tools we demonstrate produce presentations suitable for large screen video. We encourage the reader to utilize the “zoom” feature of their pdf-reader to enlarge and inspect the detail of these visualizations.

The Big-Picture

Figure 3 in Appendix B is an interactive heat map of the topics vs. the CC-Year where the depth of color reflects the depth of coverage for that CC-Year with Deep green (4) is the highest and white is the lowest). We can follow the evolution of the topics for a 38-years stretch: when a topic was introduced, how much emphasis, and when a topic was retired. We can see how IT → “*Software Development*” was obliterated in the 2010 Year. In Figure 4 can also see the inconsistencies – how “*Low Level Data Structures*” and “*Algorithmic Design, Data, Object and Files Structure*” emphasis remained while the rest of the Software Development disappeared in 2010. We can also see that “*Modeling and Design, construction, Schema tools*” emphasis expanded in later years.

Taking another view

Figure 4 in Appendix B is a stacked bar chart of the same data in Figure 3 in Appendix B, the purpose of which is to produce multiple representations of the same data to accommodate different visual styles. The length and width are proportional to the level of coverage of a topic across the CC-Years.

Figure 5 in Appendix B is a tree-map based hierarchical visualization of the data. Tree-maps is a framework for visualizing multi-layers’ hierarchical data. It was popularized by Shneiderman and Wattenberg (1992, 2001). It allows us to visualize the hierarchical content of the data as a rectangle within rectangle within rectangle where the size of a rectangle is the metric that needs to be measured and color can be any dimension we want to group by. Figure 5 shows the Year → Knowledge Area → Body of Knowledge → Level of coverage hierarchy where the size of a rectangular area and the depth of color are both proportional to the Level of Coverage.

Figure 6 in Appendix B is a filtered treemap comparing the topics covered in only Year-73 and Year-2010. The size of the rectangle reflects the level of coverage of a topic for that year and the color represents the topic itself. For example, the coverage of Database Modeling and Design changed from level 2 to 4 between 1973 and 2010 indicating increased emphasis; Data Warehousing was not covered at all in 1973 but, appears at level 2 in 2010.

Putting it all together

Figures 3-6 in Appendix B demonstrate the power of visual analytics to expose and highlight different aspects of the curriculum, however when all of these aspects are put together into a dashboard (see Appendix B, Figure 7), the thinking eye and the seeing brain are empowered to work together and in-tandem. Like gazing at Da Vinci's Mona Lisa, the data visualized admits to pattern recognition and insights that engage the intuitive depths of visual recognition and interpretation tapping into the individual analyst's tacit understanding and expectation of curriculum. Each intimated pattern suggests questions to be explored by reorienting the visualization parameters to expose details of curriculum often obscured by the mere volume and complex relationships of the raw data.

| Knowledge Area | CE | | CS | | IS | | IT | | SE | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | min | max | min | max | min | max | min | max | min | max |
| Programming Fundamentals | 4 | 4 | 4 | 5 | 2 | 4 | 2 | 4 | 5 | 5 |
| Integrative Programming | 0 | 2 | 1 | 3 | 2 | 4 | 3 | 5 | 1 | 3 |
| Algorithms and Complexity | 2 | 4 | 4 | 5 | 1 | 2 | 1 | 2 | 3 | 4 |
| Computer Architecture and Organization | 5 | 5 | 2 | 4 | 1 | 2 | 1 | 2 | 2 | 4 |
| Operating Systems Principles & Design | 2 | 5 | 3 | 5 | 1 | 1 | 1 | 2 | 3 | 4 |
| Operating Systems Configuration & Use | 2 | 3 | 2 | 4 | 2 | 3 | 3 | 5 | 2 | 4 |
| Net Centric Principles and Design | 1 | 3 | 2 | 4 | 1 | 3 | 3 | 4 | 2 | 4 |
| Net Centric Use and configuration | 1 | 2 | 2 | 3 | 2 | 4 | 4 | 5 | 2 | 3 |
| Platform technologies | 0 | 1 | 0 | 2 | 1 | 3 | 2 | 4 | 0 | 3 |
| Theory of Programming Languages | 1 | 2 | 3 | 5 | 0 | 1 | 0 | 1 | 2 | 4 |
| Human-Computer Interaction | 2 | 5 | 2 | 4 | 2 | 5 | 4 | 5 | 3 | 5 |
| Graphics and Visualization | 1 | 3 | 1 | 5 | 1 | 1 | 0 | 1 | 1 | 3 |
| Intelligent Systems (AI) | 1 | 3 | 2 | 5 | 1 | 1 | 0 | 0 | 0 | 0 |
| Information Management (DB) Theory | 1 | 3 | 2 | 5 | 1 | 3 | 1 | 1 | 2 | 5 |
| Information Management (DB) Practice | 1 | 2 | 1 | 4 | 4 | 5 | 3 | 4 | 1 | 4 |
| Scientific computing (Numerical mthds) | 0 | 2 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| Legal / Professional / Ethics / Society | 2 | 5 | 2 | 4 | 2 | 5 | 2 | 4 | 2 | 5 |
| Information Systems Development | 0 | 2 | 0 | 2 | 5 | 5 | 1 | 3 | 2 | 4 |
| Analysis of Business Requirements | 0 | 1 | 0 | 1 | 5 | 5 | 1 | 2 | 1 | 3 |
| E-business | 0 | 0 | 0 | 0 | 4 | 5 | 1 | 2 | 0 | 3 |
| Analysis of Technical Requirements | 2 | 5 | 2 | 4 | 2 | 4 | 3 | 5 | 3 | 5 |
| Engineering Foundations for SW | 1 | 2 | 1 | 2 | 1 | 1 | 0 | 0 | 2 | 5 |
| Engineering Economics for SW | 1 | 3 | 0 | 1 | 1 | 2 | 0 | 1 | 2 | 3 |
| Software Modeling and Analysis | 1 | 3 | 2 | 3 | 3 | 3 | 1 | 3 | 4 | 5 |
| Software Design | 2 | 4 | 3 | 5 | 1 | 3 | 1 | 2 | 5 | 5 |
| Software Verification and Validation | 1 | 3 | 1 | 2 | 1 | 2 | 1 | 2 | 4 | 5 |
| Software Evolution (maintenance) | 1 | 3 | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 4 |
| Software Process | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 2 | 5 |
| Software Quality | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 4 |
| Comp Systems Engineering | 5 | 5 | 1 | 2 | 0 | 0 | 0 | 0 | 2 | 3 |
| Digital logic | 5 | 5 | 2 | 3 | 1 | 1 | 1 | 1 | 0 | 3 |
| Embedded Systems | 2 | 5 | 0 | 3 | 0 | 0 | 0 | 1 | 0 | 4 |
| Distributed Systems | 3 | 5 | 1 | 3 | 2 | 4 | 1 | 3 | 2 | 4 |
| Security: issues and principles | 2 | 3 | 1 | 4 | 2 | 3 | 1 | 3 | 1 | 3 |
| Security: implementation and mgr | 1 | 2 | 1 | 3 | 1 | 3 | 3 | 5 | 1 | 3 |
| Systems administration | 1 | 2 | 1 | 1 | 1 | 3 | 3 | 5 | 1 | 2 |
| Management of Info Systems Org. | 0 | 0 | 0 | 3 | 5 | 0 | 0 | 0 | 0 | 0 |
| Systems integration | 1 | 4 | 1 | 2 | 1 | 4 | 4 | 5 | 1 | 4 |
| Digital media development | 0 | 2 | 0 | 1 | 1 | 2 | 3 | 5 | 0 | 1 |
| Technical support | 0 | 1 | 0 | 1 | 1 | 3 | 5 | 5 | 0 | 1 |

Table 4. Sample Comparative weight of computing topics across the five kinds of degree programs in CC2005

Another View

In Figure 8 in Appendix B, we also demonstrate how to visualize information as ranges, the "Comparative weight of computing topics (200) across the five kinds of degree programs" which is Table 3.1 of the CC-2005 Curriculum report (reproduced sample in Table 4). We use R and ggplot2 (Grammar of Graphics Package). The table illustrates the min and max levels of coverage called for by the curriculum guidelines across discipline.

Visualizing the Field of Learning Objectives as a Wordcloud

We should also be able to analyze the Learning Objectives, elements of an ontological view of the curriculum. To demonstrate, we produced a 3-4-Grams wordcloud visualization of the 138 unique word phrases tokenized with R-tm from text extracted from the skill-set text published in the Longenecker, Feinstein and Clark's 2012 survey data. (See Appendix B, Figure 9.) The font size of a token (3-4-grams) is proportionate to its frequency of appearance as a token. In an interactive visualization environment, we will also be able to navigate from the phrase to its *in-situ* source text location(s).

Usage Scenarios: Analysis Questions

1. Tree-Maps
 - 1.1. Can we view Bodies of Knowledge, Knowledge Areas, topics, and their relative levels of coverage?
 - 1.2. Can we filter on Bodies of Knowledge, knowledge areas to identify critical relationships between knowledge units and competencies, etc?
 - 1.3. Can we compare the level of coverage of a topic across multiple knowledge areas?
 - 1.4. Can we compare topics across knowledge areas (lends itself to heat-maps)?
2. Word Clouds
 - 2.1. Can we identify recurrent computing topics across programs or courses?
 - 2.2. Can we identify objectives that recur across programs or courses?
 - 2.3. Can we identify "missing" topics or objectives across programs or courses?
 - 2.4. Can we locate objectives or essential concepts as recurring themes in text?

All of the computing disciplines have a shared identity but as noted in CC2005,

"Each computing discipline must articulate its own identity, recognize the identities of the other disciplines, and contribute to the shared identity of computing." (Shackelford et al, 2005, p. 8)

6. DISCUSSION

Because of the topic overlap within and across computing curricula, it is a challenge to analyze and compare the breadth and depth of topic coverage in and amongst our various curricula. That analysis and comparison is critical to curriculum design, assessment, and accreditation as well as exploiting and optimizing resource synergies across collocated computing programs. Curricular data visualization offers tools to support the understanding and analysis of these complex intellectual artifacts.

In this paper, we argued for the use of interactive visual analytics and off-the-shelf visual analytics tools to understand, compare and gain insights into curriculum design and curricular models. However, to make the process repeatable and for the tools to be generic, there has to be an agreed-upon unified ontological schema that defines the structures, their relationships and the currencies to be used in the process. It is our goal in this visualization analysis exploration to prototype and evaluate the options for both the ontology(s) and visualizations that will best support the advancement of computing curricula.

7. REFERENCES

- ACM (1968), "Recommendations for Academic Programs in Computer Science," *Communications of the ACM*, Vol 10, No 3, March 1968.
- ACM (2013), Computer Science Curricula 2013: Curricula Guidelines for Undergraduate Degree Programs in Computer Science, ACM, IEEE, December 20, 2013: Retrieved 8 August 2017: www.acm.org/education/CS2013-final-report.pdf
- ACM (2016), MSIS 2016: Global Competency Model for Graduate Degree Programs in Information Systems, ACM, AIS: 23 May 2017: retrieved 8 August 2017: www.acm.org/binaries/content/assets/education/msis2016.pdf
- ACM (2017), Curriculum Guidelines for Undergraduate Degree Programs in Information Technology (Final Draft), ACM, IEEE: 27 July 2017: Retrieved 8 August 2017: www.acm.org/binaries/content/assets/education/it2017.pdf
- Ashenhurst, R. L. (Ed.) (1972). A Report of the ACM Curriculum Committee on Computer Education for Management. *Association for Computing Machinery, Inc.*, 1972.
- Cassel, L., Clements, A., Davies, G., Gusdial, M., McCauley, R., McGettrick, A., Sloan, B., Snyder, L., Tymann, P. & Weide, B. W. (2008). Computer Science Curriculum 2008: An Interim Revision of CS2001, ACM, IEEE, December 2008: Retrieved 8 August 2017: www.acm.org/education/curricula/ComputerScience2008.pdf
- CCSL, Council of Chief State School Officers. (2013). Knowledge, Skills, and Dispositions: The Innovation Lab Network State Framework for College, Career, and Citizenship Readiness, and Implications for State Policy.
- Couger, J. (Ed.) (1973). Curriculum Recommendations for Undergraduate Programs in Information Systems, *Communications of the ACM*, Volume 16, Number 12, December 1973, pp. 727-749.
- Couger, J. D., Davis, G. B., Dologite, D. G., Feinstein, D. L., Gorgone, J. T., Jenkins, M., Kasper, G. M. Little, J. C., Longenecker, H. E. Jr., and Valachic, J. S. (1995). IS'95: Guideline for Undergraduate IS Curriculum, *MIS Quarterly* (19:3), 1995, pp. 341-360.
- Couger, J. D., Davis, G.B., Feinstein, D.L., Gorgone, J.T. and Longenecker, H.E. (1997). IS'97: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, *Data Base*, Vol. 26 No. 1, pp. 1-94.
- DPMA (1981). *DPMA Model Curriculum*, 1981. Park Ridge, Illinois: Data Processing Management Association.
- DPMA (1986). *DPMA Model Curriculum*, 1986. Park Ridge, Illinois: Data Processing Management Association.
- Gorgone, J.T., Davis, G.B. Valacich, J., Topi, H., Feinstein, D.L. and Longenecker, H.E. (2003). IS 2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. Retrieved 20 September 2017, www.acm.org/education/education/curric_vol_s/is2002.pdf

- Gorgone, J.T., Gray, P., Stohr, E.A., Valacich, J. & Wigand, R. T. (2006). MSIS 2006: Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems, *Communications of the AIS*, Vol 17, Article 1, January 2006, retrieved August 8, 2017, www.acm.org/education/education/curric_vols/MSIS%202006.pdf
- Henderson, D., B. Champlin, D. Coleman, P. Cupoli, J. Hoffer, L. Howarth, K. Sivier, A., M. Smith, and E. Smith (2006). DAMA Foundation Model Curriculum Framework for Post Secondary Education Programs in Data Resource Management. *Information Systems Education Journal*, 4 (47). isedj.org/4/47/. ISSN: 1545-679X. 32 p.
- IEEE/ACM (2008). IEEE/ACM Joint Task Force on Computing Curricula. Information Technology 2008, Curriculum Guidelines for Undergraduate Degree Programs in Information Technology, ACM and IEEE-Computer Society, November 2008. Retrieved 21 August 2017, www.acm.org/education/education/curricula-recommendations
- Longenecker, H.E., Feinstein, D. L. (Eds.) (1991c). *IS'90: The DPMA Model Curriculum for Information Systems for 4 Year Undergraduates*. Park Ridge, Illinois: Data Processing Management Association.
- Nunamaker, J.F., Couger, J.D. and Davis, G.B. (1982). "Information Systems Curriculum Recommendations for the 80s: Undergraduate and Graduate Programs," *Communications of the ACM*, Volume 25, Number 11, November 1982, pp. 781-805.
- Polanyi, M. (1969). *Knowing and Being: Essays by Michael Polanyi*, University of Chicago Press, Chicago, IL.
- Shackelford, R., McGettrick, A., Sloan, R., Topi, H., Davies, G., Kamali, R., Cross, J., Impagliazzo, J., LeBlanc, R., & Lunt, B. (2006). Computing Curricula 2005: The overview report. *ACM SIGCSE Bulletin*, 38(1), 456-457.
- Romer, B. (2015). A picture is worth more than a thousand words. Thomsonreuters.com: Retrieved 10 August 2017, blogs.thomsonreuters.com/answerson/data-visualization-worth-thousand-words/
- Savin, M., Alrumaih, H., Impagliazzo, J., Lunt, B., Zhang, M., Byers, B., Newhouse, W., Paterson, B, Peltsverger, S., Tang, C., van der Veer, G., & Viola, B., (2017) Information Technology Curriculum 2017 (Interim report), Association of Computing Machinery (ACM) & IEEE Computing Society (IEEE-CS), retrieved 9 September 2017, www.acm.org/binaries/content/assets/education/it2017.pdf
- Shneiderman, B., & Wattenberg, M. (2001) Ordered Treemaps Layouts. *Proceedings of the IEEE Symposium on Information Visualization*, ISSN: 1522-404X, 73-78.
- Shneiderman, B. (1996) The Eyes Have It: A Task by Data Type Taxonomy of Information Visualization. *Proceedings of IEEE Symposium on Visual Languages*. 336-343.
- Shneiderman, B. (1992). Tree Visualization with Treemaps: a 2-d space-filling approach. *ACM Transaction on Graphics*, 11(1), 92-99.
- Topi, H., Valacich, J., Wright, R.T., Kaiser, K.M., Nunamaker, J.F., Sipior, J.C., and Vreede, G.J. (2010). IS 2010 Curriculum Guidelines for Undergraduate Degree Programs in Information Systems, Association for Computing Machinery (ACM), Association for Information Systems (AIS)", retrieved August 7, 2017, www.acm.org/education/curricula/IS%202010%20ACM%20final.pdf
- Tucker, A. B. (Ed.) (1991) *Computing Curricula '91*. Association for Computing Machinery and the Computer Society of the Institute of Electrical and Electronics Engineers, *Communications of the ACM*, 34(6), 68-84.
- Tufte E. (1983). *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, CT.
- Tufte E. (2006). *Beautiful Evidence*, Graphics Press, Cheshire, CT.

Appendix A -

An ACM CS-2013 realization of the Bodies of Knowledge, Knowledge Areas and Topics

1. Algorithms and Complexity (AL)
2. Architecture and Organization (AR)
3. Computational Science (CN)
4. Discrete Structures (DS)
5. Graphics and Visualization (GV)
6. Human-Computer Interaction (HCI)
7. Information Assurance and Security (IAS)
8. Information Management (IM)
9. Intelligent Systems (IS)
10. Networking and Communication (NC)
11. Operating Systems (OS)
12. Platform-Based Development (PBD)
13. Parallel and Distributed Computing (PD)
14. Programming Languages (PL)
15. Software Development Fundamentals (SDF)
16. Software Engineering (SE)
17. Systems Fundamentals (SF)
18. Social Issues and Professional Practice (SP)

• **Body of Knowledge:** Information Management (IM)

- Knowledge Area: Information Management Concepts [*1 Core-Tier1 hour; 2 Core-Tier2 hours*]

• **Topics:**

- [Core-Tier1]
 - Information systems as socio-technical systems
 - Basic information storage and retrieval (IS&R) concepts
 - Information capture and representation
 - Supporting human needs: searching, retrieving, linking, browsing, navigating
- [Core-Tier2]
 - Information management applications
 - Declarative and navigational queries, use of links
 - Analysis and indexing
 - Quality issues: reliability, scalability, efficiency, and effectiveness

• **Learning Outcomes:**

- [Core-Tier1]
 1. Describe how humans gain access to information and data to support their needs. [Familiarity]
 2. Describe the advantages and disadvantages of central organizational control over data. [Assessment]
 3. Identify the careers/roles associated with information management (e.g., database administrator, data modeler, application developer, end-user). [Familiarity]
 4. Compare and contrast information with data and knowledge. [Assessment]
 5. Demonstrate uses of explicitly stored metadata/schema associated with data. [Usage]
 6. Identify issues of data persistence for an organization. [Familiarity]
- [Core-Tier2]
 1. Critique an information application with regard to satisfying user information needs. [Assessment]
 2. Explain uses of declarative queries. [Familiarity]
 3. Give a declarative version for a navigational query. [Familiarity]
 4. Describe several technical solutions to the problems related to information privacy, integrity, security, and preservation. [Familiarity]
 5. Explain measures of efficiency (throughput, response time) and effectiveness (recall, precision). [Familiarity]
 6. Describe approaches to scale up information systems. [Familiarity]
 7. Identify vulnerabilities and failure scenarios in common forms of information systems. [Usage]
- Knowledge Area: Database Systems
.....
- Knowledge Area: Data Modeling [*4 Core-Tier2 hours*]

- **Topics:**
 - Data modeling
 - Conceptual models (e.g., entity-relationship, UML diagrams)
 - Spreadsheet models
 - Relational data models
 - Object-oriented models (cross-reference PL/Object-Oriented Programming)
 - Semi-structured data model (expressed using DTD or XML Schema, for example)
- **Learning Outcomes:**
 1. Compare and contrast appropriate data models, including internal structures, for different types of data. [Assessment]
 2. Describe concepts in modeling notation (e.g., Entity-Relation Diagrams or UML) and how they would be used. [Familiarity]
 3. Define the fundamental terminology used in the relational data model. [Familiarity]
 4. Describe the basic principles of the relational data model. [Familiarity]
 5. Apply the modeling concepts and notation of the relational data model. [Usage]
 6. Describe the main concepts of the OO model such as object identity, type constructors, encapsulation, inheritance, polymorphism, and versioning. [Familiarity]
 7. Describe the differences between relational and semi-structured data models. [Assessment]
 8. Give a semi-structured equivalent (e.g., in DTD or XML Schema) for a given relational schema. [Usage]
- **Knowledge Area: Indexing [Elective]**
 - **Topics:**
 - The impact of indices on query performance
 - The basic structure of an index
 - Keeping a buffer of data in memory
 - Creating indexes with SQL
 - Indexing text
 - Indexing the web (e.g., web crawling)
 - **Learning Outcomes:**
 1. Generate an index file for a collection of resources. [Usage]
 2. Explain the role of an inverted index in locating a document in a collection. [Familiarity]
 3. Explain how stemming and stop words affect indexing. [Familiarity]
 4. Identify appropriate indices for given relational schema and query set. [Usage]
 5. Estimate time to retrieve information, when indices are used compared to when they are not used. [Usage]
 6. Describe key challenges in web crawling, e.g., detecting duplicate documents, determining the crawling frontier. [Familiarity]
- **Knowledge Area: Relational Databases**
- **Knowledge Area: Query Languages**
- **Knowledge Area: Transaction Processing**
- **Knowledge Area: Distributed Databases**
- **Knowledge Area: Physical Database Design**
- **Knowledge Area: Data Mining**
- **Knowledge Area: Information Storage and Retrieval**
- **Knowledge Area: Multimedia Systems**

Appendix B -
Data Visualization Examples

| Topic | Y73 | Y81 | Y86 | Y90 | Y02 | Y10 | |
|-------------------------------|---|-----|-----|-----|-----|-----|---|
| Database | Administration: security, safety, backup,.. | 1 | 2 | 1 | 1 | 2 | |
| | Data Models | 2 | 2 | | | | |
| | Data Quality: dimensions, assessment, i.. | 1 | | | 2 | 2 | |
| | Data sources and advanced types | | | 1 | 1 | 2 | |
| | Data Warehouse: design, conversions, r.. | | | | | 2 | |
| | Modeling and design, construction, sche.. | 2 | 2 | 1 | 3 | 4 | 4 |
| Information Assur.. | Triggers, Stored Procedures, Audit Contr.. | 1 | 1 | 2 | 3 | 2 | |
| | Information Assurance Model | | | 1 | 2 | 2 | |
| | Information Systems Email Management | | | | 3 | | |
| | Information Technology Security Principl.. | | | | 2 | | |
| | Privacy Impact | 2 | 1 | 2 | 2 | 3 | 2 |
| | Security Operations | | | | | 3 | 2 |
| IT Software Development | Security Policy | | 2 | | | 2 | |
| | Algorithmic Design, Data, Object and Fil.. | 2 | 2 | 2 | 3 | 3 | 2 |
| | Application Development-requirements, .. | | 3 | | 3 | 2 | |
| | Client Server Software Development | | | | 2 | 2 | |
| | Code Generators | | | | 1 | 1 | |
| | Computer Operations | | 3 | | | | |
| | File Systems | 3 | 3 | 2 | | | |
| | HCI Principles and Paradigms | | | | 3 | 1 | |
| | Information Systems | | 2 | | | 2 | |
| | Input Devices | 2 | 2 | 2 | | | |
| | Low Level Data Structures | 2 | 3 | 1 | 3 | 2 | 2 |
| | Machine Structures | 1 | 3 | | 1 | | |
| | Multiprogramming and Multiprocessing | 1 | 1 | | | | |
| | Object Oriented Languages | | | | | 1 | |
| | Output Devices | 2 | 2 | 2 | | | |
| | Problem Solving-identifying problems, s.. | | 3 | 2 | 2 | 3 | |
| | Procedural Languages | 3 | 3 | 2 | 3 | 3 | |
| | Programming-principles, objects, algorit.. | 3 | 3 | 3 | 4 | 3 | |
| | Prototype | | | 2 | | 2 | |
| | Storage Management | 3 | 1 | | | | |
| System of Programs | 2 | 2 | | | | | |
| Testing | 2 | 2 | 2 | 3 | 3 | | |
| Systems Integration | Communication system organization | | 1 | | | | |
| | Computer system hardware | 2 | 1 | | | | |
| | Computer Systems Hardware | 2 | 3 | 2 | 2 | 3 | |
| | Computer Systems Software-OS fundam.. | 2 | 3 | 2 | 1 | 2 | |
| | End-use interface | | 2 | | 2 | 2 | |
| | Inter-systems Communication | 2 | | | | | |
| | LAN/WAN Design and Management | | | | | 3 | 2 |
| | Networking (Lan/Wan) and Telecommun.. | | 1 | | 1 | 4 | 2 |
| Web De.. | Operating Systems Management-multi p.. | | | 1 | 1 | 2 | |
| | Systems Configuration, Operation, Admi.. | | | | | 2 | |
| | Web page Development-HTML, page edit.. | | | | | 3 | |
| | Web programming-thin client, asp, aspx.. | | | | | 2 | |
| Web Systems Development Tools | | | | | 2 | | |

Figure 3 - A Heat map of the level of coverage by topic within a knowledge Area and year

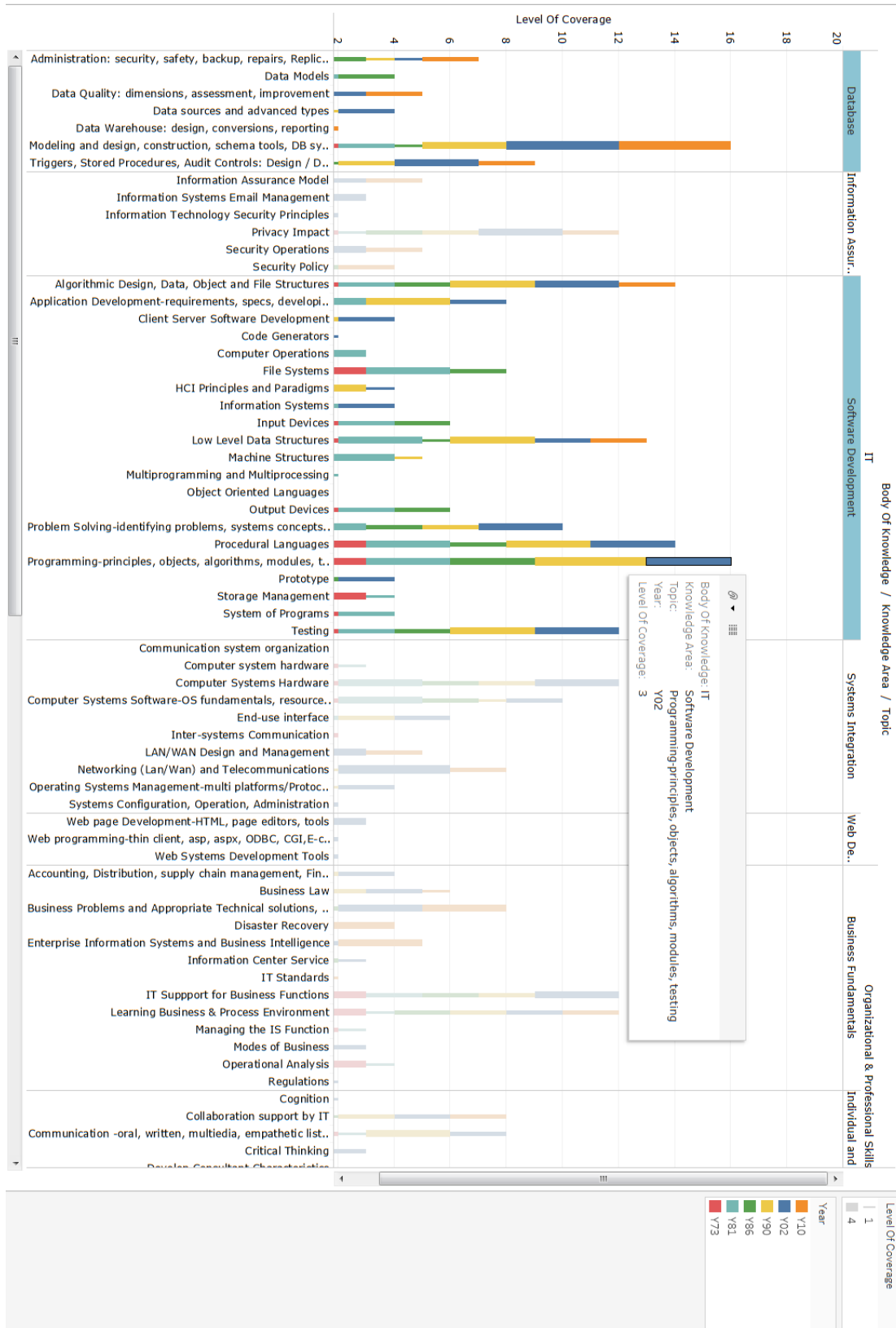


Figure 4 - A Bar chart-view showing Database and Software Development evolution and emphasis across the years

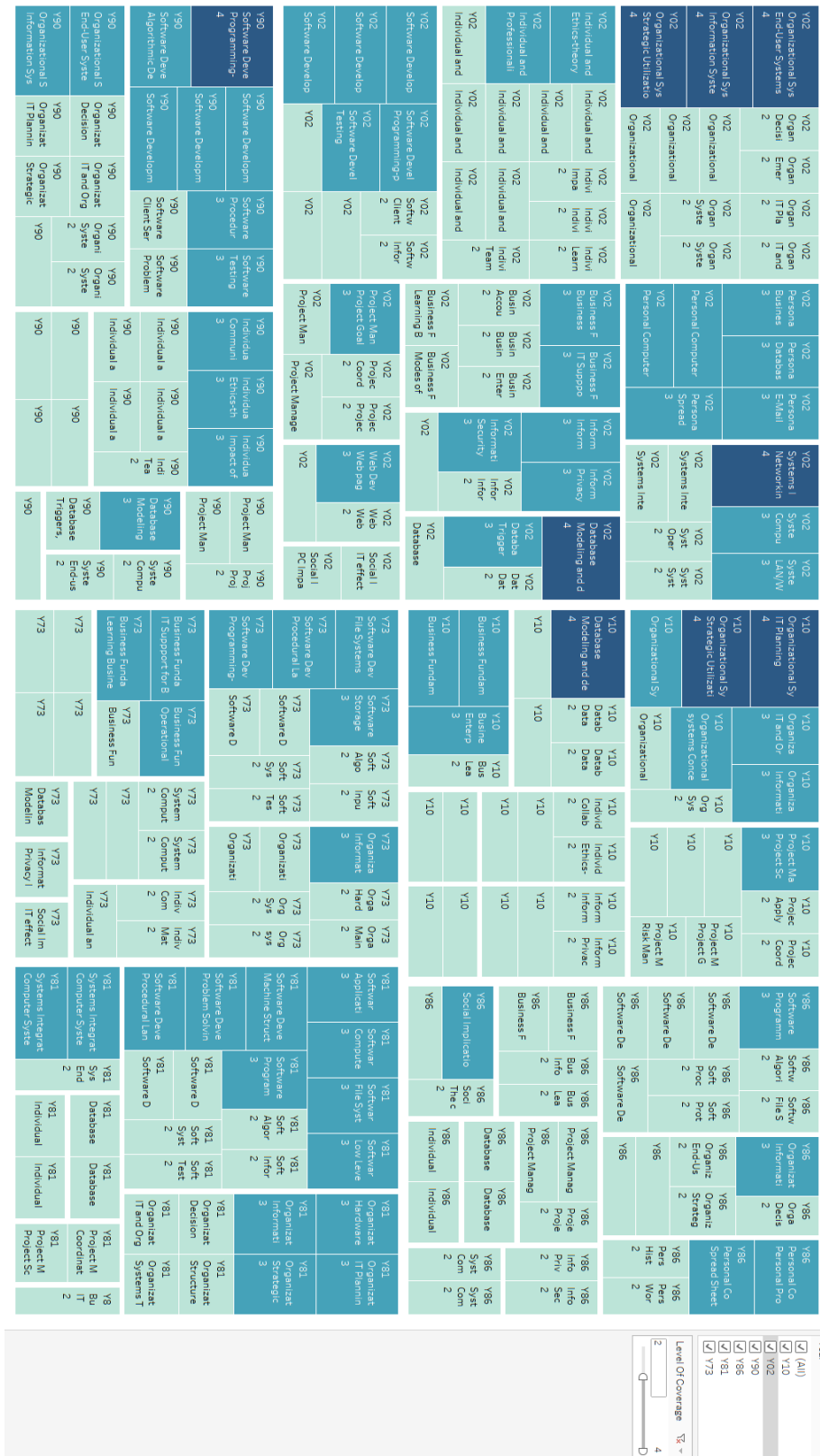


Figure 5 - A treemap View of the Topics across years where the skill is 2,3 or 4

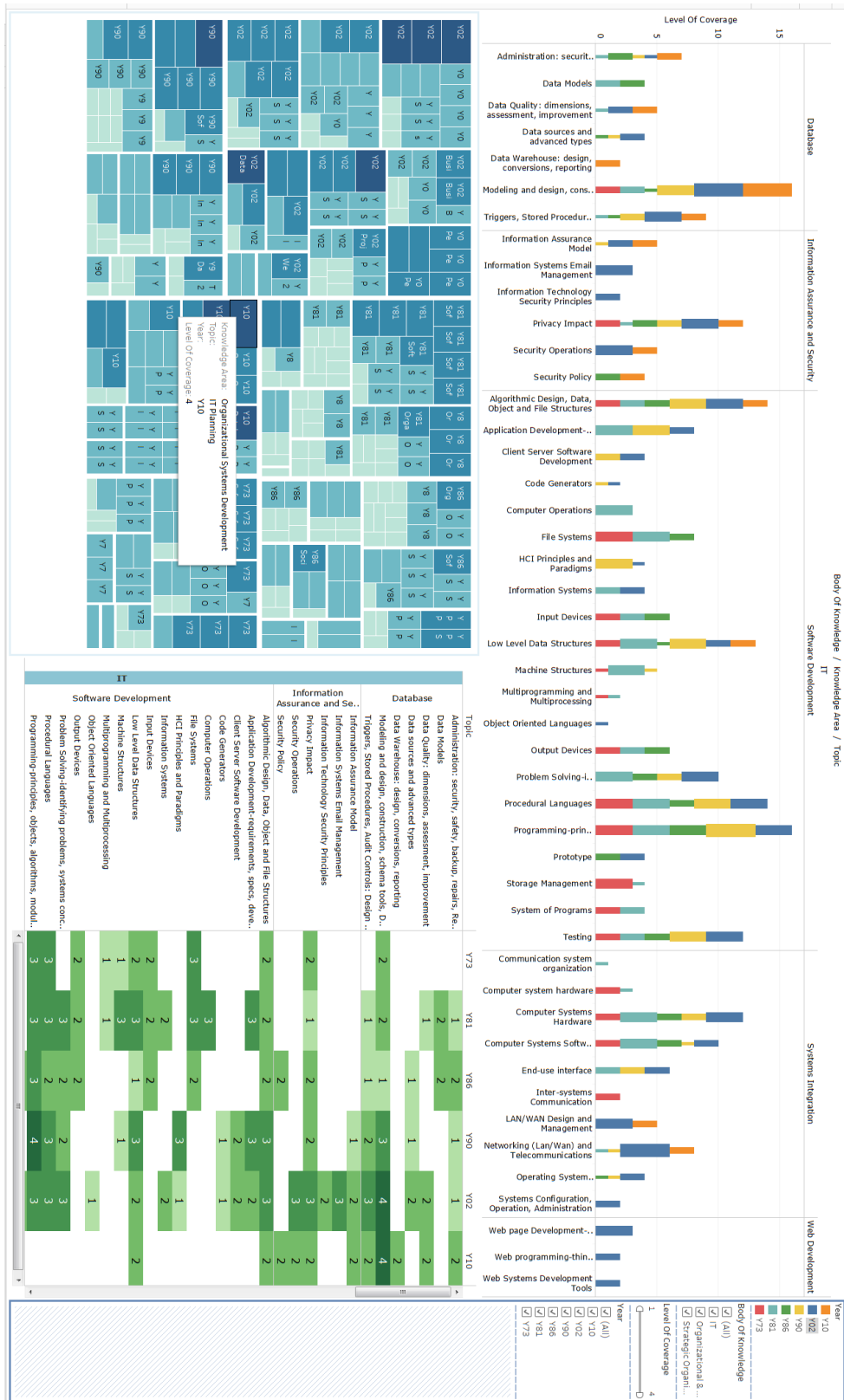


Figure 7 - An Interactive dashboard of the Multiple Views of the Data



Figure 8 - An R-ggplot2 of Table 4

The R-ggplot2 Code for Table 4 (Figure 8)

```
library(ggplot2)
ku <- read.csv("../xx.csv",
  stringsAsFactors = FALSE)
ggplot(ku, aes(x=KU, ymin=min, ymax=max,
  color=KU))+
  geom_linerange()+
  guides(color=FALSE)+
```



```
random.order=FALSE,  
colors=brewer.pal(8, "Dark2"), random.color=TRUE, rot.per=0.25,  
min.freq=1, max.words=Inf)
```