Volume 13, No. 1
January 2015
ISSN: 1545-679X

INFORMATION SYSTEMS EDUCATION JOURNAL

In this issue:

4. **Business Analytics in Practice and in Education: A Competency-based Perspective**
Stanislav Mamonov, Montclair State University
Ram Misra, Montclair State University
Rashmi Jain, Montclair State University

14. **Learning Styles, Online Content Usage and Exam Performance in a Mixed-Format Introductory Computer Information Systems Course**
Guido Lang, Quinnipiac University
Stephen D. O'Connell, CUNY Graduate School and University Center

23. **Evaluating Business Intelligence / Business Analytics Software for Use in the Information Systems Curriculum**
Gary Alan Davis, Robert Morris University
Charles R. Woratschek, Robert Morris University

30. **T0: Discrete Math and Programming Logic Topics as a Hybrid Alternative to CS0**
Nancy L. Martin, Southern Illinois University

45. **A Proposed Concentration Curriculum Design for Big Data Analytics for Information Systems Students**
John C. Molluzzo, Pace University
James P. Lawler, Pace University

58. **A Design Quality Learning Unit in OO Modeling Bridging the Engineer and the Artist**
Leslie J. Waguespack, Bentley University

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published by **EDSIG**, the Education Special Interest Group of AITP, the Association of Information Technology Professionals (Chicago, Illinois). Publishing frequency is six times per year. The first year of publication is 2003.

ISEDJ is published online (http://isedjorg). Our sister publication, the Proceedings of EDSIG (http://www.edsigcon.org) features all papers, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the conference. At that point papers are divided into award papers (top 15%), other journal papers (top 30%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 40%.

Information Systems Education Journal is pleased to be listed in the 1st Edition of Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org.

Information Systems Education Journal

Business Analytics in Practice and in Education: A Competency-based Perspective

Stanislav Mamonov
Stanislav.Mamonov@montclair.edu

Ram Misra
Ram.Misra@montclair.edu

Rashmi Jain
Rashmi.Jain@montclair.edu

IOM Department, School of Business
Montclair State University
Montclair, NJ 07043

Abstract

Business analytics is a fast-growing area in practice. The rapid growth of business analytics in practice in the recent years is mirrored by a corresponding fast evolution of new educational programs. While more than 130 graduate and undergraduate degree programs in business analytics have been launched in the past 5 years, no commonly accepted model of business analytics curriculum yet exists. Drawing on competency-based curriculum design literature, we take the first steps towards initiating a debate on the model curriculum in business analytics. We analyze a sample of business analytics job announcements from different industries and identify a preliminary set of business analytical competencies sought in practice. Further, we examine six existing graduate programs in business analytics, which reveal divergent approaches to business analytics curricula. These institutions were selected since they offered a graduate degree program in business analytics for at least two years. Our findings indicate that there are significant variations in the program structure in terms of program length (10 to 18 months) and flexibility (electives comprise 0 to 37% of the course work). We also found that the programs vary greatly in the coverage of both traditional analytics and the new emergent technologies and analytical methods. We conclude with a commentary on the emergent trends in business analytics in practice and the opportunities presented by these trends for the academia.

Keywords: business analytics, competency-based curriculum.

1. INTRODUCTION

Technology trends lead to a growing volume of available data

The convergence of several technological trends precipitated a rapid increase in the volume, velocity and variety of data that is now available to businesses. The first trend leading to an increasing volume of data is the miniaturization of computing technology, which facilitated pervasiveness of the embedded systems and mobile computing. Global smartphone shipments surpassed computer shipments in 2013 - over 1 billion smartphones were shipped worldwide (Hornyak, 2014). Modern phones feature multicore processors, memory and storage capacity that would have been the envy of the desktop computers not so long ago. Smartphones also commonly feature microphones, cameras, global positioning systems and accelerometers among other sensors, which can generate immense amounts of data potentially available for capture and analysis. The second complementary trend responsible for the increasing volume of available data is the continuous evolution of storage technology, increasing capacity accompanied by decreasing prices (McLellan, 2014), which enables capture and storage of a growing volume of data, much of which originates from mobile devices.

The third major trend contributing to the increasing volumes of available data is the coevolution of ubiquitous connectivity and social media, leading to the rapid growth in content creation, management, and dissemination. Empowered by smartphones, nearly anyone can capture pictures and videos, and quickly distribute the content. Content sharing statistics from popular social media sites illustrate this trend. Over 100 hours of video are uploaded to YouTube every minute (YouTube, 2014) and Facebook users share over 300 Petabytes of data each month (Traverso, 2013). The convergence of technological trends has led to a massive increase in the volume of data. IBM estimates that 90% of all available data was generated in the past two years and the trend is expected to continue with more data being generated in the coming years (IBM, 2013).

Data creates business opportunities

The availability of new data sources creates opportunities for business process optimization and, in some cases, for a complete reengineering of the way that business is done (Davenport, 2006). For example, customer feedback, one of the most valuable sources of information for businesses, historically was difficult and expensive to obtain. Social reviews posted on Yelp, TripAdvisor and other services now provide invaluable customer feedback information for business managers, pinpointing business strengths and areas for improvement. Social reviews generally follow closely in time with the actual consumer experiences and they are available to business managers at virtually zero cost. Marginal improvements in business efficiency can have a strong impact on business profitability (Soteriou & Zenios, 1999). It was estimated that in 2013 only 22% of the information in the digital universe was a candidate for analysis and less than 5% of that was actually analyzed. By 2020, the useful percentage is projected to grow to more than 35%, mostly because of the growth of data from embedded systems (IDC, 2014). Consequently industry thirsts for people who are able to turn new data into actionable business insights. McKinsey Global Institute estimates that by 2018 the industry will face a shortage of 1.5 million managers with data analytical skills able to translate analytical insights into practice (Manyika, Chui, Brown, Bughin, Dobbs, Roxburgh, 2011). McKinsey further projects that the industry will face a shortage of 140,000 to 190,000 people with deep data science expertise capable of leveraging large datasets.

Academic programs in business analytics

The growing need for business analytical skills has been recognized in academia. Over 130 academic programs in business analytics have been launched between 2007 and 2012 (Wixom et al., 2014). To the best of our knowledge, no model curriculum for programs in business analytics exists at present at either undergraduate or graduate level. Development of model curricula is commonly done by associations and professional societies, e.g. AIS, ACM, IEEE, which integrate input from academic institutions as well as industry experts (Bell, Mills, & Fadel, 2013; Carlsson, Hedman, & Steen, 2010). The development of a model curriculum for business analytics requires a broad coordinated effort among academia and practitioners and it is therefore outside of the scope of the current study. However, we take the first steps towards initiating a discussion concerning the structure of a model business analytics curriculum. We approach this topic from the point of view of competency-based curriculum design.

Competency-based curriculum development proceeds by 1) identifying a common set of skills which are in demand in practice and 2) development of an academic curriculum that empowers the graduates with the corresponding skillset (Bowden, 2004). Competency-based curriculum design has been applied in the development of graduate (Gorgone, Gray, & Stohr, 2006) and undergraduate curricula (Topi & Valacich, 2010) in Information Systems. In this study we take the initial steps of identifying a preliminary skillset associated with business analytics in practice. Further, we review the graduate business analytics curricula at several universities in the New York City metro area. We conclude with a commentary on the evolution of business analytics in practice and the opportunities for educational programs in business analytics.

2. BUSINESS ANALYTICS IN PRACTICE

Business analytics is commonly defined as skills, technologies, applications and practices for continuous iterative investigation of past business performance to gain insight and drive business planning (Beller & Barnett, 2009). To identify the skillset commonly expected for business analytics practitioners, we conducted a search of open position announcements using Indeed.com, a specialized search engine which indexes job postings across numerous company web sites as well as job posting aggregators. We used the keyword "business analytics" to identify open positions in the New York City metro area. We examined the job listings which were returned by the Indeed search engine and after iterative evaluation decided to retain a relatively short list of positions which 1) were offered at large established companies and 2) exemplified the skillset commonly expected in the industry for similar positions. Our rationale for focusing on the large established corporations is grounded in the expectation that large companies have more established business processes and more clearly defined job functions compared to smaller, less established companies (Humphrey, 1988). Our decision to focus on a limited number of representative positions stems from the observation that while specific industries and companies may have very distinct jobs requirements, our goal is to identify a common set of skills that is frequently required across different companies and industries. The positions selected for our analysis include the following:

- Data Visualization Consultant (Accenture)
- Data Analytics Manager (Deloitte)
- Business Intelligence Analyst (UBS)
- Compliance Office Analyst (Citibank)
- Data & Analytics Consultant (Accenture)
- Loan Operations Business Analyst (Capital One)
- Business Intelligence Architect (Nike)
- Customer Intelligence Analyst (PSEG)

Job descriptions posted by companies follow various formats, but they generally list the required skills. In order to develop a matrix representation of common skills required by each position, we draw on an often cited view of business analytics in practice, which suggests that business analytical skillset lies at the intersection of expertise from three domains 1) the specific business domain, 2) technical data management and programming expertise and 3) applied statistics. Figure 1 summarizes this view in a Venn diagram modeled after (Conway, 2013).

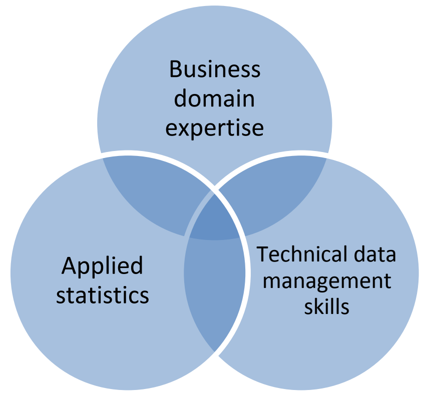


Figure 1. Business analytics skillset

The skill requirements across a representative set of positions are summarized in Table 1 in the Appendix. Our evaluation of the job requirements along the three domains in Figure 1 suggests that applied statistical skills required by the companies encompass both a theoretical understanding of statistical methods, as well as practical knowledge of software packages commonly used for statistical analysis – primarily SAS and R software. The job descriptions commonly require familiarity with regression modeling techniques. Application of regression analysis requires understanding of inherent assumptions underlying the regressions, and necessitates foundational statistical knowledge of distributions, sampling and statistical inference. Though not all job postings explicitly stated this requirement, we inferred the need for foundational statistical knowledge wherever the position required regression analysis expertise.

Data mining is a broad concept that encompasses many data model design and analytical techniques which generally include regression analysis among them (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). In our analysis we separated regression skills from the more advanced data mining methodologies, e.g. decision trees, neural networks, support vector machines as well as ensemble modeling techniques. Further, we also separately evaluated job requirements for text analytical skills, because analysis of textual data is a unique domain within data mining practice with specialized expertise related to processing and modeling of textual data (Aggarwal & Zhai, 2012). Considering that 80% of the world's data today is unstructured – these skills set are becoming extremely important.

The ability to locate, extract and prepare data for analysis is foundational for business analytics in practice. The required stated technical data management skills among the reviewed job postings span the range from the basic structured query language (SQL) competency in popular relational database management systems (RDBMS) to proficiency with large data set analysis leveraging Hadoop infrastructure. While SQL, RDBMS and data warehousing skills are nearly universally required across the positions which we reviewed, a growing number of positions also require competency with key-value stores, most commonly exemplified by Hadoop implementations in practice. Data warehousing job requirements often specifically call for experience with data extraction, transformation, loading (ETL) and cleaning. Further, two of the eight positions in our sample explicitly required expertise with Python programming language as the development platform for performing data processing and analysis.

Data visualization expertise was nearly universally required by the positions, which we included in our analysis. Data visualization represents an important area of practice. Virtually all positions in our set listed Tableau software as the dominant tool for data visualization, but several positions also suggested Qlikview as another potential software choice for data visualization. All positions emphasized the importance of soft skills: effective communication and presentation as well as the ability to work in groups, highlighting the fact that effective business analytics in practice often requires group collaboration and effective communication of insights across the enterprise. These skills become important in influencing the decision to implement the results of analytical exercise/analytics team.

In addition to specific knowledge of statistical methods and technical skills every position also included business domain specific expertise which qualified an ideal job candidate. These requirements are detailed in Table 2.

Position (Company)	**Industry specific requirements**
Data Visualization Consultant (Accenture)	Industry experience: financial services, healthcare, government
Data Analytics Manager (Deloitte)	Enterprise risk management, risk reporting, financial & regulatory reporting
Business Intelligence Analyst (UBS)	Securities research
Compliance Office Analytics (Citibank)	Anti-money laundering regulation & compliance
Data & Analytics Consultants (Accenture)	Industry experience: financial services, healthcare, high tech, government
Loan Operations Business Analyst (Capital One)	Financial auditing & risk management
Business Intelligence Architect (Nike)	High volume consumer data
Customer Intelligence Analyst (PSEG)	Customer operations/ experience

Table 2. Business domain specific expertise.

2. BUSINESS ANALYTICS CURRICULA

In the next step of our analysis we examined the curriculum structure of graduate programs in business analytics offered at several universities located in the New York metro area. In selecting the programs to be included in our analysis we focused on institutions, which have offered a graduate degree program in business analytics for at least two years. One exception to this requirement was the new program at New York University which officially launched in May 2014 (NYU, 2014). Our rationale for including the new degree program at NYU is grounded in that NYU piloted the courses comprising the new program over two years prior to launch. The new program represents a unique curriculum structure in business analytics education that may be of interest to universities looking to build a business analytics curriculum.

In reviewing the business analytics programs we specifically examined the required courses that are included in each program as well as the range of elective courses which are available. The summaries below provide the list of core and elective courses for each program that we examined. The universities are listed in alphabetical order. It is important to note that all academic programs evolved over time and the summaries in the table below present the information which was available on the universities' web sites in June 2014.

Fordham University

Degree: MS in Analytics
Program structure: 30 academic credits (3 semesters)

Required courses:

- Database management
- Data warehousing
- Data Mining for business
- Business analytics for managers
- Text analytics
- Web analytics
- Business performance & Risk management

plus 3 electives.

New Jersey Institute of Technology

Degree: MS in Information Systems with concentration in Business Analytics
Program structure: 10 courses, 1.5 years

Required courses:

- User Experience Design
- Data Analytics for IS
- Business Process Innovation
- System Analysis & Design
- Enterprise Database Management

+1 of the following:

- Information Retrieval
- Transaction Mining and Fraud Detection
- Web Mining

Electives:

- Data Management
- Business Decision Making
- Security & Network Management
- Web Systems

New York University

Degree: MS in Business Analytics
Program structure: 14 courses, 10 months.

Required courses:

- Social Media and Digital Marketing Analytics
- Foundations of Statistics Using R
- Prediction
- Data Mining for Business Analytics
- Data Driven Decision Making
- Network Analytics
- Decision Models
- Operations Analytics
- Advanced Decision Models
- Data Visualization
- Special Topics in Analytics: Revenue Management & Pricing
- Strategy, Change and Analytics
- Market Modeling
- Capstone

Rutgers University

Degree: Masters of Business & Science in Analytics
Program structure: 18 credits in science + 19 credits in business

Required courses:

- Fundamental of Analytics
- Advanced Analytics & Applications
- Regression Analysis

+1 course from

- Database Design and Management
- Database Systems
- Database System Engineering
- Advanced Database Systems

+1 course from

- Introduction to Parallel and Distributed Computing
- Parallel and Distributed Computing
- Programming Methodologies for Numerical Computing and Computational Finance
- Applications of Parallel Computers

+3 electives

Stevens Institute of Technology

Degree: MS in Business Intelligence and Analytics
Program structure: 36 academic credits, 1.5 years

Required courses:

- Financial Decision Making

Strategic Data Management Data Warehousing and Business Intelligence Process Analytics and Optimization Financial Enterprise Risk Engineering Multivariate Data Analytics Experimental Design Knowledge Discovery in Databases Statistical Learning & Analytics Social Network Analytics Web Analytics Industry practicum
University of Connecticut Degree: MS in Business Analytics and Project Management. Program structure: 8 required courses + 3 electives, 1.5 years. Required courses: Business Analytics: Business Process Modeling and Data Management Predictive Modeling Business Decision Modeling Data Mining and Business Intelligence Project management: Introduction to Project Management Project Leadership and Communications Project Risk and Cost Management Advanced Business Analytics and Project Management Electives: Real-time Enterprise Data Integration and Audit Data Analytics with R Adaptive Business Intelligence Big Data Analytics with Hadoop Ethical and Legal Issues in Project Management Managing International Development Projects Agile Project Management Graduate Field Study Internship

Table 3. Business Analytics Programs.

Examination of the business analytics curricula across the six graduate programs yields a number of observations. First, there is a significant variation in the program length and flexibility. The NYU program is estimated to require 10 months to complete, while the programs at NJIT, Stevens Institute of Technology and the University of Connecticut require 18 months. In terms of flexibility, the programs offered at Stevens Institute of Technology and the new program at NYU are comprised entirely of required courses. On the other side of the spectrum, the programs offered at other universities offer electives, which comprise up to 37% of the total credits required by the programs. These observations suggest the existence of divergent views on the core business analytics skillset across the educational institutions, which echo the multitude of perspectives that exist in practice on the core business analytics skillset.

Our second observation is that the course allocation across the three areas, which contribute to business analytics: business expertise, applied statistical analysis and technical skills, reveals a diversity of approaches across the academic programs. For example, the MB&S in Analytics program at Rutgers University, allocates 3 core courses to topics on applied statistical analysis. On the other hand, the program at Fordham University incorporates statistical analysis within the broader subjects of business analytics and specific applications of data mining. There is also a difference in terms of the emphasis given to specific tools used for statistical analysis. R software has become the *de facto* standard for statistical analysis in practice (Muenchen, 2014). Two of six programs in our sample offer dedicated courses focusing on the development of R skills. The NYU program in business analytics includes a required course, which covers Foundations of Statistics Using R, and the program at the University of Connecticut offers an elective Data Analytics with R.

There is also significant variation across the programs in our sample in the emphasis placed on data management skills. The programs at Fordham University and Stevens Institute of Technology require two courses in database management and data warehousing. The programs at University of Connecticut and NJIT offer database management courses among the electives, while the program at NYU does not include a course on database management, though related topics are discussed within other courses comprising the program. This might also have to do with how these programs have evolved and the legacy behind them.

Our third observation is that the business analytics programs in our sample often leverage existing institutional strengths. For example, the program at the University of Connecticut places a strong emphasis on the development of project management skills among the graduates – four of eight required courses within the program focus on project management. NJIT program in business analytics offers another example of leveraging institutional strengths within the business analytics program. The NJIT degree requires students to take courses on user experience and system design as a part of the core analytics curriculum.

Our fourth observation concerns the integration of internships and industry practicums in the curriculum. Industry practicums and internships have long been recognized for their role in improving information systems graduates preparedness for industry employment (Gorgone, Davis, & Valacich, 2003). Two of the six academic programs in our review, NYU and Stevens Institute of Technology, mandate an industry practicum for the degree completion, while the remaining four programs offer it as an elective.

4. DISCUSSION, GAP ANALYSIS AND CONCLUSION

Our analysis of industry business analytics job postings reveals a very healthy market demand for people with business analytical skills. In June 2014, in New York City there were over 5800 business analytics positions paying $60,000 or more with over 1100 jobs paying $140,000 or more. Our examination of the sample of positions offered by large established firms reveals that the companies expect successful candidates to have expertise in data management, applied statistics, and specific business domains, as well as to possess effective communication and presentation skills, and to work well within teams. The job postings universally expect candidates to have RDBMS, SQL and data warehousing competencies. 2 of 8 (25%) positions in our sample also required familiarity with NOSQL and Hadoop. Our findings suggest that the ability to handle (ETL) structured data using traditional relational database technologies remains the core of business analytics in practice, but a growing number of positions also require Big Data expertise exemplified by Hadoop and the newer NOSQL databases.

In terms of applied statistical knowledge, the positions in our sample nearly universally expect the candidates to have foundational statistical knowledge that extends to linear and logistic regression modeling. 6 of 8 (75%) of positions in our sample also required advanced data modeling expertise (decision trees, neural networks, support vector machines, and ensemble modeling techniques). 4 of 8 (50%) positions also required text analytics expertise. These observations suggest that companies are urgently in need of employees who can apply state of the art modeling techniques to make sense of the growing volume of data, including textual data.

Our examination of the specific software skills required by the positions in our sample reveals that Excel remains the workhorse in practice – it is required by 75% of positions in our sample. An important discovery in our analysis is that Tableau software expertise is required by 7 of 8 jobs in our sample. Tableau software offers an intuitive dashboard-driven approach to analytics and it has enjoyed rapid and broad adoption in practice (Pacampara, 2014). It appears that Tableau expertise has become as essential as Excel expertise for business analytics practitioners today.

In assessing the structure of existing graduate-level educational programs in business analytics we find significant variation in the program structure in terms of program length (10 to 18 months) and flexibility (electives comprise 0 to 37% of the course work). We also find that the programs vary greatly in the coverage of both traditional analytics (RDBMS, SQL, data warehousing) and the new emergent technologies (Hadoop, NOSQL) and analytical methods. Our findings echo observations made at a recent discussion in the Special Interest Group on Decision Support Systems (SIGDSS) of business analytics education which lamented the lack of universal inclusion of foundational data analytics skills (RDBMS, SQL, data warehousing) and advocated for the courses covering these areas to be included in the core business analytic curriculum (Wixom et al., 2014).

Our analysis of the business analytics job market also suggests several additional knowledge domains/skills, which may need to be developed within the business analytics curricula. First, we find that text analytics is very much in demand in practice, but is poorly represented within the business analytics curricula.

Text analytics has evolved its own set of analytical techniques and tools (Liu & Murphy, 2013), and business analytics programs would be well served by including a Text Analytics course. Another area requiring further development is the connection between education and practice. The recent SIGDSS discussion noted that employers are dissatisfied with the practical experience of business analytics graduates (Wixom et al., 2014). In our analysis only 2 of 6 programs require industry practicums. Business analytics appears to be particularly well suited for a closer collaboration between the academia and the industry. Internships and industry practicums would likely help the graduates to transfer their newly acquired business analytical skills from the classroom to practice (Topi & Donnellan, 2014).

In conclusion, our findings are consistent with the previous calls for Information Systems departments to take on the leadership role in addressing the growing industry need for business analytics (Sidorova, 2013). Information Systems are particularly well positioned to develop effective educational offerings in the area of business analytics, because the topics of technology procurement and management, as well as the strategic role of information technology in business have been the traditional focal points for IS in research and education. Our results suggest that while the traditional business analytical technologies (RDBMS, SQL) remain very relevant in practice today, the emergent areas of Big Data analysis (Hadoop, NOSQL) and specialized analytics (text data analysis) present attractive growth areas in practice that need to be addressed within the educational domain as well. A closer collaboration with the industry in developing these offerings would serve well all the stakeholders: students, faculty, the educational institutions and industry.

5. REFERENCES

Aggarwal, C. C., & Zhai, C. (2012). *Mining text data*. Springer.

Bell, C., Mills, R., & Fadel, K. (2013). An Analysis of Undergraduate Information Systems Curricula: Adoption of the IS 2010 Curriculum Guidelines. *Communications of the Association for Information Systems*, *32*.

Beller, M. J., & Barnett, A. (2009). Next generation business analytics. *Lightship Partners LLC, Www. Slideshare. net/LightshipPartners/nextgeneration-Business-Analytics-Presentation*.

Bowden, J. (2004). Capabilities-driven curriculum design. In *Effective learning and teaching in engineering* (pp. 36–47).

Carlsson, S., Hedman, J., & Steen, O. (2010). Integrated curriculum for a bachelor of science in business information systems design (BISD 2010). *Communications of the Association for Information Systems*, *26*.

Conway, D. (2013). THE DATA SCIENCE VENN DIAGRAM. Retrieved from http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

Davenport, T. H. (2006). Competing on analytics. *Harvard Business Review*, *84*(1), 98.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, *17*(3), 37.

Gorgone, J., Davis, G., & Valacich, J. (2003). IS 2002 model curriculum and guidelines for undergraduate degree programs in information systems. *Communications of AIS*, *11*.

Gorgone, J., Gray, P., & Stohr, E. (2006). MSIS 2006: model curriculum and guidelines for graduate degree programs in information systems. *Communications of the AIS*, *17*(2), 121–196.

Hornyak, T. (2014). 1 billion smartphones shipped worldwide in 2013. *PCWorld*.

Humphrey, W. S. (1988). Characterizing the software process: a maturity framework. *Software, IEEE*, *5*(2), 73–79.

IBM. (2013). Apply new analytics tools to reveal new opportunities. Retrieved from http://www.ibm.com/smarterplanet/us/en/business_analytics/article/it_business_intelligence.html

IDC. (2014). *The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things*.

James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, A. H. B. (2011). Big data: The next frontier for innovation, competition, and productivity. *McKinsey Global Institute Report*, *May*.

Liu, M. (Xiang), & Murphy, D. (2013). The Document Explosion in the World of Big Data – Curriculum Considerations. *2013 Proceedings of the Information Systems Educators Conference*, *30*, 1–9.

McLellan, C. (2014). Storage in 2014: An overview. Retrieved from http://www.zdnet.com/storage-in-2014-an-overview-7000024712/

Muenchen, R. A. (2014). The Popularity of Data Analysis Software.

NYU. (2014). NYU - Master of Science in Business Analytics. Retrieved from http://www.stern.nyu.edu/programs-admissions/global-degrees/business-analytics

Pacampara, B. (2014). Why Tableau Software Inc Will Keep Flying in 2014. Retrieved from http://www.fool.com/investing/general/2014/01/29/why-tableau-software-inc-will-keep-flying-in-2014.aspx

Sidorova, A. (2013). Business Analysis as an Opportunity for IS Programs in Business Schools. *Communications of AIS*, *33*, 521–540.

Soteriou, A., & Zenios, S. A. (1999). Operations, quality, and profitability in the provision of banking services. *Management Science*, *45*(9), 1221–1238.

Topi, H., & Donnellan, B. (2014). Moving Toward the Next Generation of Graduate Degree Programs in Information Systems. *Communications of the Association for Information Systems*, *34*(1).

Topi, H., & Valacich, J. (2010). IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. *Communications of the AIS*, *26*.

Traverso, M. (2013). Presto: Interacting with petabytes of data at Facebook. Retrieved from https://www.facebook.com/notes/facebook-engineering/presto-interacting-with-petabytes-of-data-at-facebook/10151786197628920

Wixom, B., Ariyachandra, T., Douglas, D., Goul, M., Gupta, B., Iyer, L., … Turetken, O. (2014). The Current State of Business Intelligence in Academia: The Arrival of Big Data. *Communications of the Association for Information Systems*, *34*, 1–13.

YouTube. (2014). Youtube viewership statistics. Retrieved from http://www.youtube.com/yt/press/statistics.html

Appendix

Table 1. A summary of skill requirements for positions in business analytics.

	Data Visualization Consultant (Accenture)	Data Analytics Manager (Deloitte)	Business Intelligence Analyst (UBS)	Compliance Office Analytics (Citibank)	Data & Analytics Consultants (Accenture)	Loan Operations Business Analyst (Capital One)	Business Intelligence Architect (Nike)	Customer Intelligence Analyst (PSEG)
Applied statistics								
Distributions, sampling & statistical inference	✓	✓	✓	✓		✓	✓	✓
Linear regression	✓	✓	✓	✓		✓	✓	✓
Logistic regression	✓	✓	✓	✓		✓	✓	✓
Advanced data mining techniques	✓	✓	✓	✓			✓	✓
Text analysis	✓	✓		✓			✓	
Technical skills								
Data storage/extraction								
Relational databases & SQL	✓	✓	✓	✓	✓	✓	✓	✓
Data warehousing	✓	✓		✓	✓	✓	✓	
NOSQL databases	✓				✓			
Hadoop	✓				✓			
Python	✓						✓	
Analytical software								
Excel	✓	✓	✓	✓		✓		✓
SAS	✓					✓		
R							✓	
Tableau	✓	✓	✓	✓	✓	✓		✓
Qlikview	✓	✓			✓			
Soft skills								
Communication & Presentation	✓	✓	✓	✓	✓	✓	✓	✓
Teamwork	✓	✓	✓	✓	✓	✓	✓	✓

Learning Styles, Online Content Usage and Exam Performance in a Mixed-Format Introductory Computer Information Systems Course

Guido Lang
guido.lang@quinnipiac.edu
Quinnipiac University
Hamden, CT 06518

Stephen D. O'Connell
soconnell@gc.cuny.edu
CUNY Graduate School and University Center
New York, NY 10035

Abstract

We investigate the relationship between learning styles, online content usage and exam performance in an undergraduate introductory Computer Information Systems class comprised of both online video tutorials and in-person classes. Our findings suggest that, across students, (1) traditional learning style classification methodologies do not predict behavioral measures of online learning, and (2) working on the online content specifically during allotted class time is positively related to exam performance. Controlling for differences across students, we find (3) accessing content on non-class days (consistency) is positively related to exam performance, while (4) working substantially ahead of the scheduled content pace is negatively related to exam performance.

Keywords: learning styles, online content usage, exam performance, mixed-format teaching

1. INTRODUCTION

The use of online learning systems in higher education and beyond has dramatically increased in recent years (Azarnush et al., 2013). Due to their web-based nature, online learning systems allow for the automatic collection of usage data. This, in turn, offers researchers and educators new opportunities to understand and improve student learning.

Recent technological and methodological advances present an array of techniques that hold the potential to gain deep insights from these data. These advances have led to novel findings in such disparate fields as medicine, marketing, logistics, and education. The field of education is particularly concerned with developing methods for exploring the unique types of data that come from educational settings, and using those methods to better understand students and the settings in which they learn (Baker & Yacef, 2009). The present work aims to bridge nascent literatures, which use such data to predict individual learning styles as well as learning outcomes.

Little is currently known about systematic differences across students in online content usage, and if these differences are associated with particular learning styles and ultimately with exam performance. In particular, one question yet to be addressed is whether traditionally-measured learning styles are useful for predicting online content usage. Early work establishing conceptual models used to define learning styles dates to the mid-1980s, suggesting such conceptualizations may not apply as readily to today's learning environment.

Specifically, the present work aims to investigate the relationship between learning styles and online content usage and the relationship between online content usage and exam performance. The work thus has theoretical implications for learner classification methodologies and practical implications for educators in mixed-format classroom settings.

2. LITERATURE REVIEW

Kolb's (1984) seminal treatise on defining student learning styles gave rise to a literature in the educational field implementing the "Kolb Model" to characterize students and assess their responsiveness to varied educational methods. Since then, new models of learning styles have come into practice, building on the work of Kolb (1984). Recent years have seen an explosion of data related to student use and interaction with learning content via computerized delivery methods, and a corresponding emergence of literature which applies conceptual learning style models to these new sources of information about learners (Khan et al. 2009, among others). The proposed work seeks to add to nascent literature which bridges conceptual models of learning styles (and their relation to learning outcomes) with the wealth of data now available to educators and researchers via new delivery platforms.

The work relates to two distinct strands of literature. The first uses data from web-learning platforms to predict student performance. Baugher et al. (2003) were among the earliest to study this relationship, examining whether total hits or inter-class consistency of hits to a course content web site have any value for predicting student performance in a course supplemented with online activities, finding stronger effects for the latter. Grabe et al. (2005) followed with a similar investigation, finding positive effects of the availability of online materials on both class absence and overall grades. In contrast to these early studies, Abdous et al. (2012) found little relationship between online activity and performance. Romero et al. (2013) use and compare the results of an array of data mining methods to predict student grades based on online usage data.

Studies comprising a second and more recent literature use advanced analytical techniques and detailed web usage information to predict student learning styles. Lu et al. (2007) and Hung and Zhang (2008) are among the first of these, investigating the relationship between learning style measures, online behavior and learning outcomes. More recent studies on the same topic include Ballenger and Garvis (2009), Hung and Crooks (2009) and Bousbia (2010). Clewley et al. (2011) and Azarnoush et al. (2013) use related methods to detect learner styles, and examine how content delivery systems can be adapted dynamically to adjust to individual-user learning types. Recent reviews which more comprehensively detail these and related studies can be found in Romero and Ventura (2007), Baker and Yacef (2009) and, specifically for business-related disciplines, Arbaugh et al. (2009).

Overall, existing studies yield some common conclusions. First, it is important to be aware of, and also detect, student learning styles—especially when using online courses, and that consideration should be given to the diversity of learning styles when designing and developing online learning modules (in terms of content presentation and design features, for example). Second, educational data mining combined with traditional statistical analysis can give a deeper understanding of the determinant of student learning and performance. The current study seeks to build on the work of Lu et al. (2007) and Hung and Zhang (2008) in that we apply data mining techniques to investigate the relationship between learning styles (as measured by a separately-administered survey) and online content usage, and then assess the relationships between online content usage and exam performance.

3. METHODOLOGY

We use a combination of server log files, exam scores, and surveys which were collected over the course of the Spring 2014 semester in three sections of CIS 101 (Introduction to Information Systems) at a mid-sized private university in the northeastern United States. CIS 101 introduces students to various aspects of developing and managing computer information systems and is a required class for all business freshmen. As part of CIS 101, students received three weeks of intensive Microsoft Excel training online (referred to as the Excel Boot Camp). The Excel Boot Camp content is delivered online and consists of 22 lessons, each of which consists of a short video tutorial and an exercise. Although the Excel Boot Camp is delivered online, students were still required to attend class (Mondays, Wednesdays, and Fridays for 50 minutes each). In each class, students were asked to work on the Excel Boot Camp individually. Although the Excel Boot Camp is self-paced, students were required to submit their completed exercises according to a pre-defined schedule (averaging about seven exercises per week).

The web-based nature of the platform allowed us to capture user interaction with the content as recorded in server log files. The log files contain information about each user's online content usage, such as login and logout times, as well as the time spent on each page. Consequently, the log files provide a rich source to quantify various aspects of student online content usage. Specifically, the following behavioral measures were calculated in order to quantify online content usage:

- *time online during class time* (hours spent viewing online content while being in class),
- *consistency* (number of non-class days during which a student visited the online content before the beginning of the exam study period),
- *time online in exam study period* (hours spent viewing online content during the exam study period), and
- *time online working ahead* (hours spent working ahead of the class before the exam study period).

Two weeks after the end of the three-week Excel Boot Camp, students were tested on their knowledge of Microsoft Excel. The exam consisted of 20 multiple-choice questions, most of which require students to download an Excel worksheet and perform analyses in order to derive an answer. All exam questions were directly linked to one of the 22 lessons in the Excel Boot Camp. Given the two-week lag between the completion of the Excel Boot Camp and the exam, students were encouraged to go back and review lessons in the Excel Boot Camp during the exam study period.

At the end of the semester, students were asked to complete a survey on learning styles. The survey was comprised of questions which form the basis for the Index of Learning Styles (Felder & Silverman, 1988; Felder & Spurlin, 2005) and the Kolb Learning Style Inventory (Smith & Kolb, 1986).

The Index of Learning Styles (hereafter "ILS") assesses learning preferences on four dimensions. Each of the four scales consists of 11 items. For each item, students complete a sentence by choosing one of two options representing opposite ends of the dimension. The four dimensions are (see Felder & Spurlin, 2005):

- *sensing* (concrete, practical, oriented toward facts and procedures) or *intuitive* (conceptual, innovative, oriented toward theories and underlying meanings),
- *visual* (prefer visual representations of presented material, such as pictures, diagrams, and flow charts) or *verbal* (prefer written and spoken explanations),
- *active* (learn by trying things out, enjoy working in groups) or *reflective* (learn by thinking things through, prefer working alone or with one or two familiar partners), and
- *sequential* (linear thinking process, learn in incremental steps) or *global* (holistic thinking process, learn in large leaps).

The Kolb Learning Style Inventory (hereafter "Kolb LSI") assesses students' preference for perceiving and processing information. It is based on Kolb's experiential learning theory (Kolb, 1985), which posits that how a person perceives information can be classified as concrete experience or abstract conceptualization, and how a person processes information can be classified as active experimentation or reflective observation (Simpson & Du, 2004). The Kolb LSI asks students to rank order four endings for 12 sentences according to how well they think each one fits them. Each of the four endings represents one of the four dimensions in Kolb's experiential learning theory, which can be described as (see Lu et al., 2007):

- *concrete experience* (tends towards peer orientation and benefits most from discussion with fellow learners),
- *abstract conceptualization* (tends to be oriented more towards symbols and learns best in authority-directed, impersonal learning situations, which

emphasized theory and systematic analysis),

- *active experimentation* (tends to be an active, "doing" orientation to learning that relies heavily on experimentation and learns best while engaging in projects), and
- *reflective observation* (tends to rely heavily on careful observation in making judgments).

4. RESULTS

A total of 91 students were enrolled in three sections of CIS 101. All students completed the Excel Boot Camp and the accompanying exam. Of those, 82 (90%) completed the end-of-semester survey on learning styles. As seen in summary statistics presented in Table 1 (see Appendix), students spent far more time online during class time (M = 6.36, SD = 2.42) as they spent online working ahead (M = 1.10, SD = 0.79). On average, students visited the online content on eight of the 15 non-class days before the beginning of the exam study period. During the exam study period, students spent on average less than half an hour accessing the online content (M = 0.28, SD = 0.45).

The correlations between all measures used in this study are presented in Table 2 (see Appendix). With regards to learning styles, we found only small to moderate correlations between the ILS and the Kolb LSI (all $r \leq 0.32$), suggesting that the two instruments measure different aspects of learning styles. The largest correlation is between the active-reflective dimension of the ILS and the active experimentation dimension of the Kolb LSI. This is not surprising, given that both of these dimensions measure a preference for "learning by doing".

There is also substantial variation in the learning styles across students in the class, suggesting that the class composition was not overly skewed in terms of attracting only a certain type (or certain types) of students, as show in Figure 1 (see Appendix).

In order to examine the relationship between learning styles and online content usage, we conducted ordinary least squares regressions of learning style dimensions (both ILS and Kolb LSI) on the various measures of online content usage. Although there is substantial variation in the distribution of the types of students taking the class (Table 1 and Figure 1), the regression results in Table 3 (see Appendix) show that neither of the learning style typologies yields strong predictions of online content usage. Among the learning style measures captured by our survey, only one type has a statistically significant relationship with any of the online content usage measures: a higher score on the "reflective observation" component of the Kolb LSI is associated with a higher likelihood of working ahead of schedule (although the effect is small: β = .035, p < .05). This finding suggests that students who are more reflective learners, and thus tend to rely more on careful observation, may be more intrigued to work ahead and explore the content ahead of class than students who are less reflective learners.

We next investigate the relationship between online content usage and exam performance at two levels of aggregation: between students (i.e. across students) and within students (i.e. across lessons controlling for student-level differences). The latter level of aggregation is possible due to the server log data identifying which specific lessons a user was viewing, and for how long, and by linking exam questions to specific lessons we are able to construct a topic level panel. The estimating equation for the between-students regression is the following:

$$(1)\ \boldsymbol{Score_i = \beta_0 + \beta_1 * class\ time\ spent\ online_i + \beta_2 * time\ spent\ online\ during\ exam\ study_i + \beta_3 * consistency_i + \beta_4 * time\ spent\ working\ ahead_i + u_i}$$

where i indexes students, and all regressors are initially measured as described above.

The results of the between-students regression analysis are presented in Table 4 (see Appendix). Across students, time spent online during class time (as opposed to self-study outside of class, as measured in various ways) is the single best predictor of exam performance (β = .72, p < .01). In other words, for every hour (which is slightly longer than one class period) that students accessed the online content during class time, their exam performance increased by 0.72 points (out of 20, equivalent to 3.6 percentage points). This suggests that working on online content during class time is more effective than working on online content outside of class time. Given that this measure in essence captures class attendance, this finding suggests that irrespective or student-level differences, the single most important factor influencing exam performance is coming to class and working on the assigned online content. Although this finding might not seem particularly novel at first, we believe this points to the importance of blended learning strategies that combine online and in-person classes.

The results of the between-students regression analysis paint a different, more nuanced picture. The estimating equation is:

$$(2)\ \boldsymbol{Score_{i,j}} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1} * \boldsymbol{class\ time\ spent\ online_{i,j}} + \boldsymbol{\beta_2} * \boldsymbol{time\ spent\ online\ during\ exam\ study_{i,j}} + \boldsymbol{\beta_3} * \boldsymbol{consistency_{i,j}} + \boldsymbol{\beta_4} * \boldsymbol{time\ spent\ working\ ahead_{i,j}} + \Gamma * \mathrm{X}_j + \boldsymbol{u_{i,j}}$$

where *i* still indexes students, *j* indexes lessons, and X_j is a vector of student fixed effects.

Table 5 (see Appendix) contains the results of the estimation of equation (2). When accounting for individual differences, time spent online during class time is no longer a significant predictor for exam performance (β = .0004, p > .05). In contrast, we found that consistency, as measured by the number of non-class days during which a student used the online content, is significantly related to exam performance (β = .073, p < .001). In other words, for every non-class day that students accessed the online content for a particular lesson, their chance of correctly answering the exam question relating to that lesson increased by 7.3 percentage points. Interestingly, the significant negative coefficient for time spent online working ahead of the class (β = -0.11, p < .01) suggests that working ahead decreases students' exam performance. Specifically, this suggests that for every hour spent working ahead of the class, students' chance of answering the exam question relating to the lesson that they should be working on decreased by 11 percentage points.

5. CONCLUSIONS

The aim of this study was to understand the relationships between learning styles, online content usage, and exam performance. We analyzed a unique dataset that gave us rich information on student access patterns of online content that was part of a hybrid university course. Our findings suggest that (1) traditional typologies of learning styles may not accurately classify the different ways students have of interacting with online content, (2) the number of hours spent working on the online content during class time is positively related to exam performance, (3) the number of non-class days during which students access online content is positively related to exam performance, and (4) the number of hours that students work ahead of the class is negatively related to exam performance. Based on these empirical findings, one can deduce three prescriptive guidelines for instructors using online content in their classes: (1) ensure that students work on the online content during class, (2) encourage students to review online content between classes, (3) discourage students from working ahead of the class. These suggestions should help students make the best use of the online content and improve exam performance when delivering content in a hybrid format.

There are a few caveats to the analysis. Specifically, a relatively small sample from a private university in the Northeastern US is probably not representative of the general undergraduate student population. Also, the hybrid class format of combining video-based tutorials with in-person class meetings is unique, which might limit the applicability of our findings to traditional in-person or purely online classes. Lastly, we collected the survey of learning styles at the end of the semester. It is possible, though we believe unlikely, that the experience of the Excel Boot Camp had an effect on students' responses to the survey.

This study adds to an emerging literature using large datasets that capture detailed information on student interaction with educational content to detect patterns of content usage and predict learning outcomes. There are several directions in which future work in this area can go. The first is to use such data collection for semester-long classes, which will provide richer variation to use in classifying student interaction with learning materials. Another will build on this type of data to generate new learning style typologies that are more suited to online learning behaviors.

6. REFERENCES

Abdous, M., He, W. & Yen, C.-J. (2012). Using Data Mining for Predicting Relationships between Online Question Theme and Final Grade. *Educational Technology & Society*, 15(3), 77–88.

Arbaugh, J.B. et al. (2009). Research in online and blended learning in the business disciplines: Key findings and possible future directions. *The Internet and Higher Education*, 12(2), 71–87.

Azarnoush, B. et al. (2013). Toward a Framework for Learner Segmentation. *Journal of Educational Data Mining*, 5(2), 102–126.

Baker, R.S.J.D. & Yacef, K. (2009). The State of Educational Data Mining in 2009: A Review and Future Visions. *Journal of Educational Data Mining*, 1(1), 3–16.

Ballenger, R.M. & Garvis, D.M. (2009). Student Usage of Instructional Technologies: Differences in Online Learning Styles. *Proceedings of the Information Systems Education Conference*, 1–19.

Baugher, D., Varanelli, A. & Weisbord, E. (2003). Student Hits in an Internet-Supported Course : How Can Instructors Use Them and What Do They Mean ? *Decision Sciences Journal of Innovative Education*, 1(2), 159–179.

Berkhin, P. (2006). A Survey of Clustering Data Mining Techniques. In Kogan, Jacob, Nicholas, Charles, & Teboulle, Marc (Eds.), *Grouping Multidimensional Data*, 25-71, Springer: New York.

Bousbia, N. et al. (2010). Analysing the Relationship between Learning Styles and Navigation Behaviour in Web-Based Educational System. *Knowledge Management & E-Learning: An International Journal*, 2(4), 400–421.

Clewley, N., Chen, S.Y. & Liu, X. (2011). Mining Learning Preferences in Web-based Instruction: Holists vs. Serialists. *Educational Technology & Society*, 14(4), 266–277.

Felder, R. & Silverman, L. K. (1988). Learning and Teaching Styles in Engineering Education" *Engineering Education*, 78(7), 674-681.

Felder, R. & Spurlin, J. (2005). Applications, Reliability, and Validity of the Index of Learning Styles. *International Journal of Engineering Education*, 21(1), 103-112.

Grabe, M., Christopherson, K. & Douglas, J. (2005). Providing Introductory Psychology Students Access to Online Lecture Notes: The Relationship of Note Use to Performance and Class Attendance. *Journal of Educational Technology Systems*, 33(3), 295–308.

Hung, J. & Zhang, K. (2008). Revealing Online Learning Behaviors and Activity Patterns and Making Predictions with Data Mining Techniques in Online Teaching. *MERLOT Journal of Online Learning and Teaching*, 4(4), 426–437.

Hung, J.-L. & Crooks, S.M. (2009). Examining Online Learning Patterns with Data Mining Techniques in Peer-Moderated and Teacher-Moderated Courses. *Journal of Educational Computing Research*, 40(2), 183–210.

Khan, F.A. et al. (2009). Integrated Approach for the Detection of Learning Styles & Affective States. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*. 753–761.

Kolb, David (1984). *Experiential learning: Experience as the source of learning and development.* Englewood Cliffs, NJ: Prentice-Hall.

Lu, H. et al. (2007). The Relationship of Kolb Learning Styles, Online Learning Behaviors and Learning Outcomes. *Educational Technology & Society*, 10(4), 187–196.

Romero, C. & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1), 135–146. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0957417406001266 [Accessed March 21, 2014].

Romero, C. et al. (2013). Web usage mining for predicting final marks of students that use Moodle courses. *Computer Applications in Engineering Education*, 21(1), 135–146. Available at: http://doi.wiley.com/10.1002/cae.20456 [Accessed March 21, 2014].

Smith, D. M., & Kolb, D. A. (1986). *Learning Style Inventory*: User's guide, Boston: McBer & Company.

Editor’s Note: *This paper was selected for inclusion in the journal as an ISECON 2014 Distinguished Paper. The acceptance rate is typically 7% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2014.*

Appendix: Tables and Figures

Table 1: Summary statistics of online content usage, learning styles and exam performance

	mean	sd	median	min	max	N
by student:						
time online during classtime (hours)	6.36	2.42	5.89	2.07	13.35	91
consistency: non-class days visiting content	8.02	2.58	8	1	15	91
time online in exam study period (hours)	0.28	0.45	0.00	0.00	2.06	91
time online working ahead (hours)	1.10	0.79	0.85	0.13	3.65	91
Active-Reflective (ILS)	-1.37	3.98	-1	-9	9	82
Sensing-Intuitive (ILS)	-2.35	5.64	-3	-11	11	82
Visual-Verbal (ILS)	-4.17	4.65	-5	-11	9	82
Sequential-Global (ILS)	-1.78	3.51	-3	-9	9	82
Abstract Conceptualization (Kolb)	31.00	5.73	32	15	44	82
Active Experimentation (Kolb)	31.85	8.40	33	16	48	82
Concrete Experience (Kolb)	28.77	7.89	29	13	46	82
Reflective Observation (Kolb)	28.28	6.94	27	14	48	82
Exam score (# correct out of 20)	16.18	3.40	17	5	20	91
by student-lesson:						
time viewing lesson content during classtime (hours)	0.32	0.30	0.26	0.00	2.04	819
consistency: non-class days visiting lesson content	1.04	0.71	1	0	4	819
time viewing lesson content in exam study period (hours)	0.02	0.06	0.00	0.00	0.54	819
time viewing lesson content working ahead (hours)	0.11	0.24	0.00	0.00	1.96	819
% correct of questions asked on lesson content	0.79	0.35	1	0	1	819
across individual pageviews:						
time on page (individual view; in seconds)	762.99	769.21	569	60	7190	2829

Notes: Table presents summary statistics of online content use, learning styles and exam performance at various levels of aggregation.

Table 2: Pairwise correlations of online content usage, learning styles and exam performance

	1	2	3	4	5	6	7	8	9	10	11	12	13
1. Exam score	1												
2. time online during classtime (hours)	0.26	1											
3. time online in exam study period (hours)	0.17	0.16	1										
4. consistency: non-class days visiting content	0	-0.3	0.2	1									
5. time online working ahead (hours)	0.05	0.01	0.09	0.21	1								
6. Active-Reflective (ILS)	-0.03	-0.05	-0.07	-0.07	-0.15	1							
7. Sensing-Intuitive (ILS)	-0.1	-0.12	-0.01	-0.03	0.23	-0.29	1						
8. Visual-Verbal (ILS)	0.06	0.05	0.21	0.15	0.02	0.14	-0.23	1					
9. Sequential-Global (ILS)	-0.03	-0.2	0.05	0.05	0.2	-0.01	0.21	-0.05	1				
10. Abstract Conceptualization (Kolb)	-0.05	-0.06	-0.04	-0.08	0.01	-0.06	-0.18	0.16	-0.16	1			
11. Active Experimentation (Kolb)	-0.15	0.06	-0.09	-0.12	-0.09	0.32	0.04	-0.01	-0.1	-0.14	1		
12. Concrete Experience (Kolb)	0.09	0.01	-0.01	0.06	-0.12	-0.04	-0.17	0.03	0.11	-0.39	-0.53	1	
13. Reflective Observation (Kolb)	0.14	-0.02	0.16	0.13	0.24	-0.28	0.3	-0.18	0.15	-0.22	-0.5	-0.17	1

Notes: Table presents pairwise Pearson correlation coefficients across combinations of student-level measures.

Table 3: Predicting online content usage with learning styles, OLS regression results, student-level variation

Dependent variable	**time online during classtime (hours)**	**time online in exam study period (hours)**	**consistency: non-class days visiting content**	**time online working ahead (hours)**
Panel A: Using ILS				
Active-Reflective (ILS)	-0.024	-0.011	-0.064	-0.022
	(0.04)	(0.01)	(0.07)	(0.02)
Sensing-Intuitive (ILS)	-0.022	0.0001	-0.015	0.026
	(0.03)	(0.01)	(0.05)	(0.02)
Visual-Verbal (ILS)	0.007	0.020*	0.084	0.014
	(0.03)	(0.01)	(0.06)	(0.02)
Sequential-Global (ILS)	-0.06	0.007	0.045	0.037
	(0.04)	(0.01)	(0.08)	(0.03)
Constant	1.621***	0.350***	8.541***	1.275***
	(0.21)	(0.07)	(0.44)	(0.14)
Observations	82	82	82	82
R2	0.052	0.057	0.036	0.094
Panel B: Using Kolb Learning Styles				
Abstract Conceptualization (Kolb)	-0.012	-0.001	-0.036	0.012
	(0.03)	(0.01)	(0.05)	(0.02)
Active Experimentation (Kolb)	0.006	-0.001	-0.028	0.007
	(0.02)	(0.01)	(0.04)	(0.01)
Reflective Observation (Kolb)	-0.003	0.009	0.021	0.035**
	(0.02)	(0.01)	(0.05)	(0.02)
Constant	2.059	0.084	9.646***	-0.484
	(1.63)	(0.55)	(3.25)	(1.03)
Observations	81	82	82	82
R2	0.006	0.026	0.026	0.068

Notes: Table presents coefficients from a linear regression estimating various aspects of online behavior. Standard errors in parentheses. Significance levels indicated by * .10, ** .05, ***.01.

Table 4: Predicting exam scores with online content usage, OLS regression results, student-level variation

	(1)	**(2)**	**(3)**	**(4)**
time online during classtime (hours)	0.720**	0.450**	0.616**	0.412**
	(0.31)	(0.18)	(0.31)	(0.18)
time online in exam study period (hours)	0.846	0.11	0.869	0.104
	(0.82)	(0.10)	(0.80)	(0.10)
consistency: non-class days visiting content	0.062	0.067	-0.051	-0.039
	(0.15)	(0.16)	(0.15)	(0.16)
time online working ahead (hours)	0.131	0.191	0.099	0.092
	(0.46)	(0.52)	(0.45)	(0.52)
Constant	14.021***	10.055**	15.252***	12.151***
	(1.47)	(4.48)	(1.53)	(4.55)
Observations	91	91	88	88
R2	0.089	0.089	0.082	0.087

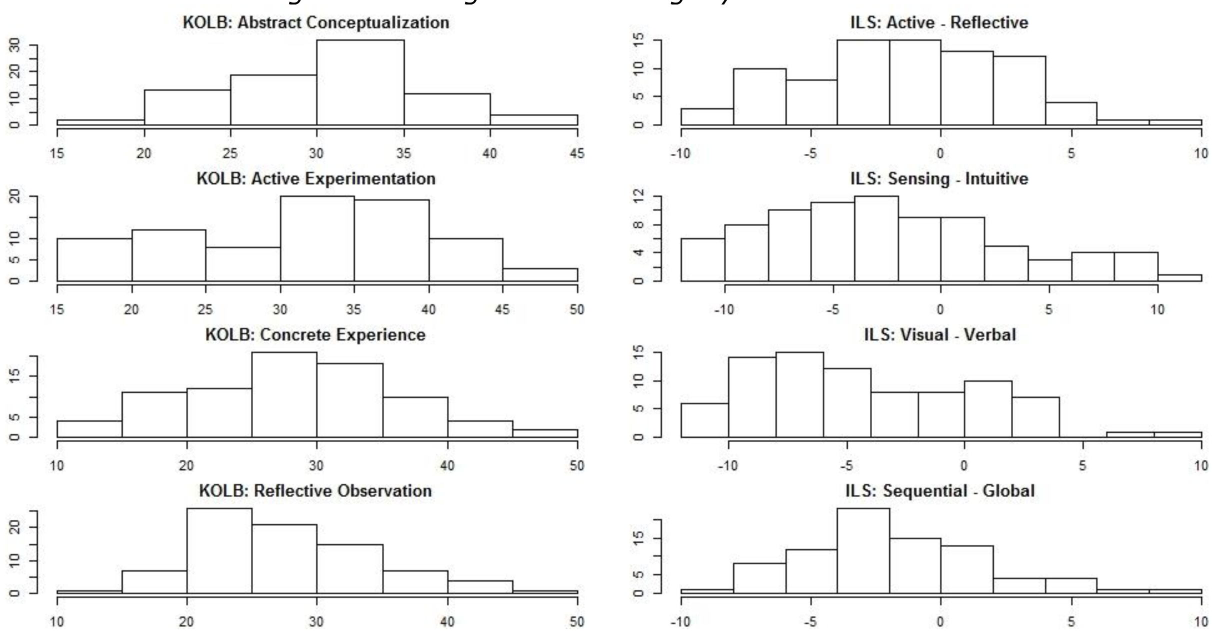
Notes: Dependent variable exam score. Table presents coefficients from a linear regression estimating exam scores with measures of online behavior. Columns 3 and 4 exclude a small number of students with outlier values in online use measures. Columns 2 and 4 estimate the specification using natural logs of the online time measures. Standard errors in parentheses. Significance levels indicated by * .10, ** .05, ***.01.

Table 5: Predicting exam scores with online content usage, OLS regression results, student-lesson variation

	(1)	(2)
time online during classtime (hours)	0.0004	0.0003
spent on lesson	(0.10)	(0.01)
time online in exam study period (hours)	0.004	0.007
spent on lesson	(0.25)	(0.01)
consistency: non-class days visiting lesson	0.073***	0.066**
	(0.02)	(0.03)
time online working ahead (hours)	-0.110**	-0.007*
spent on lesson	(0.05)	(0.00)
Student fixed effects	Y	Y
Observations	819	819
R2	0.886	0.886

Notes: Dependent variable: Percentage correct of questions pertaining to lesson. Table presents coefficients from a linear regression estimating question scores across lessons and students. Column 2 estimates the specification using natural logs of the online time measures. Standard errors in parentheses. Significance levels indicated by * .10, ** .05, ***.01.

Figure 1: Histogram of learning styles across students



Evaluating Business Intelligence / Business Analytics Software for Use in the Information Systems Curriculum

Gary Alan Davis
davis@rmu.edu

Charles R. Woratschek
woratschek@rmu.edu

Computer and Information Systems
Robert Morris University
Moon Township, PA 15108 USA

Abstract

Business Intelligence (BI) and Business Analytics (BA) Software has been included in many Information Systems (IS) curricula. This study surveyed current and past undergraduate and graduate students to evaluate various BI/BA tools. Specifically, this study compared several software tools from two of the major software providers in the BI/BA field. The participants in the study evaluated each software tool according to three key criteria: 1) functionality, 2) ease of use, and 3) learning effectiveness. The "learning effectiveness" criterion was used to determine which BI/BA tools provided the most effective learning of BI/BA concepts in the IS classroom. The three criteria were used to develop recommendations for including specific BI/BA software tools in the IS curriculum. Based on the findings of the study, the authors recommend that colleges and universities consider the use of the IBM-Cognos suite of tools as a viable means for teaching BI/BA concepts in their Information Systems curricula. The results of the study are relevant to any college or university that currently includes (or is considering the inclusion of) Business Intelligence / Business Analytics concepts in its Information Systems curriculum.

Keywords: Business Intelligence, Business Analytics, Software Evaluation, Information Systems Curriculum

1. INTRODUCTION

Business Intelligence (BI) and Business Analytics (BA) Software has been included in many Information Systems (IS) curricula (Davis, Woratschek, & Kohun, 2005; Olsen & Bryant, 2012). The authors of this study sought to determine which BI/BA software tools are the most effective in IS curricula. To determine software effectiveness, the authors surveyed current and past undergraduate and graduate students who are attending/have attended BI/BA-related courses. Specifically, the students who were surveyed were asked to compare several software tools from two of the major software providers in the BI/BA field. In order to determine a level of "effectiveness," the participants in the study evaluated each software tool according to three key criteria: 1) functionality, 2) ease of use, and 3) learning effectiveness. The "learning effectiveness" criterion was used to determine which BI/BA tools provided the most effective learning of BI/BA concepts in the IS classroom. The three criteria were used to develop recommendations for including specific BI/BA software tools in the IS curriculum. The results of this study are relevant to any college or university that currently includes (or considers including) Business Intelligence / Business Analytics concepts in its Information Systems curriculum.

2. RESEARCH QUESTIONS

The current study attempted to determine the effectiveness of Business Intelligence (BI) / Business Analytics (BA) software, in regard to classroom use, by answering the following research questions:

1. Which suite of BI/BA software tools (i.e., IBM-Cognos or Microsoft) was rated by survey participants as having greater functionality?

2. Which suite of BI/BA software tools (i.e., IBM-Cognos or Microsoft) was rated by survey participants as having greater ease of use?

3. Which suite of BI/BA software tools (i.e., IBM-Cognos or Microsoft) was rated by survey participants as providing greater learning effectiveness?

4. If there are noted differences between IBM-Cognos and Microsoft BI/BA tool suites, are the differences statistically significant?

3. BACKGROUND LITERATURE

The term Business Intelligence (BI) was originally coined by Richard Millar Devens in 1865. He used the term to describe how a banker profited by receiving and acting upon information about his environment before his competitors could (Devens, 1865). Collecting and acting upon information retrieved is still the basis of the definition of BI used today.

A little less than a century later, the term BI was used by IBM researcher, Hans Peter Luhn. Luhn used Webster's dictionary definition of intelligence: " . . . the ability to apprehend the interrelationships of presented facts in such a way as to guide action towards a desired goal" (Luhn, 1958, p.314).

Business Intelligence, as the term is used today, evolved from the decision support systems (DSS) that began in the 1960s and developed throughout the mid-1980s. Modern BI systems only became a reality in the 1990s with the advent of the data warehouse. Many authors assert that Modern BI is not a technology. Rather, it is described as a process of generating information from raw data by using a combination of hardware, architectures, tools, methods, and databases (Turban, Sharda, Delen, & King, 2011).

A review of the literature finds that many colleges/universities do not offer a degree in BI. A 2010 survey was conducted by the BI Congress to determine the state of BI in academia. This Congress is the work of the Teradata University Network (TUN) and the Special Interest Group on Decision Support, Knowledge and Data Management Systems (SIGDSS). Approximately 130 colleges/ universities were represented in this survey and 173 professors responded. Only three schools reported having an undergraduate degree in BI: Augusta State University, St. Joseph's University, and Stuttgart Media University (Germany).

Twelve schools reported having a graduate degree in BI: Augusta State University, University of Denver, St. Joseph's University, Stuttgart Media University (Germany), Sofia University (Bulgaria), North Carolina State University, Singapore Management University (Singapore), Texas Tech University, Loyola University Chicago, Xavier University, University of Muenster (Germany), and Universidade Portucalense (Portugal) (Wixom & Ariyachandra, 2011).

For those colleges/universities teaching BI courses, 34% indicated that having access to BI software was one of the challenges in teaching BI (Wixom, B. H. and T. Ariyachandra, 2011). Academic partnerships were used to access BI software/resources, specifically: Teradata University Network (48%) Microsoft Educational Consortium (46%), IBM Academic Alliance (28%), and Oracle Academy (14%) (Wixom & Ariyachandra, 2011).

In 2012, the BI Congress once again surveyed colleges/universities to determine the state of BI in academia. Forty-three countries and 319 professors were represented in the survey. The United States had the most respondents at 206

(66.5%). Germany came in second with 19 respondents (6.1%). Only 26% of the respondents stated that one of the challenges in teaching BI was having "access to contemporary, enterprise software" (Wixom, Ariyachandra, & Mooney, 2013). The academic partnerships used to access BI software/resources were Microsoft Educational Consortium (46%), Teradata University Network (30%), IBM Academic Alliance (28%), and Oracle Academy (12%) (Wixom, Ariyachandra, & Mooney, 2013).

Robert Morris University acquired a license for the academic use of Cognos' OLAP tool (i.e., PowerPlay) in 2003. However, that license expired and the software became unaffordable. Finding affordable BI software/resources for academic use was challenging. The solution was to join the Microsoft Educational Consortium, based out of the University of Arkansas, and use Microsoft's BI tools. The BI courses at Robert Morris University have used Microsoft's BI tools for the past three years.

The large scale software vendors such as IBM, Oracle, SAP, Teradata, and Microsoft all boast of a BI solution to business problems. However, questions arise as to the affordability and vendor support of each of these solutions for collegiate classroom use. As equally important, what are the advantages/ disadvantages, from the standpoint of student learning outcomes, in using one vendor solution over another?

4. METHODS AND PROCEDURES

Approach and Sample

This study involved the administration of a survey to current and past students who are currently taking (or have taken) Business Intelligence (BI) / Business Analytics (BA)-related courses. The survey population was obtained from student rosters of both undergraduate-level and graduate-level courses. The survey participants attended (or are currently attending) Robert Morris University, a private, medium-sized, Mid-Atlantic school. *QuestionPro Online Survey Software*, from QuestionPro, Inc. was used to administer the survey, collect results, and analyze results. All survey participants were over the age of 18, and participation in the study was completely voluntary. In addition, all survey responses were captured and stored anonymously (i.e., no personally-identifying information was solicited nor captured from the survey participants).

The *QuestionPro* online survey link was sent (via electronic mail) to 325 current and past students. The survey link was active and available from April 1, 2014 until April 30, 2014. During the 30-day period that the survey link was available, 46 respondents completed the survey and submitted their responses for analysis. The completion rate for the online survey was just over 14%.

Survey Instrument

The survey instrument consisted of a total of 27 questions; 25 of the questions were closed-ended, and two of the questions were open-ended. The survey asked the participants to provide ratings for BI/BA software tools sold by IBM-Cognos and by Microsoft. These two software tools were chosen because of their use in Robert Morris University's BI courses, past or present. Questions one through nine asked participants to rate various BI/BA software tools provided by IBM-Cognos (i.e., Data Manager, Transformer, Analysis Studio, and Report Studio). Question 10 asked participants to give an *overall* rating to the *suite* of BI/BA tools sold by IBM-Cognos. Questions 11 through 19 asked participants to rate various BI/BA software tools provided by Microsoft (i.e., Integration Services, Analysis Services, Excel, and Reporting Services). Question 20 asked participants to give an *overall* rating to the *suite* of BI/BA tools sold by Microsoft. In all of the questions that solicited a rating, participants were asked to rate the tools according to *functionality*, *ease of use*, and *learning effectiveness*. For each of the aforementioned criterion, participants were asked to provide a rating of (1) Poor, (2) Average, (3) Good, or (4) Excellent. The four-point rating system was used in the survey in order to require a "forced-choice" from the participants and, therefore, avoid "central tendency" bias.

In addition to the questions that solicited a rating, the survey also contained several demographic questions. The demographic questions asked participants to indicate their degree (i.e., either earned or in-progress), their sex, whether or not they are currently working in BI/BA, and (if "yes") what BI/BA tool(s) they currently use in their job.

Toward the end of the survey, participants were asked what they felt would be the next "hot topics" in the field of BI/BA. At the very end of the survey, participants were asked to list the

BI/BA topics that they feel should be included in Information Systems curricula.

Once collected, all survey results were analyzed using SPSS (Statistical Package for the Social Sciences). Descriptive statistics were generated in SPSS to calculate the participants' mean rating scores, as related to software functionality, ease of use, and learning effectiveness. In addition, the *Independent Samples T-Test* was used to determine if any noted differences in mean rating scores between the two software vendors were statistically significant.

5. RESULTS

Functionality

To address the first research question (which Business Intelligence/Business Analytics tool suite was rated by participants as having greater functionality?), the survey contained questions that asked participants to compare the functionality of IBM-Cognos BI /BA tools with the functionality of Microsoft BI/BA tools. The functionality was categorized by ETL (Extract, Transform, and Load) functionality, OLAP (On-line Analytical Processing) functionality, and Reporting/BPM (Business Performance Management) functionality. Overall, the survey participants rated the functionality of IBM-Cognos BI/BA tools as being greater ($\bar{x}$ = 3.12) than the Microsoft BI/BA tools ($\bar{x}$ = 2.90).

The Independent Samples T-Test was used to determine whether or not the difference in functionality was statistically significant. Although the IBM-Cognos tools were rated as having greater functionality than the Microsoft tools, the difference in means was not statistically significant at the .05 level (t = 1.013, p = .316). The results from the responses regarding functionality are summarized in *Appendix A - Table 1: Independent Samples T-Test for BI/BA Functionality.*

Ease of Use

To address the second research question (which BI/BA tool suite was rated by participants as having greater usability?), the survey contained questions that asked participants to compare the usability of IBM-Cognos BI /BA tools with the usability of Microsoft BI/BA tools. In a similar manner as functionality, usability was again categorized according to the usability of ETL tools, OLAP tools, and Reporting/BPM tools. Overall, the survey participants rated the usability of IBM-Cognos BI/BA tools as being greater ($\bar{x}$ = 3.00) than that of Microsoft BI/BA tools ($\bar{x}$ = 2.62).

The Independent Samples T-Test was used to determine whether or not the difference in usability was statistically significant. The IBM-Cognos tools were rated as having greater usability than the Microsoft tools; however, the difference in means was not statistically significant at the .05 level (t = 1.653, p = .105). The results from the responses regarding usability are summarized in *Appendix A - Table 2: Independent Samples T-Test for Ease of Use.*

Learning Effectiveness

To address the third research question (which BI/BA tool suite was rated by participants as having greater learning effectiveness?), the survey contained questions that asked participants to compare the learning effectiveness of IBM-Cognos BI /BA tools with the learning effectiveness of Microsoft BI/BA tools. As with the prior criteria, learning effectiveness was categorized according to the learning effectiveness of ETL tools, OLAP tools, and Reporting/BPM tools. Once again, the survey participants rated the IBM-Cognos tools higher than Microsoft. Overall, the participants rated the learning effectiveness of IBM-Cognos BI/BA tools as being greater ($\bar{x}$ = 3.18) than the Microsoft BI/BA tools ($\bar{x}$ = 2.86).

The Independent Samples T-Test was used to determine whether or not the difference in learning effectiveness was statistically significant. The IBM-Cognos tools were also rated as having greater learning effectiveness than the Microsoft tools; however, the difference in means was not statistically significant at the .05 level (t = 1.711, p = .094). The results from the responses regarding usability are summarized in *Appendix A - Table 2: Independent Samples T-Test for Learning Effectiveness.*

T-Test for Statistical Significance

As discussed in the **METHODS AND PROCEDURES** section, the Independent Samples T-Test was used to detect statistical significance in any noted difference in survey results between the two vendors. None of the criteria tested (i.e., functionality, ease of use, nor learning effectiveness) differed between the two software vendors in a statistically significant way. Out of all three criteria tested, the learning effectiveness criterion *came closest* to a statistically significant difference. However, as discussed above, the difference in means between IBM-Cognos BI/BA tools and Microsoft BI/BA tools for learning effectiveness was not statistically significant at the .05 threshold (t = 1.711, p = .094).

6. CONCLUSIONS

This research surveyed current and past undergraduate and graduate students to answer the following questions regarding IBM-Cognos and Microsoft BI/BA (Business Intelligence / Business Analytics) software tools: 1) Which suite of software tools was rated by survey participants as having greater functionality?, 2) Which suite of software tools was rated by survey participants as having greater ease of use?, 3) Which suite of software tools was rated by survey participants as providing greater learning effectiveness?, and 4) Are any noted differences in participant ratings between the two software vendors statistically significant?

As discussed in the **RESULTS** section, the IBM-Cognos tools were rated higher than Microsoft tools by survey participants in all three categories: 1) functionality, 2) ease of use, and 3) learning effectiveness. However, *none* of the differences were statistically significant at the .05 level. Despite the lack of statistical significance, these results seem to indicate that, when compared to Microsoft, the IBM-Cognos BI/BA suite of tools offer greater functionality for performing BI/BA tasks, and greater ease of use for the end user. These results also seem to indicate that the IBM-Cognos tools provide more effective learning of BI/BA concepts (when used in the classroom) than the Microsoft toolset.

The above findings are interesting when viewed in the context of responses received from other survey questions. For example, more survey participants (61%) reported as having used the Microsoft BI/BA toolset than the IBM-Cognos toolset (39%). This pattern of software use is not surprising, given that Robert Morris University has been using the Microsoft BI/BA toolset since 2011. In other words, it is expected that current and past students of the University would have had exposure to the Microsoft toolset, as part of their BI/BA classes.

The findings, regarding the use of BI/BA software in academia, are also consistent with the aforementioned findings by the BI Congress. As discussed previously, a 2010 survey by the BI Congress found that 46% of schools surveyed utilize the Microsoft Educational Consortium to provide BI/BA software to students. In the 2010 survey, the percentage of schools utilizing the Microsoft Educational Consortium was significantly higher than the percentage of schools utilizing the IBM Academic Alliance (28%) (Wixom & Ariyachandra, 2011).

However, when the current study asked which suite of tools was used in the workplace (by participants who currently work in the BI/BA field), IBM-Cognos was reported as the tool of choice for 21% of the participants’ employers. The Microsoft BI/BA toolset, however, was reported as being used by only 13% of employers. It is also interesting to note that Oracle BI/BA tools were also reported as being used by 21% of employers. The other major BI/BA software vendors used by participants' employers included SAP-Business Objects (13%), and Informatica (15%). Oracle, SAP-Business Objects, and Informatica software tools were *not* rated by participants in the current study. The remaining 17% of employers in question were either represented by the category "Other" (i.e., we use a BI/BA tool that was not listed on the survey) or by "My organization does not use BI/BA software tools."

Recommendations

Based on the findings from the current research, colleges and universities should consider the use of the IBM-Cognos suite of tools as a viable means to for teaching BI/BA concepts in their Information Systems curricula. However, these findings, and the resulting recommendations, are contrary to the reported market shares of leading BI/BA software suites. In North America, the top three BI/BA vendors, in terms of 2013 market share, were Microsoft (43%), Oracle (30%), and SAP-Business Objects (28%) (Henschen, 2014). In terms of 2013 *worldwide* market share, the top three BI/BA vendors were SAP-Business Objects (21%), Oracle (14%), and IBM-Cognos (13%) (Columbus, 2013).

The recommendations from the current study also seem to conflict with the current use of academic partnerships by colleges and universities. As discussed previously, the most prevalent academic partnerships leveraged by colleges and universities are Microsoft Educational Consortium (46%), Teradata University Network (30%), IBM Academic Alliance (28%), and Oracle Academy (12%) (Wixom, Ariyachandra, & Mooney, 2013).

Future Research

Even though the IBM-Cognos suite of BI/BA tools received the highest rating in all three question categories, the sample size of the current study was quite limited (n = 46). Future research could solicit responses from a larger sample, both in terms of number of participants and in geographical area. Participant ratings for additional BI/BA software vendors (e.g., Oracle, SAP-Business Objects, Teradata, Informatica, et al.) might also be requested. Finally, the current research focused on *suites* of BI/BA tools. Future research could solicit and analyze participant ratings for suites of tools, as well as *individual* types of tools, such as ETL, OLAP, Reporting, and BPM.

7. REFERENCES

Borkovich, D. J., Noah, P. (2014). Big data in the information age: exploring the intellectual foundation of communication theory. *Information Systems Education Journal*, 12(1) pp. 15-26.

Columbus, L. (2014). 2013 Business intelligence and analytics market share update: SAP continues market leadership. Retrieved June 17, 2014 from http://www.forbes.com/sites/ louiscolumbus/2014/04/29/2013-business-intelligence-and-analytics-market-share-update-sap-continues-market-leadership.

Davis, G. A., Woratschek, C. R., & Kohun, F. G. (2005). Curriculum development: Developing a graduate degree program in Competitive Intelligence. *Issues in Information Systems Journal*, 6(1), 318-324.

Devens, R. M. (1865). Cyclopaedia of commercial and business anecdotes; comprising interesting reminiscences and facts, remarkable traits and humors of merchants, traders, bankers, etc. in all ages and countries, p. 210. Retrieved 11 June 2014 from D. Appleton and company.

Halevi, G., Moed, H. (2012). The evolution of big data as a research and scientific topic. *Research Trends*, Issue 30, September 2012.

Henschen, D. (2014). 2014 BI outlook: who's hot, who's not. *InformationWeek*. Retrieved 6/17/2014 from http://www.informationweek.com/software/information-management/ 2014-bi-outlook-whos-hot-whos-not/d/d-id/1113002.

Luhn, H.P. (1958). A business intelligence system. *IBM Journal* 2(4): 314. doi:10.1147/rd.24.0314.

Olsen, D. H., & Bryant, P. (2012). Business intelligence and information systems: enhancing student knowledge in database courses. *The Review of Business Information Systems* (Online), 16(1), 1.

Sharda, R., Delen, D., & Turban, E. (2014). *Business intelligence: a managerial perspective on analytics–3rd edition*. Saddle River, NJ: Pearson-Prentice Hall.

Turban, E., Sharda, R., Delen, D., & King, D. (2011). *Business intelligence: a managerial approach – 2nd edition.* Saddle River, NJ: Pearson-Prentice Hall.

Wixom, B. H. & T Arivachandra (2011). State of business intelligence in academic 2010. BI congress 2, Retrieved June 14, 2014 from http://www2.cmmerce.virginia.edu/bic3/ 2010-State-of-BI-Report.pdf

Wixom, B. H. & T Arivachandra and J. Mooney (2013). State of business intelligence in academia, BI congress 3. Retrieved June 14, 2014 from http://www2.commerce.virginia.edu/bic3/ content/2012-State-of-BI-Report.pdf.

APPENDIX A – T-TEST RESULTS

Table 1: Independent Samples T-Test Results
Independent Samples T-Test Results for BI/BA Functionality

	Mean	Std. Dev.	t-test	df	Sig.
IBM-Cognos Suite of BI/BA Software	3.12	.781	1.013	44	.316
Microsoft Suite of BI/BA Software	2.90	.673			

Table 2: Independent Samples T-Test Results
Independent Samples T-Test Results for Ease of Use

	Mean	Std. Dev.	t-test	df	Sig.
IBM-Cognos Suite of BI/BA Software	3.00	.707	1.653	44	.105
Microsoft Suite of BI/BA Software	2.62	.775			

Table 3: Independent Samples T-Test Results
Independent Samples T-Test Results for Learning Effectiveness

	Mean	Std. Dev.	t-test	df	Sig.
IBM-Cognos Suite of BI/BA Software	3.18	.636	1.711	44	.094
Microsoft Suite of BI/BA Software	2.86	.581			

IT0: Discrete Math and Programming Logic Topics as a Hybrid Alternative to CS0

Nancy L. Martin
nlmartin@siu.edu
Information Systems Technologies
Southern Illinois University
Carbondale, IL 62901, U.S.A.

Abstract

This paper describes the development of a hybrid introductory course for students in their first or second year of an information systems technologies degree program at a large Midwestern university. The course combines topics from discrete mathematics and programming logic and design, a unique twist on most introductory courses. The objective of the new course is to better prepare students for more advanced computing courses. Two primary drivers motivated development of the new course: 1) faculty evidence of deficient foundation skills in advanced level courses, and 2) consideration of program accreditation criteria.

Keywords: Course Design, Introductory Course, Programming, Discrete Mathematics, IT2008 Model Curriculum, Course Development

1. PROGRAM BACKGROUND

Since its inception more than twenty years ago, the Information Systems Technologies (IST) degree program has continually evolved to meet the changing needs of its stakeholders. The IST program is housed in the College of Applied Sciences and Arts, reflecting its focus on applied, hands-on skills in the field of information technology (IT). Since the late 1990's, the IST major has progressed from an office systems degree that was based on the Organizational and End-user Information Systems (OEIS) curricula guide (The Organizational Systems Research Association, 2004) to today's program with courses in programming, networking, databases, web systems development, and other core topics. Concentrations are currently available in two tracks: network and information security and web systems development.

As the IST major grew from the OEIS model, more and more technical courses were added to the curriculum. For example, several information assurance-focused courses are now part of the Network and Information Security Track, and the program earned designation as a National Center of Academic Excellence in Information Assurance Education in 2011. In recent years, faculty have implemented numerous new elective courses and revised courses in the core and tracks to keep the curriculum current with the needs of the program stakeholders and the job market. Also, as the program has progressed, the technical components of the courses have become more rigorous.

The IST program consists of 16 core courses plus elective courses, some of which are offered in specialized tracks. Students may complete a track or combine a variety of elective courses to create a more personalized program. University core curriculum requirements comprise 41 credit hours; the IST core comprises another 49 credit hours, and the remaining 30 hours are open to IST electives. All courses are one-semester, three credit hours except the required internship, which is four credit hours. For a mathematical foundation, IST students have been required to take an applied statistics course and only the minimal mathematics required in the university core curriculum. Students are encouraged to take a philosophy-based logic course as part of the university core curriculum, but it is not required. The IST core courses, electives, and tracks are listed in Appendix A.

Student Demographics
In recent years, average enrollment for the IST program has been 215 students, and the program confers an average of 73 degrees annually. Most students are male, with only about 15 percent female enrollment. About five percent of students enroll directly into the program as freshman. The majority of students transfer from pre-major or other majors within the university, and roughly 11 percent transfer from community colleges or other universities. Most within-university transfers come from computer science and computer engineering. The IST program does not require calculus or other upper level mathematics courses which draws many students to the program from computer science and engineering. The retention rate in IST is one of the highest in the university, averaging over 90 percent.

IST graduates are recruited by a number of international, national, and regional companies. Examples of major employers include a large aircraft company, a national insurance company, an international information security company, and a national healthcare software vendor. A recent survey of graduates found that about 32 percent were employed or had job offers prior to graduation and another 53 percent were employed within six months of graduation (Legier, Woodward, & Martin, 2013).

2. MOTIVATION

Two primary factors motivated the development of a new foundation course in the IST major. First, anecdotal evidence from faculty revealed that some students in higher-level courses struggle with concepts normally covered in prerequisite courses such as discrete mathematics and programming logic. Second, faculty began considering the feasibility of seeking accreditation for the IST program.

Student Performance
Curriculum enhancements over the past several years have created a stronger, more current program with courses such as advanced web systems development, software engineering, and advanced enterprise networking. As more advanced courses were developed, lower level courses were also updated to better equip students both for later courses and also for the job market in general. For example, systems analysis and design was previously taught as a two-course sequence with the first course offered in the sophomore year and the second course offered in the senior year. Those two courses have now been combined into one upper level course. Another example is that client side web technologies was taught as a separate course, but that content has now been "pushed" down into the introductory web applications course. Additionally, students were only required to take one Java-based programming course, and relatively few took the Programming II course. Coverage in the first course was restricted to basic programming concepts and initial coverage of arrays, which is considered limited programming knowledge in the IST curriculum.

Over time it became clear to faculty of upper level programming, web development, and network and security courses that some students lacked foundation skills needed to be successful in those advanced topics. For example, some students found their initial real application of binary and hexadecimal number systems in the first information assurance course. The instructor found it necessary to spend valuable course time reviewing those topics; similarly, faculty in advanced web systems and software engineering courses spent too much reviewing basic programming concepts.

Curriculum Review for Accreditation
Curriculum review is an ongoing process for IST faculty. However, in spring of 2013, IST faculty undertook a comprehensive review of the curriculum in consideration of pursuing program accreditation through the Accreditation Board for Engineering and Technology (ABET). The first step was to determine where the IST program best fit within ABET computing programs. The IST program was obviously not a computer science program; so, only information systems and information technology programs were considered. Also, as part of the review process, model curricula were considered for information technology (IT2008) (Lunt et al., 2008) and information systems (IS2010) (Topi et al., 2010).

Careful comparison of IST program objectives and desired student outcomes with ABET accreditation criteria revealed a good alignment with ABET's information technology program (ABET, 2012); similarly, the IT2008 curriculum model philosophy, body of knowledge, and learning outcomes most closely fit with the existing IST program. Although the entire review is outside the scope of this paper, two areas were important in the creation of a new foundation course: programming and mathematics.

3. COMPUTING CURRICULA

There are a variety of computing degrees available to today's college students. The Association of Computing Machinery defines five distinct computing curricula (CC2005): computer engineering (CE), computer science (CS), information systems (IS), information technology (IT), and software engineering (SE) (ACM/AIS/IEEE-CS Joint Task Force for Computing Curricula, 2005). The difference among the five is a varying emphasis on computing knowledge areas, goals, and capabilities of graduates. It is also important to note that there are options other than the five distinct areas since some programs blur the lines of distinction between the ACM curricula (e.g., Connolly & Paterson, 2011). Regardless of the curricula followed, one commonality is that some degree of programming knowledge and some level of mathematics are recommended.

CC2005, for the first time, defined IT separately from other computing degrees. Soon after, the first model curriculum, IT2008, was released and provided this definition: "IT, as an academic discipline, is concerned with issues related to advocating for users and meeting their needs within an organizational and societal context through the selection, creation, application, integration and administration of computing technologies" (Lunt, et al., 2008, p. 9). IT degree programs had arisen from an industry need for "professionals to select, create, apply, integrate, and administer an organizational IT infrastructure" (Lunt, Ekstrom, Reichgelt, Bailey, & LeBlanc, 2010, p. 133) and that need was not being met by computer science or information systems programs. The IST program evolved in the same way and in parallel to the IT academic discipline.

Programming in the IT2008 Curriculum

IT2008 defines 13 knowledge areas which are subdivided into units and topics within units. The Programming Fundamentals knowledge area comprises five units with the recommended minimum coverage hours displayed in Table 1. The recommended coverage totals 38 hours, which, in a three-credit hour course environment, could be delivered in a one-semester course. This approach is the most common in computing programs. IT2008 points out "that the number of core hours prescribed for this knowledge area is dependent on some previous programming experience" (Lunt, et al., 2008, p. 103).

Unit	Recommended Min Hours
Fundamental Data Structures	10
Fundamental Programming Constructs	10
Object-Oriented Programming	9
Algorithms and Problem-Solving	6
Event-Driven Programming	3

Table 1. IT2008 Programming Fundamentals Units

The introductory programming course, often called CS1, is the cornerstone of any computing curricula. However, over the past decade, universities have become increasingly concerned about declining enrollments and retention in computing programs. Eager to find solutions to those problems, the CS1 course has been an obvious place to focus efforts. Many students come into a computing major with little or no previous programming experience, a category of learners dubbed "novice programmers". If student performance in CS1 can be improved, future success in a computing major is more likely.

However, learning to program is notoriously difficult for novice programmers, and as a result, much attention and research has focused on this perennial problem. (See Robins, Rountree, & Rountree, 2003). Many have studied the characteristics, habits, and success factors of novices in the CS1 course (e.g., Bennedsen & Caspersen, 2005a; Porter, Guzdial, McDowell, & Simon, 2013; Rountree, Rountree, & Robins, 2002). Others have focused on specific methods or approaches to improve performance in the CS1 course (e.g., Benander & Benander, 2008; Bennedsen & Caspersen, 2005b; Gill & Holton, 2006; Pears et al., 2007; Vihavainen, Paksula, & Luukkainen, 2011; Williams, Wiebe, Yang, Ferzli,

& Miller, 2002; Zhang, Zhang, Stafford, & Zhang, 2013).

The CS0 Course

One approach to improve CS1 performance is to require CS0, a "preprogramming" course. In some computing curricula, the CS1 course is preceded by CS0 or some other form of introductory course, and these prerequisite courses can improve performance in CS1 (Brown, 2013; Chor & Hod, 2012; Dierbach, Taylor, Zhou, & Zimand, 2005; Faux, 2006).

CS0 was first introduced as an orientation to the computer science major (Cook, 1997), and was implemented similarly to orientation courses in other disciplines. In addition to basic computing skills, the course included topics on time management, problem solving, professionalism, and career exploration. Over time, CS0 course designers embraced a variety of approaches and topics. Two common formats are breadth-first and depth-first. In a breadth-first CS0 course, exposure to programming language is limited to basic concepts. Topics may include those similar to (Cook, 1997), or focus on "authentic" everyday computing tasks to help students more easily comprehend computer science concepts (McFall & DeJongh, 2011). A depth-first approach usually depends on a specific programming language to develop problem-solving skills (Tucker & Garnick, 1991). More recently, CS0 courses are being implemented using a high level language in an attempt to attract students to a major in computing or to improve retention rates for at-risk students (e.g., Rizvi & Humphries, 2012; Uludag, Karakus, & Turner, 2011).

Another common form of CS0 is a programming logic course; however, some have found that particular type of course did not improve students' performance in advanced programming (Hoskey & Murino, 2011). Others have developed the CS0 course to address specific deficiencies. For example, one CS0 course focuses on mental models and concepts of programming (Dierbach, et al., 2005). Another found that emphasis on problem solving techniques and algorithm development prior to programming is beneficial (Faux, 2006). Others have also reported success with a CS0 course focused on problem solving skills (e.g., Cortina, 2007; Middleton, 2012; Mitchell, 2001; Van Dyne & Braun, 2014).

Regardless of the course arrangement or focus, the common objective of a CS0 course remains: to improve success in subsequent programming and computing courses. With the same objective in mind and recalling that the IT2008 coverage recommendations for programming assume some sort of previous exposure or experience, the IST faculty began planning the development of an introductory course, IT0.

Mathematics in the IT2008 Curriculum

IT2008 also addresses fundamental IT knowledge areas including mathematical foundations. The Math and Statistics for IT knowledge area comprises seven units with the specific recommended minimum coverage displayed in Table 2. The total recommended coverage is 38 hours.

Unit	Recommended Min Hours
Basic Logic	10
Discrete Probability	6
Functions, Relations and Sets	6
Hypothesis Testing	5
Sampling and Descriptive Statistics	5
Graphs and Trees	4
Application of Math & Statistics to IT	2

Table 2. IT2008 Math and Statistics for IT Units

The emphasis in IT2008 is on topics in statistics and discrete mathematics with a notable absence of calculus. Rigorous math requirements such as calculus have likely scared more than a few students away from computing majors, especially since some students have difficulty understanding how abstract mathematical concepts relate to the real world.

In research, the relationship between students' math background and success in computing courses has been a topic of interest for decades. How students perform in mathematics courses can sometimes predict success or failure in programming courses or in the entire computing curriculum (e.g., Campbell & McCabe, 1984; Capstick, Gordon, & Salvadori, 1975; Konvalina, Wileman, & Stephens, 1983; White & Sivitanides, 2003; Wilson & Shrock, 2001). Studies have usually focused on the number or type of mathematics courses taken or on scores on standardized tests. Regardless of findings, educators overwhelmingly agree that skills in mathematical thinking and reasoning transfer to success in working with abstract concepts and symbol manipulation in programming (e.g., Bruce, Scot, Kelemen, & Tucker, 2003; Henderson, 2005; Kelemen, Tucker, Henderson, Astrachan, & Bruce, 2000; Ralston, 2005).

There is also an ongoing debate as to exactly what kind of mathematics is really needed in various computing programs (e.g., Bruce, et al., 2003; Glass, 2000). However, a common theme throughout the literature is that of all mathematics courses taken, discrete mathematics may be the most important predictor of success in computing (Pioro, 2006; Sidbury, 1986). Moreover, a further dissection of the "how much math" debate reveals that most educators agree that topics in discrete mathematics are the most relevant for computing professionals. This sentiment toward coverage of topics in discrete mathematics is evident in the IT2008 model curriculum.

The Discrete Mathematics Course

Discrete mathematics is commonly taught in a one or two semester sequence in computing programs. Topics covered in these courses include logic, sets, functions, relations, counting, proofs, probability, and trees and graphs, among others. As with other mathematics courses, students do not necessarily recognize how discrete mathematics applies to their profession or to their future studies (Remshagen, 2010).

While the approaches to teaching discrete mathematics are not as varied or numerous as those for CS0, some universities have taken an integrative approach, incorporating discrete mathematics topics into core curriculum (Harvey, Wu, Turchek, & Longenecker, 2007), or into other courses such as data structures or formal methods, or simply focusing on making the topics more relevant to students. (Gegg-Harrison, 2005; Remshagen, 2010). Others have argued for combining discrete mathematics and functional programming into one course (VanDrunen, 2011).

Seeing the need for relevance in information systems programs, an interdisciplinary committee of faculty at one university developed a unique discrete mathematics course. The course was designed to relate real world uses and examples to selected discrete mathematics topics, all while "making learning easier and enjoyable" and increasing student confidence (Wood, Harvey, & Kohun, 2005, p. 387). The team developed customized course materials and have found their approach valuable in the information systems curriculum.

4. THE IT0 COURSE

Motivated by the need to better equip IST students for advanced coursework and the possibility of seeking ABET accreditation, the concept for a new IT0 course was formed. While the programming topics recommended in IT2008 were already being covered, the depth of coverage and the assumption of prior experience needed to be addressed. Additionally, within the mathematics knowledge area, some topics were covered in the applied statistics course; however other topics were only being haphazardly addressed elsewhere in the curriculum.

Overall, IST faculty felt that combining portions of both the Math and Statistics and the Programming Fundamentals knowledge areas from IT2008 would create a well-rounded preparatory course for the IST program and would greatly benefit IST students. However, there was not room in the curriculum to incorporate an additional mathematics course and a CS0-type course. Moreover, an entire semester of either course was not deemed necessary for the IST program. An opportunity presented itself when the former two course systems analysis and design sequence was compressed into one course. That change freed up a sophomore level course which would be used to create the hybrid discrete mathematics and programming logic and design course, IT0.

Course Content

With a one-course equivalent available in the curriculum, faculty began researching options or models for a combined discrete mathematics topics/programming logic course. Unfortunately, none were found, and it became clear the course would need to be developed from scratch. Since the IST program is most closely aligned with IT2008, faculty turned to those requirements for guidance in creating the new IT0 hybrid course.

The IT2008 Programming Fundamentals knowledge area units and recommended coverage hours were shown in Table 1. All programming units were previously covered in the introductory programming course, however some were covered only at a shallow level due to the time constraint of a one-semester course. With the new IT0 course, most units move to the new hybrid course and object-oriented programming will be introduced only in the context of simple problem solving exercises. Coverage for the new course is shown in Table 3.

Unit	Hours	Covered in Intro. Programming Course	Covered in New IT0 Course
Fundamental Data Structures	10	X	
Fundamental Programming Constructs	10	X	X
Object-Oriented Programming	9	X	
Algorithms and Problem-Solving	6	X	X
Event-Driven Programming	3	X	

Table 3. Programming Fundamentals Unit Coverage

IT2008 also provides specific topics and learning outcomes for each knowledge area unit. This detail-enabled faculty to clearly define learning objectives for the new course and ensure it met the prerequisite needs of more advanced courses in the IST program. The Programming Fundamentals units and specific topics with the associated learning outcomes as described in IT2008 are provided in Appendix B.

Unit	Hours	Covered in Statistics Course	Covered in New IT0 Course
Basic Logic	10		X
Discrete Probability	6	X	
Functions, Relations and Sets	6		X
Hypothesis Testing	5	X	
Sampling and Descriptive Statistics	5	X	
Graphs and Trees	4		X
Application of Math & Statistics to IT	2	X	X

Table 4: Math and Statistics for IT Unit Coverage

The Math and Statistics knowledge area units and hours were outlined in Table 2. Table 4 displays which of those units are currently covered in the applied statistics course for IST. Upon review, the topics of Basic Logic, Functions, Relations and Sets, and Graphs and Trees were nearly exact matches to the topic list IST faculty had devised as being areas of deficiency. These units will be addressed in the new IT0 course. Specific topics and learning outcomes for the discrete mathematics portion of the new course were also taken from IT2008 and are available in Appendix C.

The master course syllabus identifies the amount of time to be dedicated to each topic area and is included as Appendix D.

Course Materials

Since introductory courses in the IST program are sometimes taught by term instructors, the program requires the use of textbooks and other materials that are listed in the master syllabus for a course as a means to insure consistency. To assure the new IT0 course meets the stated objectives, the next task was to identify standard course materials. Needless to say, one textbook that covered all the IT0 topics did not exist.

Selecting a resource to cover the programming fundamentals portion of the course was fairly easy; a book by the same publisher and author as is used in the introductory programming course was selected. The similarity in the language and approach of the texts would afford a smoother transition to more advanced programming concepts for IST students.

Finding an appropriate resource for the mathematics portion of the class proved to be more difficult. Dozens of books and numerous web resources in the areas of discrete mathematics and introduction to computer science were reviewed. In each case, the text contained a great deal of information that would not be covered and thus not warrant requiring students to purchase a second book for the IT0 course. Faculty then began searching for freely available sources including entire books or individual modules in an effort to provide future instructors with a complete set of course materials. As of this writing, material for the mathematics portion of the IT0 course will be prepared by the assigned instructor.

5. CONCLUSION

The primary goal of creating the IT0 course was to provide IST students with a firm foundation in mathematical reasoning and problem-solving skills while introducing major programming concepts needed for success in more advanced courses. Identifying the IT2008 model curriculum components that closely aligned with the deficiencies observed by the faculty confirmed the need for such a course.

The opportunity to cover basic programming concepts in an earlier course means that the introductory programming course can now provide deeper coverage of important topics such as object orientation. Also, by providing all students with a solid foundation in discrete mathematics topics and problem-solving skills, higher-level courses across the curriculum will benefit. For example, the domino effect of pushing content from the introductory programming course into the IT0 course allows content from the advanced programming course to be pushed into the introductory course. This move allows some courses that previously had Programming II as a prerequisite to now only require introductory programming. Additionally, information security and database programming courses can spend less time covering basic concepts, thereby addressing more advanced content.

The new IT0 course has been approved through university channels and will be offered, and required, for the first time in fall 2014. While the initial work is complete, there is much more to be done. For example, faculty must now measure the effectiveness of the new approach. Plans are underway to provide a pretest and post-test for the IT0 course to ensure learning objectives are being met. Additionally, faculty in upper level courses will monitor the preparedness of IST students, some also using a pretest. It will be difficult to measure the direct impact on the introductory programming course since the content is changing along with the new IT0 prerequisite. However, faculty will be keenly aware of any needed adjustments with the new curriculum.

In addition to measuring effectiveness, future plans for the IT0 course include the development of a custom textbook that will meet the students' needs and serve as a basis in the event instructor assignments change.

Our experience reinforces the fact that each program and its stakeholders are different. While others have found success eliminating a two semester approach to teaching programming (e.g., Colton & Curtis, 2010), ours has been the opposite experience. We believe that the new IT0 course, based on a widely accepted curriculum model, will provide the foundation skills our students need to be successful not only in the IST program, but also as future IT professionals. Further, we hope that our experience in creating a hybrid course to meet specific program needs will be of value to other educators.

6. REFERENCES

ABET. (2012). Criteria for accrediting computing programs. Retrieved from http://www.abet.org/DisplayTemplates/DocsHandbook.aspx?id=3148

ACM/AIS/IEEE-CS Joint Task Force for Computing Curricula. (2005). Computing curricula 2005: The overview report.

Benander, A. C., & Benander, B. A. (2008). Student monks - Teaching recursion in an IS or CS programming course using the Towers of Hanoi. *Journal of Information Systems Education, 19*(4), 455-467.

Bennedsen, J., & Caspersen, M. E. (2005a). *An investigation of potential success factors for an introductory model-driven programming course*. Proceedings of the First International Workshop on Computing Education Research, Seattle, WA.

Bennedsen, J., & Caspersen, M. E. (2005b). Revealing the programming process. *SIGCSE Bulletin, 37*(1), 186-190.

Brown, M. (2013). CS0 as an indicator of student risk for failure to complete a degree in computing. *Journal of Computing Sciences in Colleges, 28*(5), 9-16.

Bruce, K. B., Scot, R. L., Kelemen, C., & Tucker, A. (2003). Why math? *Communications of the ACM, 46*(9), 41-44.

Campbell, P. F., & McCabe, G. P. (1984). Predicting the success of freshmen in a computer science major. *Communications of the ACM, 27*(11), 1108-1113.

Capstick, C. K., Gordon, J. D., & Salvadori, A. (1975). Predicting performance by university students in introductory computing courses. *SIGCSE Bulletin, 7*(3), 21-29.

Chor, B., & Hod, R. (2012). *Cs1001.Py: A topic-based introduction to computer science.* Proceedings of the 17th Annual Conference on Innovation and Technology in Computer Science Education, Haifa, Israel.

Colton, D., & Curtis, A. (2010). Programming proficiency in one semester: Lessons learned. *Information Systems Education Journal, 8*(48), 3-15.

Connolly, R. W., & Paterson, B. (2011). *Even so with the pieces borrowed from others: Dressing an IS program in IT clothing.* Proceedings of the 2011 Conference on Information Technology Education, West Point, NY.

Cook, C. R. (1997). CS0: Computer science orientation course. *SIGCSE Bulletin, 29*(1), 87-91.

Cortina, T. J. (2007). An introduction to computer science for non-majors using principles of computation. *SIGCSE Bulletin, 39*(1), 218-222.

Dierbach, C., Taylor, B., Zhou, H., & Zimand, I. (2005). Experiences with a CS0 course targeted for CS1 success. *SIGCSE Bulletin, 37*(1), 317-320.

Faux, R. (2006). Impact of preprogramming course curriculum on learning in the first programming course. *IEEE Transactions on Education, 49*(1), 11-15.

Gegg-Harrison, T. S. (2005). Constructing contracts: Making discrete mathematics relevant to beginning programmers. *Journal on Educational Resources in Computing, 5*(2), 3.

Gill, G., & Holton, C. F. (2006). A self-paced introductory programming course. *Journal of Information Technology Education, 5*, 95-105.

Glass, R. L. (2000). A new answer to "How important is mathematics to the software practitioner?". *IEEE Software, 17*(6), 135-136.

Harvey, V. J., Wu, P. Y., Turchek, J. C., & Longenecker, J., Herbert E. (2007). Coordinated topic presentations for information systems core curriculum and discrete mathematics courses. *Information Systems Education Journal, 5*(8), 3-10.

Henderson, P. B. (2005). Mathematics in the curricula. *SIGCSE Bulletin, 37*(2), 20-22.

Hoskey, A., & Murino, P. S. (2011). Beyond introductory programming: Success factors for advanced programming. *Information Systems Education Journal, 9*(5), 61-70.

Kelemen, C., Tucker, A., Henderson, P., Astrachan, O., & Bruce, K. (2000). *Has our curriculum become math-phobic? (an American perspective)*. Proceedings of the 5th Annual Conference on Innovation and Technology in Computer Science Education, Helsinki, Finland.

Konvalina, J., Wileman, S. A., & Stephens, L. J. (1983). Math proficiency: A key to success for computer science students. *Communications of the ACM, 26*(5), 377-382.

Legier, J., Woodward, B., & Martin, N. (2013). Reassessing the skills required of graduates of an information systems program: An updated analysis. *Information Systems Education Journal, 11*(3), 79-89.

Lunt, B., Ekstrom, J. J., Gorka, S., Hislop, G., Kamali, R., Lawson, E., . . . Miller, J. (2008). Information technology 2008, curriculum guidelines for undergraduate programs in information technology: ACM and IEEE Computer Society.

Lunt, B., Ekstrom, J. J., Reichgelt, H., Bailey, M., & LeBlanc, R. (2010). IT 2008: The history of a new computing discipline. *Communications of the ACM, 53*(12), 133-141.

McFall, R. L., & DeJongh, M. (2011). *Increasing engagement and enrollment in breadth-first introductory courses using authentic computing tasks.* Proceedings of the 42nd Technical Symposium on Computer Science Education (pp. 429-434). Dallas, TX.

Middleton, D. (2012). Trying to teach problem-solving instead of just assigning it: Some practical issues. *Journal of Computing Sciences in Colleges, 27*(5), 60-65.

Mitchell, W. (2001). Another look at CS0. *Journal of Computing Sciences in Colleges, 17*(1), 194-205.

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Paterson, J.

(2007). A survey of literature on the teaching of introductory programming. *SIGCSE Bulletin, 39*(4), 204-223.

Pioro, B. T. (2006). Introductory computer programming: Gender, major, discrete mathematics, and calculus. *Journal of Computing Sciences in Colleges, 21*(5), 123-129.

Porter, L., Guzdial, M., McDowell, C., & Simon, B. (2013). Success in introductory programming: What works? *Communications of the ACM, 56*(8), 34-36.

Ralston, A. (2005). Do we need any mathematics in computer science curricula? *SIGCSE Bulletin, 37*(2), 6-9.

Remshagen, A. (2010). *Making discrete mathematics relevant*. Proceedings of the 48th Annual ACM Southeast Regional Conference, Oxford, MS.

Rizvi, M., & Humphries, T. (2012). *A Scratch-based CS0 course for at-risk computer science majors.* Proceedings of the Frontiers in Education Conference, Seattle, WA.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education, 13*(2), 137-172.

Rountree, N., Rountree, J., & Robins, A. (2002). Predictors of success and failure in a CS1 course. *SIGCSE Bulletin, 34*(4), 121-124.

Sidbury, J. R. (1986). A statistical analysis of the effect of discrete mathematics on the performance of computer science majors in beginning computing classes. *SIGCSE Bulletin, 18*(1), 134-137.

The Organizational Systems Research Association. (2004). Organizational & end-user information systems model curriculum.

Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K. M., Nunamaker, Jr., J.F., Sipior, J. C., & de Vreede, G. J. (2010). IS2010: Curriculum guidelines for undergraduate degree programs in information systems.

Tucker, A., & Garnick, D. (1991). A breadth-first introductory curriculum in computing. *Computer Science Education, 2*(3), 271.

Uludag, S., Karakus, M., & Turner, S. W. (2011). *Implementing IT0/CS0 with Scratch, App Inventor for Android, and Lego Mindstorms.* Proceedings of the 2011 Conference on Information Technology Education, West Point, NY.

Van Dyne, M., & Braun, J. (2014). *Effectiveness of a computational thinking (CS0) course on student analytical skills.* Proceedings of the 45th Technical Symposium on Computer Science Education, Atlanta, GA.

VanDrunen, T. (2011). *The case for teaching functional programming in discrete math*. Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications, Portland, OR.

Vihavainen, A., Paksula, M., & Luukkainen, M. (2011). *Extreme apprenticeship method in teaching programming for beginners.* Proceedings of the 42nd Technical Symposium on Computer Science Education, Dallas, TX.

White, G., & Sivitanides, M. (2003). An empirical investigation of the relationship between success in mathematics and visual programming courses. *Journal of Information Systems Education, 14*(4), 409-416.

Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In support of pair programming in the introductory computer science course. *Computer Science Education, 12*(3), 197.

Wilson, B. C., & Shrock, S. (2001). Contributing to success in an introductory computer science course: A study of twelve factors. *SIGCSE Bulletin, 33*(1), 184-188.

Wood, D. F., Harvey, V. J., & Kohun, F. (2005). Life-long learning - Making discrete math relevant for information systems professionals. *Issues In Information Systems, VI*(1), 386-390.

Zhang, X., Zhang, C., Stafford, T. F., & Zhang, P. (2013). Teaching introductory programming to IS students: The impact of teaching approaches on learning performance. *Journal of Information Systems Education, 24*(2), 147-155.

Editor's Note:

This paper was selected for inclusion in the journal as an ISECON 2014 Meritorious Paper. The acceptance rate is typically 15% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2014.

APPENDIX A
IST Core Courses

Year 1	
	Installing & Upgrading Computer Systems
	Optimizing & Troubleshooting Operating Systems
	Computing for Business Administration
Year 2	
	LAN Installation & Administration
	Fiscal Aspects of Applied Sciences
	Intro to Programming Logic & Design (IT0 course)
	Introduction to Programming
Year 3	
	Data Applications & Interpretation
	Technical Communication
	Ethical & Legal Issues in IT
	Database Design
	Database Programming
	Web-Based Applications
	IST Electives – 4 courses
Year 4	
	Systems Analysis & Design
	IT Project Management
	Internship
	IST Electives – 6 courses

IST Tracks and Electives

Track: Network & Information Security	**Track: Web Systems Development**	**Non-Track Electives**
Information Assurance	Programming II	Android Application Development
Network Security	Server-Side Web Development	Application Development Environments
WAN Installation & Admin	Advanced Web Application Development	Assistive Technologies & Accessible Web Design
Enterprise Network Mgmt	Software Engineering & Mgmt	Cases in Information Systems Technology
Advanced Enterprise Net Mgmt		Database Administration
		Desktop Publishing Applications
		Health Information Technology
		Information Storage & Mgmt
		Intro to Video Game Design & Industry

APPENDIX B
IT2008 Programming Fundamentals Units/Topics and Core Learning Outcomes in IT0

Topics	Core Learning Outcomes
Fundamental Programming Constructs: *10 hours*	
• Basic syntax and semantics of a higher-level language • Variables, types, expressions, and assignment • Conditional and iterative control structures • Simple I/O • Functions and parameter passing • Structured decomposition • Recursion	1. Analyze and explain the behavior of simple programs involving the fundamental programming constructs covered by this unit. 2. Modify and expand short programs that use standard conditional and iterative control structures and functions. 3. Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions. 4. Choose appropriate conditional and iteration constructs for a given programming task. 5. Apply the techniques of structured (functional) decomposition to break a program into smaller pieces. 6. Describe the mechanics of parameter passing and the issues associated with scoping. 7. Describe the concept of recursion and give examples of its use.
Algorithms and Problem Solving: *6 hours*	
• Problem solving strategies • The role of algorithms in the problem-solving process • Implementation strategies for algorithms • Debugging strategies • The concept and properties of algorithms	1. Discuss the importance of algorithms in the problem-solving process. 2. Identify the necessary properties of good algorithms. 3. Create algorithms for solving simple problems. 4. Use a programming language to implement, test, and debug algorithms for solving simple problems. 5. Apply effective debugging strategies.

APPENDIX C
IT2008 Math and Statistics for IT Units/Topics and Core Learning Outcomes in IT0

Topics	Core Learning Outcomes
Basic Logic: *10 hours*	
• Propositional logic • Logical connectives • Truth tables and validity • Predicate logic • Universal and existential quantification • Limitations of predicate logic	1. Apply formal methods of propositional and predicate logic. 2. Create a truth table to determine whether a given formula in predicate logic is valid. 3. Render a well-formed formula in predicate logic in English. 4. Explain the importance and limitations of predicate logic.
Functions, Relations and Sets: *6 hours*	
• Functions • Relations • Sets and set operations	1. Explain, with examples, the basic terminology of functions, relations, and sets. 2. Perform the standard operations associated with sets, functions, and relations. 3. Relate practical examples to the appropriate set, functions, or relation model, and interpret the associated operations and terminology in context.
Graphs and Trees: *4 hours*	
• Trees • Undirected graphs • Directed graphs • Spanning trees • Traversal strategies	1. Illustrate, by example, the basic terminology of graph theory, and some of the properties and special cases of each type of graph. 2. Demonstrate different traversal methods for trees and graphs. 3. Model problems in IT using graphs and trees.

APPENDIX D
IT0 Master Syllabus

MASTER SYLLABUS

COURSE NO., HOURS, AND TITLE: IST 207-3 Programming Logic and Design

COURSE DESCRIPTION:
This course provides students with the foundation for computer programming covering topics such as problem analysis, flowcharting, pseudocode, and algorithm development. Concepts such as documentations, structured design and modularity are emphasized. The course also introduces topics in discrete mathematics such as number systems, sets and logic, relations and functions, truth tables, trees, and graphs.

PREREQUISITES TO: IST 209

COURSE OBJECTIVES:

Upon successful completion of this course, the student will be able to:

1. Perform conversions and calculations with a variety of number systems
2. Apply formal methods of propositional and predicate logic.
3. Create a truth table to determine whether a given formula in predicate logic is valid.
4. Explain the basic terminology and perform of functions, relations, and sets.
5. Perform standard operations associated with functions, relations, and sets.
6. Illustrate, by example, the basic terminology of graph theory.
7. Demonstrate different traversal methods for trees and graphs.
8. Design structured problem solutions using tools such as flowcharts, pseudocode, and algorithms.
9. Demonstrate knowledge of input, output, variables, data types and validation.
10. Demonstrate knowledge of decision and repetition structures.
11. Demonstrate knowledge of programming functions and modularization.

TOPICAL OUTLINE:

Topics	**Percentages of Time**
I. Topics in Discrete Mathematics	
A. Number systems	4%
B. Propositional logic	4%
C. Truth tables and validity	4%
D. Predicate logic	4%
E. Functions	4%
F. Relations	4%
G. Sets and set operations	4%
H. Trees	4%
I. Graphs	4%

IST 207 Master Syllabus revised 02/18/2013

II. Programming Logic and Design

A. Input, processing, and output	15%
B. Modules	10%
C. Decision structures	10%
D. Repetition structures	15%
E. Functions	10%
F. Input validation	4%

TEXTBOOKS:

Required:

Gaddis, T. (2013). Starting out with programming logic and design (3rd ed.) Upper Saddle River, NJ: Addison-Wesley.

IST 207 Master Syllabus revised 02/18/2013

A Proposed Concentration Curriculum Design for Big Data Analytics for Information Systems Students

John C. Molluzzo
jmolluzzo@pace.edu

James P. Lawler
jlawler@pace.edu

Pace University
One Pace Plaza
New York, New York 10038

Abstract

Big Data is becoming a critical component of the Information Systems curriculum. Educators are enhancing gradually the concentration curriculum for Big Data in schools of computer science and information systems. This paper proposes a creative curriculum design for Big Data Analytics for a program at a major metropolitan university. The design emphasizes expanded learning of business, mathematical and statistical, and presentation skills, in projects of teams, in addition to skills in technology. This paper will be beneficial to educators considering improvement of the curriculum for Big Data Analytics and to students desiring a more contemporary program.

Keywords: analytics, big data, computing curricula, data mining, data science, privacy, security.

1. BACKGROUND AND DEFINITION

"Data is the New Oil" (Smolan and Erwitt, 2012).

Big Data is defined as "bigger and bigger and bigger" (Aiden, & Michel, 2013) aggregates of data that challenge business firms in analyzing business benefit with common software. Big Data dimensions are defined in a diverse variety of structured data, such as traditional transaction data, and non-structured data, such as mobile sensor and social media networking data; in a velocity as to rapid sensitivity to real time timeliness of the data; in a veracity as to the purity of sizable volume; and in sheer streaming volume (Ohlhorst, 2013). Big Data is essentially a data management paradigm shift (Borkovich, & Noah, 2013). Big Data is estimated to be in dozens of terabytes to multiples of petabytes (IBM, 2014), growing 50% each year and 100% every 2 years in business firms (Lohr, 2012); and is estimated to be further impacted by increased information from the "internet of things" (Morozov, 2014), such as consumer wearables (Minsker, 2014a). Firms in the retail industry, such as Walmart, store 2.5 petabytes or 1 quadrillion bytes of data (McAfee, & Brynjolfsson, 2012). The storing of data however is less important than the business benefit to be acquired from the analysis of the data, in cost control, decision improvement and design improvements in processes, products and services (Davenport, 2014), especially from the cross-fertilization of customer social networking and transaction data streaming into firms (Brustein, 2014). The accessibility of such data is apparently a "big deal" (eWeek, 2013), as firms exploit the potential of data analytics in this perceived revolution of technology (Freeland, 2012).

The benefits of Big Data Analytics (henceforth referred to as BDA) are cited frequently in firms (IBM, 2014). Amazon is analyzing data for competitive customer micro-segmentation of products to customize its products and services; Google is analyzing messaging to improve its services (Rosenblatt, 2014); and Tesco and Walmart are analyzing demographics to lower inventory pricing of products and services at their stores. Twitter is analyzing hashtags for more patterns of potential sales from subject trends. Firms are clearly interested in BDA to optimize the outcomes of processes, products and services. Even the government and the health industry (Kim, Trimi, & Chung, 2014) are commencing initiatives in efficiency, decision improvement and cost control from BDA to optimize the outcomes of processes and services (Liyakasa, 2013). Though firms may be storing increased data without increased insight (Minsker, 2014b), the potential of BDA as a profitable attribute, beyond the benefits of Business Intelligence and Operations Research, is evident in the literature (King, 2014). This potential invites consideration of BDA as a differential feature of learning in schools of computer science and information systems.

Graduates from schools of computer science and information systems can contribute to the field of data analytics if the curricula of the schools include BDA. Though firms are investing in BDA, they do not have enough data scientists or specialists (May, 2013) for extracting the potential of their data (McCafferty, 2013). Graduates can contribute to the field if they have analytical business skills (Janicki, Cummings, & Kline, 2013) and content domain expertise skills (Poremba, 2013) to critically evaluate the business (Pratt, 2013) of Big Data. They can contribute to this evaluation if they have computational mathematical and statistical skills (Hulme, 2013), can interpret in a high performance environment the complex event significances of structured and non-structured data, and evaluate potential problem solutions or proposed strategies (Pratt, 2013). They can contribute further if they have privacy and security sensitivity skills in standards for Big Data housed by organizations, especially given intrusion issues as discussed in the literature (Lohr, 2013, Sengupta, 2013, & Angwin, 2014). Moreover, they can contribute notably to the field if they have persuasive presentation skills (Miller, 2013), as individual contributors or members of teams, in proposing solutions and strategies; and they can contribute powerfully if they have skills in visualization (Rao, & Halter, 2013). These skills are beyond the data base analysis, design and development skills in technology (King, 2013). Though few graduates, or even practitioners in industry (MacSweeney, 2013a), have all of these skills to be data scientists, a creative curriculum design in Data Science may improve the breadth of ensemble learning for students currently enrolled in schools of computer science and information systems.

2. FOCUS OF PAPER

"The Big Data [R]evolution holds the promise of empowering all of us with knowledge" (Smolan and Erwitt, 2012).

The proposed concentration of Data Science at Pace University is the focus of this paper. The focus is apt, as firms desire BDA personnel but do not have enough expertise for initiatives on projects (Davenport, 2014). Many firms have a BDA expertise gap (Olavsrud, 2014), despite the hype of pundits. The literature indicates that industry needs 140.0 – 190.0 thousand BDA professionals if not data scientists in 2014 (Manyika, Chui, Brown, Bughin, Dobbs, Roxburgh, & Byers, 2011) and even a high of 4.4 million scientists in 2015 (IBM, 2014). Even though a concentration of Data Science is not enough for an immediate solution, the convergence is current to the expectations of industry and organizations, as they initiate investment in Big Data strategies (Messmer, 2014). The focus of this paper on the Data Science curriculum will benefit educators and students desiring a foundation for an immediately marketable program.

3. CONCENTRATION METHODOLOGY – DATA SCIENCE

"Having the data is only the beginning "(Smolan and Erwitt, 2012).

Pace University is anticipating beginning a concentration in Data Science in 2015 with the offering of the Concepts of Big Data Analytics course. The concentration covers descriptive, predictive and prescriptive analytics (Camm, Cochran, Fry, Ohlmann, Anderson, Sweeney, & Williams, 2015) for data-driven decision making. The concentration is designed for undergraduate students to learn business, mathematical and statistical, presentation, team-playing and high-level technology skills. In the concentration**,** projects are assigned to incubating pseudo data scientist teams (O'Neil, & Schutt, 2014) consisting of different skilled students (Schmerken, 2013) of 3–5 individuals. The projects are focused on the design of processes, products or services in a discrete industry, such as energy, entertainment, finance, health and life sciences, or retail. The projects are to be focused on BDA problems in the industries and are to be furnished with data sets of a massive scale from non-proprietary Web sites and systems, such as www.data.gov, www.enigma.io (Singer, 2014) and www.openwebanalytics.com. The projects are to include internship and mentoring of the student teams from a few firms in the industry that are partnered with the school and even have data analytics employment positions (ITBusinessEdge, 2014). In 2016–2017 a few boutique data scientist firms may be partnered with the school. Technologies in the concentration include, but are not limited to, Apache, Hadoop, MapReduce, NLP for text, NoSQL, Python tools (Knorr, 2013) and SAS tools. The concentration in Data Science is planned to begin in 2016–2017 after the Concepts of Big Data Analytics course, by expanded learning of mathematical, statistical and technology skills that will involve other faculty (King, 2013) in the school. Few schools of computer science and information systems have curriculum design initiatives in Data Science (MacSweeney, 2013b) as envisioned in this paper.

The generic learning objectives of the Data Science concentration are defined below:

- Analyze a business process, product or service for experimental improvement in an organization that can benefit by BDA;

- Collaboratively design a discovery and exploratory method for interpreting the customer data domain dynamics of the process, product or service that include structured and unstructured data sources;

- Collaboratively develop a conceptual data business model for the process, product or service problem and for the solution, infused by intelligence learned in the discovery and exploratory process and by leveraging of a BDA tool(s), integrating a data service process prototype scenario(s) – *what can the firm do better now with the information that it could not do before it had it?* (Provost, & Fawcett, 2013);

- Collaboratively develop a governance plan for the new product or service or a process solution for the firm, informed by customer data privacy rights and security sensitivity standards; and

- Formulate an organizational production plan for integrating the data sources, systems and technologies for the proposed process, product or service solution and for integrating BDA as an overall business strategy.

The proposed courses in the concentration are 3 credits. The outcomes of the concentration are in analytical business skills, creative problem-solving skills, Big Data modeling skills, fundamental mathematical and statistical skills, and presentation and team-playing skills, and also privacy and security sensitivity skills. The goal of the concentration is for its graduates to be business data scientists, not mere scientist technologists. The curriculum is a foundation from which there may be employment postings of BDA specialties for the students upon graduation from the university.

Pending approval by an internal curriculum committee of the school, the concentration of Data Science will begin in fall 2015 with the offering of Concepts of Big Data Analytics, as discussed in this paper. The plan is to rollout the full concentration during 2015–2017 with the following courses.

Concept Courses
- Concepts of Big Data Analytics, a course on critical Big Data modeling of a process, product or service in industry;
- Big Data Maturity Model, a conceptual course on benchmarking of best Big Data organizational practices in industry; and
- Customer Relationship Management (CRM) and Big Data, a conceptual course integrating BDA and household priority relationship strategy in industry.

Domain Courses
- Big Data Analytics in Energy, a domain course integrating BDA projects for decision-making in the energy industry;
- Big Data Analytics in Entertainment, a domain course integrating BDA projects for decision-

making in the entertainment and sports industries;
- Big Data Analytics in the Financial Industry, a domain course integrating BDA projects for decision-making in the international financial services industry;
- Big Data Analytics in Health and Life Sciences, a domain course integrating BDA projects for decision-making in the health and life sciences industry, including ObamaCare initiatives; and
- Big Data Analytics in Retail Industries, a domain course integrating BDA projects for decision-making in the retail industries.

Enabling Courses
- Big Data Ethical Framework, an integrative course on BDA privacy, regulatory and security standards governing analytics professionals in industries and organizations; and
- Big Data Foundation Technology, an integrative course on required BDA high performance infrastructure platform and storage technologies and tools.

The concentration of Data Science is depicted in Figure 1. The concentration is fulfilled in the three conceptual courses, three of the five domain courses, and the two enabling courses of the plan. The concentration is currently designed for the undergraduate students of the school but may be expanded in 2017 for graduate students of the School and of the School of Business of the university.

Table 1 lists courses that can give the student requisite skills in business, mathematics, statistics, and presentation, team-playing and high-level technology. The descriptions are fairly generic and reflect existing courses in most institutions that have a business major.

4. COURSE MODEL – BIG DATA ANALYTICS

" … Big Data is much more than big data" (Smolan and Erwitt, 2012).

The field of data science or data analytics is relatively new, with few consistencies in the content or names of introductory courses.

During January-February, 2014, a scan of the Internet disclosed the following names for introductory courses:

- Advanced Big Data Analytics
- Analytics and Decision Analysis
- Applied Data Science
- Big Data Analytics
- Business Analytics
- Business Intelligence and Analytics
- Data Analytics
- Data Analytics for Information Systems
- Data and Decision Analytics
- Data Warehousing and Analytics
- Elements of Data Analysis
- Introduction to Business Analytics
- Introduction to Data Analytics
- Introduction to Data Science
- Large-scale Data Analysis

For this paper, the Concepts of Big Data Analytics course is outlined in Table 2 of the Appendix.

Table 2 contains five columns corresponding to the online syllabi of five university introductory Data Science or Analytics courses. Over the period of February – March 2014, the authors reviewed the online syllabi of 21 introductory courses that contained Data Analytics or Data Science in their titles, all from Tier I and Tier II universities. The courses represented in Table 2 are a representative sample of the 21 courses. The five columns can be used to compare the Concepts of Big Data Analytics course of this paper to those at these other universities. Note that the omission of a checkmark does not mean the topic is not covered in that course. The checkmarks indicate what was available on the Web sites of the universities. Table 2 does not name the universities corresponding to the five columns**,** in order to avoid any criticism of the universities - table is for comparison only.

The Concepts of Big Data Analytics course emphasizes the concepts behind modern Data Science. The course is conceptual in the sense that the principles behind Data Science are emphasized rather than the tools with which to implement them. Therefore, topics such as R and Python programming, Hadoop, MapReduce, and so on are not covered to any extent in this course. Instead, they are covered, as they are needed in the domain courses to solve industry-specific problems. Some topics in the course require knowledge of probability and statistics. Therefore, the basic statistics course required of all computing majors is a prerequisite for this course. Because the course has no programming requirements, it is accessible to any student in the university who has the statistics prerequisite, which thus includes all business majors in the university.

The course emphasizes the strategic value of data and the use of Data Science teams in an organization, and the use of data-driven decision-making. It introduces students to data-analytic thinking and Data Science principles to facilitate communication between business stakeholders and the Data Science teams. The course also discusses the limitations and pitfalls (e.g., overfitting) of Data Science and the necessity of human involvement in choosing the right data and evaluating the processes and results of the Data Science projects. The ultimate goal of this course is to enable students to participate in the development and proper evaluation of a Data Analytics solution to a business problem.

The text will be Provost, F., & Fawcett, T. (2013), Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. Supporting texts will be Davenport, T.H. (2014), Big Data at Work: Dispelling the Myths, Uncovering the Opportunities. Alternately the following can be used as the text: Davenport, T.H., & Harris, J.G. (2007), Competing on Analytics: The New Science of Winning. The text will supplemented by Analysis INFORMS Magazine.

The course will also discuss in detail several recent case studies of the application of BDA to real business situations. There are many online resources to obtain such cases, (e.g., BDA sites of IBM [2014] and HP Vertica [2014])

5. IMPLICATIONS

"Big Data started as a series of small waves but is morphing into the greatest tsunami of information that humans have ever seen" (Smolan and Erwitt, 2012).

The terms "Big Data", "Data Science", "Data Analytics" are so ubiquitous in the practitioner press that it seems that they are just the next hyped fad that will fade into oblivion in a few years. However, with the ability to process and store the many kinds of data collected by firms and organizations, and the need to use these data to strategic advantage, the field of Data Science will not disappear soon. Several studies, such as Brynjolfsson, Hitt, and Kim (2011), and Tambe (2014), have shown that the more data-driven a firm, the more successful is the firm. Therefore, more and more organizations will be hiring data scientists to take advantage of their ever-growing store of data. There is, however, a problem – our universities are not keeping up with the demand.

> There will be a shortage of talent necessary for organizations to take advantage of big data. By 2018, the United States alone could face a shortage of 140,000 to 190,000 people with deep analytical skills as well as 1.5 million managers and analysts with the know-how to use the analysis of big data to make effective decisions. (Manyika, et al, 2011).

Therefore, it is important for universities to begin developing data scientists who have the requisite technical skills, and the business and content domain knowledge to leverage the data that organizations are now accumulating for advantage (Gillespie, 2014). The program in Data Science proposed in this paper helps to fill this need.

6. LIMITATIONS AND OPPORTUNITIES

The concentration of Data Science and the course on Big Data Analytics are beginning as a program in fall 2015, but evaluation of the impact of the initial program on the students may not be finished until fall 2016.

The curriculum for Data Science with Big Data Analytics is not clearly defined in the literature (Dietrich, Newton, & Corley, 2013), and the field is immature in instruction. Within the next year**,** the authors plan to survey instructors of introductory Data Analytics and Data Science courses with a view towards determining which topics are essential for such courses, and which topics are less so. The authors hope that this future research will help create a common core of topics for an introductory course in Data Analytics/Data Science.

The curriculum design in this paper furnishes important input to instructors in schools of computer science and information systems who want to have an initial program in tandem with trends. The literature indicates BDA as a desirable norm in organizations (Ohlhorst, 2013), an opportunity for the response of schools of computer science and information systems. The model of this paper provides a first step.

7. CONCLUSION

This paper posits a curriculum design of Big Data Analytics in the proposed concentration of Data Science at Pace University. The design includes a discovery and exploratory method of critical Big Data modeling and the improvement of a process, product or service in industry. The design provides for inclusion of an organizational plan for process, product or service solutions, and a production strategy integrating non-traditional and traditional technologies and BDA tools. The design further provides privacy rights and security sensitivity standards. The design is ideal as firms and organizations pursue BDA projects. Few organizations have the full prerequisite skills for BDA projects and strategies. Throughout this paper, the design of BDA proposes the relevance of business, mathematical and statistical, and presentation and team-playing skills, augmenting prerequisite skills in the traditional data base management technologies and in the new non-traditional BDA tools. Overall, this paper provides a beneficial proposal to instructors desiring to initiate BDA and Data Science programs to be in tandem with industrial and organizational trends, and to undergraduate students intending to be in tandem with technological trends.

8. REFERENCES

Aiden, E., & Michel, J-B. (2013). Uncharted: Big data as a lens on human culture. New York, New York: Riverhead Books, 11.

Angwin, J. (2014). Has privacy become a luxury good? *The New York Times*, March 4, 25.

Borkovich, D.J., & Noah, P.D. (2013). Big data in the information age: Exploring the intellectual foundation of communication theory. *Proceedings of the Information Systems Educators Conference (ISECON)*, 30(2550), 2.

Brynjolfsson, E., Hitt, L., and Kim, H.H. (2011). Strength in numbers: How does data-driven decision making affect firm performance? Retrieved May 7, 2014 from http://ssrn.com/abstract=1819486.

Brustein, J. (2014). Data analysis. *Bloomberg Business Week*, March 3, 70.

Camm, J.D., Cochran, J.J., Fry, M.J., Ohlmann, J.W., Anderson, D.R., Sweeney, D.J., & Williams, T.A. (2015). Essentials of business analytics. Stamford, Connecticut: Cengage Learning, 5, 6.

Davenport, T.H. (2014). Big data at work: Dispelling the myths, uncovering the opportunities. Boston, Massachusetts: Harvard Business School Publishing Corporation, 6, 22.

Davenport, T.H., & Harris, J.G. (2007). Competing on analytics: The new science of winning. Boston, Massachusetts: Harvard Business School Publishing Corporation.

Dietrich, D., Newton, P., & Corley, K. (2013). Big data in the information technology / information systems / computer science curriculum. *Proceedings of the Information Systems Educators Conference (ISECON)*, San Antonio, Texas, November, 1-2.

Freeland, C. (2012). In big data, potential for big division. The *New York Times*, January 12, 1-3.

Gillespie, S. (2014). Who is afraid of big data? What big data can do for financial organizations and how to prepare for it. Wall Street & Technology, April 22, 1-3.

Hulme, G.V. (2013). Start class: How crunching numbers can drive decisions. *CSO*, May, 14.

HP Vertica (2014) Retrieved May 8, 2014 from http://www.vertica.com/customers/case-studies/.

IBM (2014). Retrieved May 8, 2014 from http://www.ibm.com/big-data/us/en/big-data-and-analytics/case-studies.html.

Janicki, T.N., Cummings, J., & Kline, D. (2013). Information technology job skill needs and implications for information technology course content. *Proceedings of the Information Systems Educators Conference (ISECON)*, 30(2504), 8.

Kim, G-H, Trimi, S., & Chung, J. (2014). Big-data applications in the government sector. Communications of the ACM, 57(3), 78-85.

King, J. (2013). Spring training for business intelligence experts: It is a whole new ballgame for traditional data analysts, as training focuses on deep knowledge of statistics and computer science. *Computerworld*, May 6, 29, 31.

King, J. (2014). A reinvention convention: Premier 100 snapshots. *Computerworld*, February 24, 16.

Knorr, E. (2013). The big picture for big data. Infoworld: *Modernizing Enterprise Information Technology*, Winter, 4-5.

Liyakasa, K. (2013). Executive demand for data analysis grows. *Customer Relationship Management*, June, 20-21.

Lohr, S. (2012). The age of big data. *The New York Times*: *Sunday Review*, February 11, 2.

Lohr, S. (2013). Big data is opening doors, but maybe too many. *The Sunday New York Times*: *Sunday Business*, March 24, 3.

Lutz, M., & Ascher, D. (2013). Learning Python. Sebastopol, California: O'Reilly Media, Inc.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A.H. (2011). Big data: The next frontier for innovation, competition, and productivity. *McKinsey Global Institute White Paper*, May, 1.

May, T. (2013). Stalking the elusive data scientist. Computerworld, May 6, 14.

McAfee, A., & Brynjolfsson, E. (2012). Big data: The management revolution. *Harvard Business Review*, October, 1-5.

McCafferty, D. (2013). Business must address big data knowledge gaps. *Baseline*, November 13, 2.

MacSweeney, G. (2013a). Banks face big data and information technology talent shortage. *Wall Street & Technology*, September 19, 1-4.

MacSweeney, G. (2013b). Good luck finding a data scientist. *Wall Street & Technology*, October 31, 1-3.

Messmer, E. (2014). Big data: 64% of organizations looking at big data projects. *CIO*, April, 7-8.

Miller, C.C. (2013). The numbers of our lives: Big data, big money, big skill set required. Universities are on it. *The Sunday New York Times*: *Education Life*, April 14, 18-19.

Minsker, M. (2014a). Wearables will be a marketing winner. *Customer Relationship Managemen*t, April, 14.

Minisker, M. (2014b). Thirteen digital marketing myths debunked. *Customer Relationship Management*, April, 23.

Morozov, E. (2014). Silicon Valley is turning our lives into an asset class. *Financial Times*, March 14, 10.

O'Neil, C., & Schutt, R. (2014). Doing data science: Straight talk from the front line. Sebastopol, California: O'Reilly Media, Inc., xv-xvi, 10.

Ohlhorst, F.J. (2013). Big data analytics: Turning big data into big money. Hoboken, New Jersey: John Wiley & Sons, Inc., 3,113.

Olavsrud, T.(2014). Big data: How to close the big data skills gap by training your information technology staff, *CIO*, April, 2

Poremba, S. (2013). The right skills for big data jobs. *Forbes*, December 4, 2.

Pratt, M.K. (2013). From crunch to hunch: Chief information officers must teach data scientists to solve real business problems, not just play with data. *CIO*, December 1, 16, 18.

Provost, F., & Fawcett, T. (2013). Data science for business: What you need to know about data mining and data-analytic thinking. Sebastopol, California: O'Reilly Media, Inc.

Rao, A.S., & Halter, O. (2013). How to build a big data team in five steps. *Baseline*, May 29, 1-2.

Rosenblatt, J. (2014). Is Google too big to sue? Bloomberg *Business Week*, March 6, 38-39.

Schmerken, I (2013). Welcome to the new quants: Data scientists. *Advanced Trading*, May 17, 1-4.

Sengupta, S. (2013). Web privacy becomes a business imperative. *The New York Times*: *Business Day*, March 4, B1, B8.

Singer, N. (2014). A harvest of company details, all in one basket. The New York Times, Business, March 16, p. 3.

Smolan, R., & Erwitt, J. (2012). The human face of big data. Sausalito, California: Against All Odd Productions, 14,20,21,77, 134,161.

Tambe, P. (2014). Big data investment, skills and firm value. Working Paper. Retrieved May 7, 2014 from http://www.datascienceassn.org/sites/default/files/Big Data Investment, Skills, and Firm Value.pdf.

(2013). Big data becoming a big deal. *eWeek*, May 23, 12.

(2014). Big data: What is big data? *IBM White Paper*, March 7, 1-2.

(2014). A big market for big data jobs. *ITBusinessEdge*, March 7, 2.

Editor's Note:

This paper was selected for inclusion in the journal as an ISECON 2014 Distinguished Paper. The acceptance rate is typically 7% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2014.

APPENDIX

Figure 1: Data Science Concentration

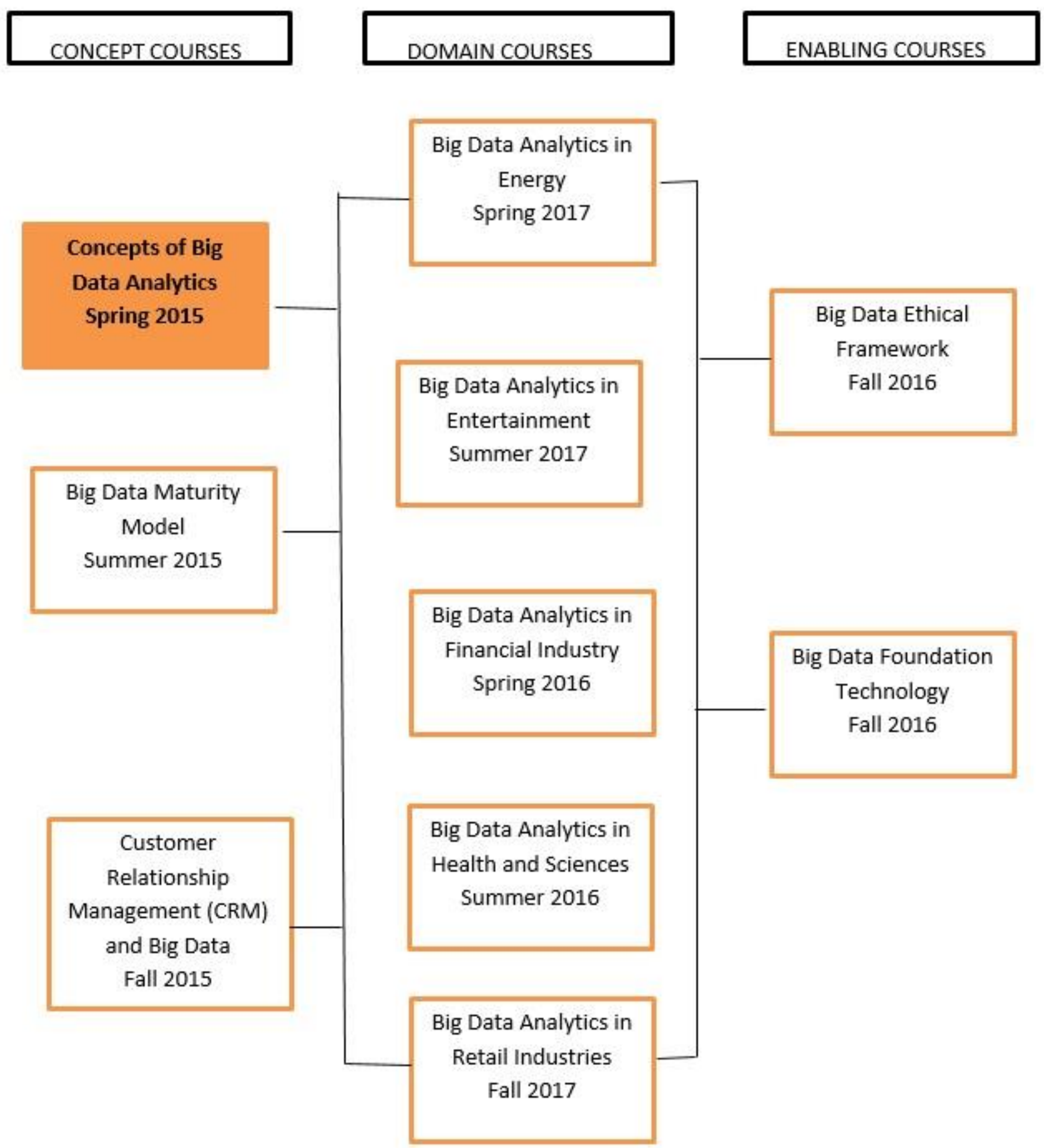


Table 1 - Possible Support Courses	
Course	**Description**
Contemporary Business Practice	The functions of business and their interrelationships. Students work in teams to run simulated companies. Development of business writing and speaking, presentation, and data analysis skills are emphasized.
Calculus I	Limits, continuity, derivatives of algebraic, exponential and logarithmic functions, optimization problems, introduction to integral calculus, the fundamental theorem of integral calculus. Business and economic applications are stressed throughout.
Probability and Statistics	Random processes; finite sample spaces, probability models, independent events, and conditional probability. Bayes' theorem, random variables, mathematical expectation; statistical applications of probability, introduction to sampling theory, confidence intervals and hypothesis testing.
Public Speaking	The mechanics of writing and presenting one's own material. This includes outlining, addressing varied audiences, styles, and appropriate techniques of delivery, as well as the use of technology to enhance one's presentation.
Introduction to Computer Systems	The basic components of a computer, how they are organized, and how they work together under the control of an operating system. Students examine theoretical concepts underlying hardware functions, troubleshooting and preventative maintenance techniques, safety precautions, system procurement, and upgrades, and discuss networking and software as it pertains to hardware functionality.
Financial Accounting	Accounting's role in satisfying society's needs for information and its function in business, government, and the non-profit sector. Students learn from a user-oriented perspective about the accounting cycle, the nature of financial statements and the process for preparing them, and the use of accounting information as a basis for decision making.
Managerial Accounting	A study of the fundamental managerial accounting concepts and techniques that aid in management decision-making, performance evaluation, planning and controlling operations. The emphasis is on the use of accounting data as a management tool rather than on the techniques of data accumulation. The course includes such topics as cost behavior patterns, budgeting and cost-volume-profit relationships. Quantitative methods applicable to managerial accounting are studied.
Managerial and Organizational Concepts	This course examines basic managerial functions of planning, organizing, motivating, leading, and controlling. Emphasis is also given to the behavior of individual and groups within organizations.
Principles of Marketing	This course examines marketing's place in the firm and in society. Considered and analyzed are marketing research and strategies for product development, pricing, physical distribution and promotion, including personal selling, advertising, sales promotion and public relations.
Microeconomics	Theory of demand, production and costs, allocation of resources, product and factor pricing, income distribution, market failure, international economics, and comparative economic systems.
Macroeconomics	National income determination, money and banking, business cycles and economic fluctuations, monetary and fiscal policy, economic growth, and current microeconomic issues.

Table 2 - Syllabus for Concepts of Big Data Analytics						
Week	Topic	Schools				
		A	B	C	D	E
1	**Data-Analytic Thinking**					
	Data Science and Data-Driven Decision Making	X	X	X	X	X
	Data as a Strategic Asset Executive Firm Mentor Presentation	X	X	X	X	X
	Data-Analytic Thinking	X	X	X	X	X
2	**Data Science Solutions to Business Problems**					
	The Data Mining Process	X	X	X		X
	Other Analytics Techniques	X	X	X		
3	**Predictive Modeling**					
	Models		X	X	X	X
	Supervised Segmentation		X	X	X	X
	Visualizing Segmentations		X	X	X	X
	Trees		X	X	X	X
	Probability Estimation					
4	**Model Fitting**					
	Classification Using Mathematical Functions		X		X	X
	Linear Discriminant Function	X				
	Regression		X		X	X
	Logistic Regression		X		X	X
	Non-linear Functions, Neural Networks					X
	Principle Component Analysis					
5	**Overfitting**					
	Overfitting Examples					
	Overfitting Avoidance					
	Complexity Control					
6	**Similarity, Neighbors and Clusters**					
	Similarity and Distance		X	X	X	X
	Nearest Neighbor		X	X	X	X
	Clustering		X	X	X	X

7	**Decision-Analytic Thinking – Creating a Model**					
	Evaluating Classifiers		X			
	Generalizing Beyond Classification					
	Expected Value					
	Outlier Detection					
8	**Visualizing Model Performance**					
	Ranking		X	X		
	Profit Curves					
	ROC Graphs and Curves					X
	Area Under the ROC Curve					X
	Lift Curves					
9	**Evidence and Probabilities**					
	Combining Evidence Probabilistically	X	X			
	Bayes Rule	X	X			
	Evidence Lift	X				
10	**Representing and Mining Text**					
	The Importance of Text				X	
	Text Representation				X	
	N-gram Sequences				X	
	Named Entity Extraction				X	
	Topic Models				X	
11	**Decision-Analytic Thinking – Analytical Engineering**					
	Selection Bias		X			
	Expected Value Decomposition		X			
12	**Other Data Science Techniques**					
	Co-occurrence and Associations					
	Profiling Optional Scientist Firm Mentor Presentation					
	Link Prediction					
	Data Reduction					
	Bias, Variance, the Ensemble Method					X
	Data-Driven Causal Explanation Optional Scientist Firm Mentor Presentation					
	Time Series		X			

13	**Data Science and Business Strategy**					
	Achieving Competitive Advantage with Data Science				X	
	Sustaining Competitive Advantage with Data Science				X	
	Attracting Data Scientists and Teams Optional Scientist Firm Mentor Presentation				X	
	Evaluating Data Science Proposals				X	
	The Kaggle Model		X			
14	**Ethics and Data Science**					
	Data Security	X				
	Privacy	X	X		X	
	ACM Code of Ethics					

A Design Quality Learning Unit in OO Modeling Bridging the Engineer and the Artist

Leslie J. Waguespack
lwaguespack@bentley.edu
Computer Information Systems Department
Bentley University
Waltham, Massachusetts 02452, USA

Abstract

Recent IS curriculum guidelines compress software development pedagogy into smaller and smaller pockets of course syllabi. Where undergraduate IS students once may have practiced modeling in analysis, design, and implementation across six or more courses in a curriculum using a variety of languages and tools they commonly now experience modeling in four or fewer courses in at most a couple of paradigms. And in most of these courses their modeling decisions focus on acceptable syntax rather than principles representing and communicating concepts of quality in information systems. Where learning design quality may once have been an osmotic side effect of development practice it must now be a conscious goal in pedagogy if it is to be taught at all. This paper presents a learning unit that teaches design quality in object-oriented models. The focus on object-oriented models allows the learning to permeate analysis, design, and implementation enriching pedagogy across the systems development life cycle. The quality perspective presented is more expansive than that usually found in software engineering, the traditional "objective" notion of metrics, and integrates aspects of aesthetics, the more subjective phenomena of satisfaction. This learning unit is intended as an adaptable framework to be tailored to the coursework and the overall objectives of specific IS programs.

Keywords: design quality, design, OO modeling, IS discipline, IS curricula, IS pedagogy

1. INTRODUCTION

Over the past decade computing curricula have been repartitioned with the permeation of computing across disciplines and society. (Shackelford, Cross, Davies, Impagliazzo, Kamali, LeBlanc, Lunt, McGettrick, Sloan & Topi, 2005) There are now 5 major computing curriculum guidelines that subdivide computing. (Soldan, Hughes, Impagliazzo, McGettrick, Nelson, Srimani & Theys 2004, Cassel, Clements, Davies, Guzdial, McCauley, McGettrick, Sloan, Snyder, Tymann & Weide, 2008, Diaz-Herrara & Hilburn, 2004, Lunt, Ekstrom, Gorka, Hislop, Kamali, Lawson, LeBlanc, Miller & Reichgelt, 2008, Topi, Valacich, Wright, Kaiser, Nunamaker, Sipior & de Vreede, 2010) The co-location of IS curricula in schools of business further exacerbates the pressure on pedagogy as accreditation bodies further constrain the scope of coursework by compressing systems development into smaller and smaller pockets of course syllabi. (AACSB 2010, EQUIS 2010) Where undergraduate IS students once may have practiced modeling in analysis, design, and implementation across six or more courses in a program using a variety of languages and tools they commonly now experience modeling in four or fewer courses in at most a couple of paradigms. (Waguespack 2011) And in most of these courses their modeling decisions focus on acceptable syntax rather than principles representing and communicating concepts of quality in information systems. Where learning design quality may once have been an osmotic side effect of development practice it must now be a conscious goal in pedagogy if it is to be taught at all.

At the same time industry and academia persist in their lament over the paucity of focus on quality in system design first sounded more than four decades ago (Dijkstra, 1968) and echoing consistently since as in (Denning, 2004, Brooks 1995, 2010, Beck, Beedle, van Bennekum, Cockburn, Cunningham, Fowler, Grenning, Highsmith, Hunt, Jeffries, Kern, marick, Martin, Mellor, Schwaber, Sutherland, & Thomas 2010)

This paper presents a learning unit that teaches design quality within the object-oriented paradigm. The focus on OO models allows the learning to permeate analysis, design, and implementation enriching pedagogy across the systems development life cycle. We amplify a traditional "objective" notion of systems quality (i.e. metrics usually found in software engineering) by integrating the more subjective phenomena of satisfaction, aesthetics. This learning unit is adaptable to the coursework and objectives of specific IS programs. The paper presents: a brief overview of design quality, properties to assess design choices, the object-oriented ontology; and a discussion of how each of the design choice properties express quality through the use of object-oriented modeling constructs. Finally, there is a description of how the learning unit has been integrated in object-modeling syllabi with a comment on its efficacy.

2. WHAT IS DESIGN QUALITY?

Quality is an elusive concept, shifting and morphing on a supposed boundary between science and art: objective, engineering characteristics versus subjective, aesthetic observer or stakeholder experience. International standards of quality reflect the challenge of defining quality by offering a variety of perspectives (as gathered here by Hoyle 2009):

- A degree of excellence (Oxford English Dictionary)
- Freedom from deficiencies or defects (Juran 2009)
- Conformity to requirements (Crosby 1979)
- Fitness for use (Juran 2009)
- Fitness for purpose (Sales and Supply of Goods Act 1994)
- The degree to which the inherent characteristics fulfill requirements (ISO 9000:2005)
- Sustained satisfaction (Deming 1993)

(Waguespack 2010b) asserts that the quality of systems revolves around two primary concepts: efficiency and effectiveness defined as follows (New Oxford American Dictionary):

Efficiency [noun]- the ratio of the useful work performed […] in a process to the total energy [effort] expended

Effectiveness [noun]- successful in producing a desired or intended result

These two concepts appear primarily quantitative and therefore objective. In and of themselves they may well be. Portraying efficiency using a convenient interpretation of "work" and "effort" is genuinely objective. "How many" or "how much" or "how often" often depicts efficiency. But, when we ask "Is it enough?" apparent objectivity fades away.

Likewise, the supposed objectivity of "effectiveness" relies upon the tenuous phrase, "desired or intended result" defined as

Intend [noun]- have (a course of action) as one's purpose or objective; plan

Effectiveness (like efficiency) is a correspondence between a system and its stakeholders' intentions. Assessing effectiveness depends on comparing "what is" to "what is intended." While the former may be expressed quantitatively the latter presents challenges: clarity of conception, mode of representation, scope of contextual orientation, and fidelity of communication to name but a few. Indeed the notion of effectiveness is complicated when we contemplate identifying and quantifying the stakeholder(s) intentions objectively.

The indefiniteness or imprecision that characterizes stakeholder intention(s) is generally not a concern if an observer is asked to assess the beauty of something – an assessment generally conceded to be subjective. A detailed or even explicit intention is not expected in assessing beauty – beauty is most often perceived as an experience of observation rather than a system analysis. Most people commonly accept beauty as subjective and exempt from specific justification or explanation – "Beauty is in the eye of the beholder." and "You'll know it [beauty] when you see it." This absence of or difficulty in forming a quantitative justification of beauty is often the basis for categorizing artifacts or processes as products of art rather than of engineering. And therein lies the presumption that the aspects of design quality that we label objective and those we label subjective are somehow dichotomous. They in fact teeter between objectivity and subjectivity depending on the degree of granularity that observers choose to employ in inspecting not only the artifact but also their own disposition toward satisfaction relative to it.

3. AN ARCHITECTURAL INTERPRETATION OF QUALITY DESIGN

We will never be able to absolutely define design quality because of the relativistic nature of satisfaction in the observer experience. But, our students must still face design choices. So, as IS educators we must provide a framework for them to develop and refine their individual perceptions and understanding of systems quality. The taxonomy of design choice evaluation proposed in Waguespack (2008, 2010b), the 15 *choice properties*, is just such a framework. (See Appendix A.) Choice properties derive from Christopher Alexander's writings on design quality in physical architecture. (Alexander 2002)

Choice properties address the process of building, the resulting structure, and the behavior of systems as cultural artifacts. Every design decision, choice, contributes to the aggregate observer experience: either positively or negatively. Each choice exhibits the 15 properties with varying strengths or influence that impact the resulting observer satisfaction. The confluence of property strength results from the coincidence of the designer's choice with the collective intention of the stakeholders. The combination of all choices with their respective property strengths results in the overall, perceived design quality. Many of the properties are design characteristics long recognized in software engineering (i.e. modularization, encapsulation, cohesion, etc.). But several reach beyond engineering to explain aesthetics, the art (i.e. correctness, transparency, user friendliness, elegance, etc.). An example of the effectiveness of choice properties in explaining the design quality of production systems is reported in (Waguespack, Schiano & Yates 2010a).

4. THE ONTOLOGY OF THE OBJECT-ORIENTED PARADIGM

Illustrating design decisions in the object-oriented paradigm can be a challenge. The idiosyncrasies of OO programming syntax often obscure the intention and/or the result of a design decision. For that reason the learning unit presented here uses a paradigm description independent of programming language, the object-oriented ontology, found in (Waguespack 2009) and excerpted in Appendix B. The graphical outline of the ontology is Figure 1 below.

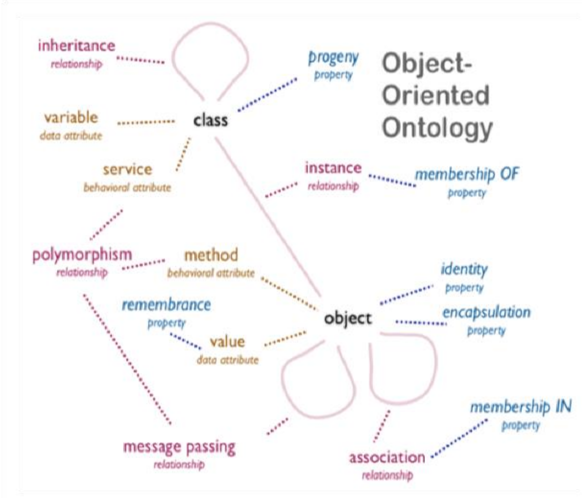


Figure 1 – Object-Oriented Ontology

The ontology captures the elements of the object-oriented paradigm eschewing the obfuscation that usually occurs with programming language syntax examples. At the same time an experienced IS teacher can readily translate the ontological elements into a relevant programming dialect.

5. CRAFTING OBJECT-ORIENTED MODELING CHOICES THAT STRENGTHEN PROPERTIES OF DESIGN QUALITY

This section, the heart of the learning unit, enumerates the 15 choice properties as defined in Waguespack (2010b) illustrating how modeling choices in the object-oriented ontology can express design quality. In this space-limited discussion one choice property often references another reflecting the confluent nature of the design quality properties as Alexander defines them in physical architecture. (Alexander 2002)

Stepwise Refinement (as the name implies) is an approach to elaboration that presumes a problem should be addressed in stages. The stages may represent degrees of detail or an expanding problem scope. (Birrell and Ould 1988) In either case quality evidence of *stepwise refinement* is demonstrated by the cogent and complete representation of a design element at whatever level of detail or scope is set at each stage. To achieve this representation the modeling paradigm must support abstraction that allows generalization of the scope of interest and then the elaboration of that scope from one stage to the next.

The class concept in OO provides this capability. Through the inheritance relationship a class can represent the more abstract, general character of a model feature while expressing all the information and behavior needed at that level of abstraction: 1) what responsibilities the objects of this class fulfill, 2) what information they manage, and 3) what services this class's objects provide the rest of the model. As the modeling stages progress greater specialization is achieved with child classes that redefine abstract behaviors: by adding data and/or behavioral attributes germane only at a lower level of abstraction, or by defining collaborations to support this class's responsibilities. *Stepwise Refinement* can mimic the concept of "need-to-know." Only that detail required to "understand" the system at that abstraction level need be revealed or perhaps is not even chosen until the need arises. When the need does arise the detail may be added within the genealogy of the class preserving the *cohesion* of a class's defined functional responsibility at the higher abstraction levels.

As an example, consider a class that defines items stored in an inventory. At the most general level the most important functional detail is the entry and removal of items. As refinement progresses simple entry and removal may be augmented by including item re-order and supplier interaction both concealed from the inventory item's client who sees only entry and removal. The supplier interaction details are *encapsulated* within the inventory item's responsibilities retaining the *cohesion* of the class's purpose (its *identity)*. And the description of the inventory item exhibits *correctness* at either level of detail with and without the supplier interaction elaboration.

Cohesion is a quality property reflecting a consistent responsibility distribution in a field of system components. (Zuse 1997) Since every object "expects" the objects around it to fulfill their responsibilities to contribute to the whole model, each object is in itself free to be single-minded in its focus on its own purpose. This is the result of well-chosen classes. This independent sufficiency accentuates the divisibility of function in terms of each object's individual purpose, its *identity*, and the clarity with which its purpose is exposed to the rest of the community of objects in the system. The single-mindedness that results also increases the feasibility of object interaction rearrangement enabling an overall change in system function while almost every class's individual purpose remains fixed. The independent sufficiency of each object's inner workings couples with the system-wide interdependency of object cooperation to promote a texture exhibiting a sense of system connectedness, *elegance.*

Encapsulation is a design quality reflected directly in the nature of the object-oriented ontology as objects *encapsulate* both their data and behavioral attributes. *Encapsulation* clearly delineates who is allowed to manipulate system information and who is not. Object data and behavior are only accessible (invoke-able) via the published services defined for each object by its class. When sustained as a discipline this boundary universally designates the object as the finest granule of *modularization*. (Scott 2006) This principle eliminates the possibility of "side effects" where system state changes occur in any manner other than the "contractual" prescription defined in the object's service interface. The isolation of the inside of the object from the outside allows both to evolve without servitude to the implementation of the other (e.g. pursuing efficiency) as an object is obligated only through the published responsibilities in its class's services.

Extensibility is the property of design quality most important in pursuing systems with sustainability essential to cost of ownership economy. This is the vehicle for seamless unfolding in system evolution. Extensibility juxtaposes the potential for new functionality with the effort required to achieve it. (van Vliet 2008). In the object-oriented paradigm class plays the pivotal role by empowering instance and inheritance relationships.

Multiplicity is achieved through instance propagation, *progeny*. Each instance is completely interoperable in any combination with its sibling objects as well as acting as an instance of any ancestor class. Interchangeability both enables and reinforces *modularization*.

Evolution or unfolding is accomplished as class definitions are refined and specialized in their child classes – the relationship called *inheritance*. When a child class extends the scope of the data and behavioral attributes of its parent it honors the pattern set out in the parent without contradiction. *Polymorphism* compensates (through dynamic binding) for any overridden methods. This extension proceeds without any impairment of *correctness* because the interfaces defined in the parent class must be supported in each child class. The parent to child unfolding specializing structure and behavior results in an unbroken thread that binds each class to its ancestry and projects an *identity* down through the generations of class.

Modularization along with *cohesion* expresses "divide and conquer" problem solving augmented by the flexibility of configuring and reconfiguring objects as cooperating agents. *Modularization* also supports *scale* permitting the composition of subsystems of varying scope that hold details in abeyance until they require focus. (Baldwin and Clark 2000) Enlightened module design exposes the solution structure envisioned by the modeler and publishes intentions for further extension by separation of concerns and isolation of *accidents of implementation*. (Brooks 1987) The OO paradigm provides ample facility for defining modules of any size and scope while aggregating and/or nesting their interfaces through deliberate information hiding. The granularity enabled through *modularization* may be applied to facilitate the modeler's formulation of structure as well as the perspective to aid stakeholder recognition and understanding.

Correctness in software engineering is often narrowly defined as computing the desired function. (Pollack 1982) Thriving Systems Theory frames this property upon two outcomes: 1) validation, the clarity and fidelity of the represented understanding of system characteristics, and 2) verification, the completeness and effectiveness of model feature testing both individually and in composition.

Validation depends on the fidelity of the unfolding process; that through the stages of *stepwise refinement* the "essence" of system characteristics are brought forward maintaining their integrity. (Brooks 1987) *Modularization* aids in cataloging and focusing on individual essential characteristics. *Correctness* is the only choice property that directly supports itself! *Correctness* must be a priority at each stage as experience shows that *correctness* shortcomings grow more and more expensive to rehabilitate as evolution progresses – notice "rehabilitate," to restore to normal *life*.

Verification depends on the effective testability of each choice to certify it as "consistent with stakeholder understanding." *Modularization* enables the verification of individual choices or modules. Then relying on the *correctness* inside modules verification can turn to the certification of behaviors resulting from *composition of function*. Experience often leads to dependable *patterns* of classes or modules applicable or adaptable to recurring modeling tasks. Verification in these situations can focus on known areas of fragility/risk limiting the effort required to reach a desired confidence level of *reliability*.

Transparency is evident structure, revealing how things fit and work together. (Kaisler 2005) In the OO ontology "fit together" and "work together" are defined by the structural and behavioral relationships. Individual objects may represent clearly delineated and encapsulated choices, but their cooperation is defined by relationships.

Inheritance explains the structural relationship of classes through the propagation of data and behavioral attributes. Inheritance not only propagates attributes, but also enables a class hierarchy's capacity for exhibiting similarity and difference between parent and child classes. That which is similar (in fact identical) inherited by the child class is assumed and becomes in effect familiar – requiring no reiteration. This "folding" of that which is not changed avoids clutter in the child class description, but may be readily reviewed in the parent.

The behavioral relationships of association, message passing, and polymorphism explain the predictable *patterns* of communication and action. Association uses the property of *identity* to designate membership, ownership, and accessibility among objects. Message passing provides the mechanism for cooperating action between objects providing a disciplined conduit through the encapsulating boundary of objects by using services to convey intention, information, and reaction. Polymorphism allows the abstraction of intention by using the same service name to evoke distinct behaviors from objects of different classes. The identical service names in classes with different methods directly realize the metaphorical abstraction of object behavior where at one level of abstraction the behaviors are the same and at a more detailed level of abstraction their behaviors are distinct.

Composition of Function - As a fundamental tool for managing complexity humans regularly attempt to decompose problems, issues, or tasks into parts that either in themselves are sufficiently simple to permit direct solution or can through recursion be subdivided successively until they become sufficiently simple. This is a defining aspect of *modularization*. When the conception of a part also anticipates reuse then the part takes on a larger significance. The combination of specifying a choice consistent with the essence of system characteristics and then designing the choice as an interchangeable component in multiple super-ordinate choices is a step toward *elegance*. Reusable choices represent an understanding of the essence of the system at a deeper level than an individual application. They represent awareness of the intention, perhaps even the philosophy of the system domain.

Composition of function as a property of design quality is realized in model features that facilitate the extension or retargeting of the model in the future. It is the capacity to combine simple functions to build more complicated ones (Meyer 1988). The retargeting capability may be provided directly to the users of the system in the form of a *programmable* interface. A choice achieving the principle of *composition of function* is marked not only by the function it initially provides the user, but also by the functionality it anticipates and supports even (perhaps) *before* the stakeholders realize the need for the capability.

Identity is at the root of recognition and is another property of design quality not usually defined in software engineering. In the physical world *identity* is literal based upon direct sensorimotor experience: by sight or touch and in some cases by sound or smell – a human experience of the "real" world. In the object-oriented paradigm *identity* is an object property. (Khoshafian and Copeland 1986) Existence is sufficient for object identification.

In other paradigms identification is achieved through possessed characteristics (attributes) that contribute to distinct recognition by a process of intersecting categorizations or the introduction of an artificial characteristic whose sole purpose is to support discrimination. Aside from the fact that these approaches to identification require some overhead (either mental or computational) they are simply not natural to humans. Humans perceive objects as possessing characteristics rather than characteristics defining objects. The former begins with certain uniqueness and progresses toward explanation while the latter begins with uncertainty and attempts to deduce uniqueness.

Characteristics are not unimportant. Classification is essential in most human problem solving activities. And recognition is virtually always accelerated by the discrimination that categorizing characteristics (attributes) provide. And most importantly in the absence of physical experience categorization through characteristics is the only choice. Class structure and the instance relationship are vital to *identity* – an object belongs to "this" class and not to "another." Described both by what an object "knows" (data attributes) and what it "knows how to do" (behavioral attributes) classes form a categorization cornerstone of the object-oriented ontology. But to model both the static and dynamic dimensions of reality (association and message passing) each object must be uniquely distinguishable.

Scale's effect on design quality is reflected in common idioms: "You can't see the forest for the trees!" and "Let's get a view from 10,000 feet." They reflect the importance of context in recognition and decision-making. *Scale* captures the modeling imperative that all choices must be kept in perspective because it is not sufficient to consider a choice only in the microcosm of itself, as it must also participate in the connectedness of the whole. By achieving scale, a system designer provides differing granularities of comprehensibility to suit the requirements of a variety of observers (Waguespack 2010).

The relationships provided in the object-oriented paradigm (association, inheritance, instance, message passing, and even polymorphism) provide ample means for designing collections of cooperating choices that are nested, intersect, or partition the full *field* of functionality essential to the model. These may be called variously subsystems, modules, or sub-modules. In those cases where the actual structure of a collection must be rendered obscure, classes and objects can be devised to serve as facades or agents to "keep up appearances." Coupled with *stepwise refinement,* as it is, *scale* is used to focus modeler and stakeholder attention to achieve the contextual understanding needed to address constituent concerns within the whole.

User Friendliness is another property of design quality more often considered aesthetic. It is a combination of: ease of learning; high speed of user task performance; low user error rate; subjective user satisfaction; and, user retention over time (Shneiderman 1992). Its impact may be easiest to consider in its absence. A modeling choice that is "unfriendly" to stakeholders is confusing, hard to comprehend, unwieldy, and perhaps worst of all, of indeterminate *correctness*. That which defies understanding cannot be determined to be correct. Satisfaction is cumulative. The sensitivity to the stakeholders' conceptions of the essence of the system to be modeled is key to the stakeholders' sense of comfort, familiarity, and expectation.

The object-oriented paradigm excels in its facility to represent systemst preserves the stakeholders' ability to recognize "their" system. Authoring object-oriented models whose elements correspond almost one-to-one with the real-world concepts and entities results in intrinsically better stakeholder understanding and interaction. The casting of "objects" in the models that have direct counterparts in the stakeholders' experience exhibits a fundamentally friendly quality. It respects the stakeholders' perceptions and it welcomes them into the processes of verification and validation that are intrinsic to *correctness*. The unified structure of "what an object knows" and "what an object knows how to do" correlates so naturally with observers of business models or process models that the natural clarity in that communication improves understanding and avoids mistakes in understanding, communication, or implementation.

And in a serendipitous quirk of language (or a profound emergence of the deep meaning of metaphors) Alexander's term from which the principle here, *user friendliness*, is derived is *roughness*. (Alexander 2002) Something has to have a certain degree of *roughness* if one is to be able to effectively grasp it!

Patterns describe versatile templates to solve particular problems in many different situations (Gamma et al. 1995). All entities in the object-oriented paradigm propagate from classes, predefined templates, or "cookie cutters." This protocol organizes what otherwise would be a bewildering multiplicity of individual computational entities to consider. It becomes less complicated in the understanding that the potential of any number of objects boils down to understanding the class(s) of which they are instances. Each instance mimics perfectly the form and function of every other of its siblings, members of that class. Class hierarchies, generations of parent-child class definitions, defining "nearly the same" and "different in specific ways" relationships significantly lessen the apparent complexity that considering only individual entities entails. Class hierarchies define the path of *unfolding* for all to see – a depiction of the analysis, solution, and design philosophies at work.

Patterns is the property of design quality that channels change (unfolding). A pattern foreshadows where and how change will need to be accounted for. Patterns of the form popularized in (Coplein, 1995) document commonly encountered design questions offering carefully considered advice and cautions. Their patterns are paradigm and modeling language independent. However, it is not surprising that many examples using *patterns* are presented in OO dialects. The reason is simple. The integration of instance, inheritance, message passing, and polymorphism relationships is an ideal toolset for expressing *patterns* with a balance of prescription and adaptability – a balance not as conveniently achieved in dialects based on pre-object-oriented paradigms.

Programmability in software engineering is often considered a feature rather than a property of design quality – the capability within hardware and software to change; to accept a new set of instructions that alter its behavior (Birrell and Ould 1988). It is closely allied with *extensibility*and addresses the need for models to welcome the future. What largely separates information systems from other human-made mechanisms is the degree of adaptability that they offer to deal gracefully with change. Unlike most appliances that support a very narrow range of use (albeit with great *reliability*), contemporary information systems are expected to provide not only amplification of effort as in computation, but also amplification of opportunity in terms of different approaches to business or organizational questions. Contemporary information systems are expected to demonstrate that they can reliably accommodate change. As with *extensibility,* successful accommodation of change relies on an understanding of the fundamental options governing the structure and behavior within a particular domain. The OO ontology offers powerful tools (structural and behavioral relationships, e.g. inheritance and polymorphism) to service the elements of change without fracturing a skeletal foundation of base classes characterizing the domain.

What sets *programmability* apart from *extensibility* is a facility that permits altering the systems behavior without having to reconstruct choices – that is to say that the system's behavior can be sensitive to the context determined by a "user" in "real time." "Real time" is relative to the "user's" role (e.g. developer or end user, etc.). This versatility is not accidental but architectural. Choices may provide an interface language for end users that permits selections of system actions to meet an immediate "real-time" need – an interface as simple as a light switch or as complex as a natural language.

Reliability is a property of design quality more often associated with implementation than design. It is the assurance that a product will perform its intended function for the required duration within a given environment (Pham 2000). Objects facilitate modularized testing and quality assurance. A certified class produces certified objects (which is not to say that certification is easy or inexpensive). As long as classes are protected from dynamic modification in deployment there is no need to be concerned with the inner workings of their objects. As long as objects are truly *encapsulated* they conform to the intention of their class. In development testing proceeds incrementally as new classes are added or rearranged in their collaboration. Once deployed testing is relegated to their interactions rather than their definition. Testing is compartmentalized and does not explode exponentially when additional classes or functionality within a class is added.

Reliability in design reflects an austerity that confines design elements to the essentials of the stakeholder's intentions. When design or implementation decisions involve additional constructs due to technology or compatibility, these *accidents of implementation* must be clearly delineated so as not to imply that they are essence rather than accident. This clear distinction will protect future system evolution from mistaking accidental "baggage" as stakeholder intentions.

Elegance is perhaps the epitome of subjective quality assessment that clearly sets choice properties of design quality apart from traditional software engineering metrics. "Pleasing grace and style in appearance or manner," that's how the dictionary expresses the meaning of "elegance". (Oxford English Dictionary)

> "A designer knows he has achieved perfection not when there is nothing left to add, but when there is nothing left to take away." (Raymond 1996)

Models composed of choices that are consistent, clear, concise, coherent, cogent, and transparently correct exude *elegance* and nurture cooperation, constructive criticism and stakeholder community confidence. These are models that confess to their own shortcomings because their clarity obscures nothing, even omissions. These are models that satisfy stakeholders. They appear "intuitively obvious." The clarity of their composite structure is so self-evident that they seem "simple." The use of the OO paradigm to construct a collection of "building blocks" in the form of a class library to encapsulate architectural design decisions facilitates this impression of what is "intuitively obvious." Using well-conceived library elements becomes so second nature, so natural, that the builder perceives the blocks as the natural primitives of construction rather than constructed artifacts.

Elegance largely proceeds from the efficient and effective representation of *essential* system characteristics along with those features emerging out of design decisions, *accidents of implementation*, that are laid out with equal clarity for separate consideration. This is the *field effect* of the beneficial, integrated, mutual support of strong choices described in Thriving Systems Theory. (Waguespack 2010b)

6. INTEGRATING THE DESIGN QUALITY LEARNING UNIT IN AN OBJECT-ORIENTED MODELING SYLLABUS

For the past six semesters the design quality learning unit presented here is woven into two object-oriented modeling syllabi: 1) undergraduate systems analysis and design and 2) masters level object-oriented systems engineering. The unit content appears throughout the pedagogy of modeling using UML-2 syntax.

After initially presenting the object-oriented paradigm using the ontology to establish its vocabulary (see Appendix B), we present use case, class, and sequence diagramming establishing the syntax and the expression of semantics in UML-2. During this UML presentation we repeatedly allude to the design quality properties through the syntax. Small student groups and then individuals conduct a series of modeling exercises based on requirement narratives establishing the students' grasp of UML syntax. On that foundation the explanation of design quality, the enumeration of the fifteen properties, and the corresponding application of OO ontology elements to strengthen the properties precede a final individual course modeling project. The design quality discussion provides a quality vocabulary for one-on-one consultations between teacher and student as each develops the object-model of their final project. In this one-on-one context each student's specific design decisions are discussed and evaluated in relationship to the design quality properties, an opportunity for individualized, reinforced learning and/or suggested improvements.

The deeper subtleties of design quality present a challenge for some students particularly in a compressed format. The "light doesn't go on" right away for all students. However, the integration of the ontology and design quality property based vocabulary establishes a touchstone that returning students report helps them "to name" the "quality elements" they rediscover in succeeding coursework and professional practice.

In your own curricular situation the distribution of learning unit elements may span more than one course (some addressed in OO programming, requirements engineering, or database design, etc.), be rearranged to suit your modeling tools, or be adjusted to your course sequencing with context-appropriate examples. Regardless, the learning unit components are flexible and robust enough to suit various specific program needs.

7. ACKNOWLEDGEMENTS

Thanks to helpful referees. Special thanks are due my colleagues David Yates and Bill Schiano at Bentley University for their insightful discussions and comments on these ideas.

8. REFERENCES

AACSB (2010). Eligibility Procedures and Accreditation Standard for Business Accreditation. Retrieved July 16, 2010 from http://www.aacsb.edu/accreditation/AAACSB-STANDARDS-2010.pdf

Alexander C, (2002). The Nature of Order An Essay on the Art of Building and the Nature of the Universe: Book I - The Phenomenon of Life, Berkeley, California: The Center for Environmental Structure, p. 119

Baldwin, C. Y., and Clark, K. B. (2000). Design Rules, Volume 1: The Power of Modularity. The MIT Press, Cambridge, MA.

Beck K., Beedle M., van Bennekum A., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R.C., Mellor S., Schwaber K., Sutherland J., & Thomas D. (2010). *Manifesto for Agile Software Development*. Retrieved July 12, 2010 from agilemanifesto.org

Birrell, N. D., and Ould, M. A. (1988). A Practical Handbook for Software Development. Cambridge University Press, Cambridge, UK.

Brooks F. P. (1987), "No Silver Bullet: Essence and Accidents of Software Engineering," Computer, Vol. 20, No. 4, pp 10-19.

Brooks, F. P. (1995). The Mythical Man-Month: Essays on Software Engineering (2ed). Addison-Wesley, Boston, MA.

Brooks, F. P. (2010). The Design of Design: Essays from as Computer Scientist. Addison-Wesley, Pearson Education, Inc., Boston, MA.

Cassel L., Clements A., Davies G., Guzdial M., McCauley R., McGettrick A., Sloan B., Snyder L, Tymann P., & Weide B.W., (2008). Computer Science Curriculum 2008 An Interim Revision of CS2001. Association of Computing Machinery (ACM), & IEEE Computing Society (IEEE-CS)

Coplien J and Schmidt D (Eds) (1995). Pattern Languages of Program Design, Addison-Wesley, Reading, MA, USA

Crosby,P. B., (1979) Quality is Free, McGraw-Hill, New York, NY, USA.

Dijkstra, E. (1968). "GOTO Statement Considered Harmful." *Communications of the ACM*, 11(3), 147-148

Diaz-Herrara, J.L., & Hilburn, Thomas B. (eds.) (2004). Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, IEEE Computing Society (IEEE-CS), Association of Computing Machinery (ACM)

Deming, W. E. (1993), The New Economics for Industry, Government, Education (2ed), Cambridge Press: MIT, Cambridge, MA, USA

Denning, P. J. (2004). "The Great Principles of Computing," *Ubiquity*, 4(48), 4–10

EQUIS (2010). *EQUIS Standards and Criteria*. Retrieved July 16, 2010 from http://www.efmd.org/attachments/tmpl_1_art_041027xvpa_att_080404qois.pdf

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, MA.

Hoyle, D. (2009). ISO 9000 Quality Systems Handbook. Butterworth-Heinemann (Elsevier); 6 ed. Burlington, MA, USA

ISO 9000 (2005), http://www.iso.org/iso/qmp

Juran, J. M., (1999). Quality Control Handbook (6ed), McGraw-Hill, New York, NY, USA

Kaisler, S. H. (2005). Software Paradigms. Wiley-Interscience, Hoboken, NJ.

Khoshafian, S. N., and Copeland, G. P. (1986). "Object identity," *Proceedings of ACM Conference on Object Oriented Programming Systems Languages and Applications*, Portland, OR, November 1986, 406-416.

Lunt, B.M., Ekstrom, J.J., Gorka, S., Hislop, G., Kamali, R., Lawson, E., LeBlanc, R., Miller, J., & Reichgelt, H. (eds.) (2008). Information Technology 2008: Curriculum Guidelines for Undergraduate Degree Programs in Information Technology, Association of Computing Machinery (ACM), IEEE Computing Society (IEEE-CS)

Meyer, B. (1988). Object-oriented Software Construction. Prentice Hall, New York, NY.

Pham, H. (2000). Software Reliability. Springer, Berlin, Germany.

Pollack, S. (Ed.). (1982). Studies in Computer Science. Mathematical Association of America, Washington, DC.

Raymond, E. S. (1996). The New Hacker's Dictionary, 3rd ed. The MIT Press, Cambridge, MA.

Sales and Supply of Goods Act 1994, Ch 35, Legislation of Her Majesty's Government, The National Archives, UK, http://www.legislation.gov.uk/ukpga/1994/35/introduction

Scott, M. L. (2006). Programming Language Pragmatics, 2nd ed. Morgan Kaufmann, Maryland Heights, MO.

Shackelford, R., Cross, J.H., Davies, G., Impagliazzo, J., Kamali, R., LeBlanc, R., Lunt, B., McGettrick, A., Sloan, R., & Topi, H., (2005). Computing Curricula 2005: The Overview Report, Association for Computing Machiner (ACM), The Association of Information Systems (AIS), The Computer Society (IEEE-CS)

Shneiderman, B. (1992). Designing the User Interface: Strategies for Effective Human-Computer Interaction, 2nd ed. Addison-Wesley, Reading, MA.

Soldan, D., Hughes, J.L.A., Impagliazzo, J., McGettrick, A., Nelson, V.P., Srimani, K., & Theys, M.D. (eds.) (2004). Computer Engineering 2004: Curriculum Guidelines for Undergraduate Degree programs in Computer Engineering, IEEE Computer Society (IEEE-CS), Association for Computing Machinery (ACM)

Topi, H., Valacich, J.S., Wright, R.T., Kaiser, K.M., Nunamaker, J.F. Jr., Sipior, J.C., & de Vreede, G.J. (eds.) (2010). IS2010: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems, Association for Computing Machinery (ACM), Association for Information Systems (AIS)

Van Vliet, H. (2008). Software Engineering: Principles and Practice, 3rd ed. Wiley, Hoboken, NJ.

Waguespack, L. J. (2008). "Hammers, Nails, Windows, Doors and Teaching Great Design," *Information Systems Education Journal*, 6 (45). http://isedj.org/6/45/. ISSN: 1545-679X

Waguespack, L. J. (2009). "A Two-Page "OO Green Card" for Students and Teachers," *Information Systems Education Journal*, 7 (61). http://isedj.org/7/61/. ISSN: 1545-679X

Waguespack, L. J., Schiano, W. T., Yates, D. J. (2010a). "Translating Architectural Design Quality from the Physical Domain to Information Systems," *Design Principles and Practices: An International Journal*, 4, 179-194

Waguespack, L. J. (2010b). Thriving Systems Theory and Metaphor-Driven Modeling, Springer, London, U.K.

Waguespack, L. (2011). "Design, The "Straw" Missing From the "Bricks" of IS Curricula," *Information Systems Education Journal,* 9(2) pp 101-108. http://isedj.org/2011-9/ ISSN: 1545-679X

Zuse, H. (1997). A Framework of Software Measurement. Walter de Gruyter, Berlin, Germany.

Appendix A – Choice Properties (Waguespack 2010b)

	Choice Property	Modeling Action	Practical Action Definition
1	Stepwise Refinement	elaborate	develop or present (a theory, policy or system) in detail
2	Cohesion	Factor	express as a product of factors
3	Encapsulation	encapsulate	enclose the essential features of something succinctly by a protective coating or membrane
4	Extensibility	extend	render something capable of expansion in scope, effect or meaning
5	Modularization	modularize	employing or involving a module or modules as the basis of design or construction
6	Correctness	align	put (things) into correct or appropriate relative positions
7	Transparency	expose	reveal the presence of (a quality or feeling)
8	Composition of Function	assemble	fit together the separate component parts of (a machine or other object)
9	Identity	identify	establish or indicate who or what (someone or something) is
10	Scale	focus	(of a person or their eyes) adapt to the prevailing level of light [abstraction] and become able to see clearly
11	User Friendliness	accommodate	fit in with the wishes or needs of
12	Patterns	pattern	give a regular or intelligible form to
13	Programmability	generalize	make or become more widely or generally applicable
14	Reliability	normalize	make something more normal, which typically means conforming to some regularity or rule
15	Elegance	coordinate	bring the different elements of (a complex activity or organization) into a relationship that is efficient or harmonious

Appendix B - OO Green Card (Waguespack 2009)

THE OBJECT-ORIENTATION "GREEN CARD" SEPTEMBER 22, 2007

The OO Paradigm

Without a Language or Syntax!

What is the object world all about?

The Object-Oriented System Ontology

This ontology is consistent with the practice in computer science and information science categorizing a domain of concepts (i.e. individuals, attributes, relationships and classes). In this ontology of the object-oriented paradigm I attempt to minimize the vestiges of implementation languages and development methodologies in order to expose the core nature and value of object-oriented concepts.

1. Individuals

The most concrete concept in the object-oriented paradigm is the ***object***. It derives from the living physical experience of humans seeing and touching things. In that experience objects are separable – distinguishable from other objects by nature of their physical presence and location regardless of any other discernible characteristics they may possess. This characteristic of "individual-ness" leads to the property of ***identity***. Identity enables the unambiguous designation or selection of every object physical or abstract within a domain of discourse.

Objects have an "inside," an "outside," and a "surface" that separates the inside from the outside. An object contains anything that exists on the "inside" of the object. Since the surface of most physical objects is opaque, usually the contents are invisible and untouchable by anyone on the outside. This property renders the object's contents impervious to meddling and is called ***encapsulation*** (or ***information hiding*).**

2. Attributes

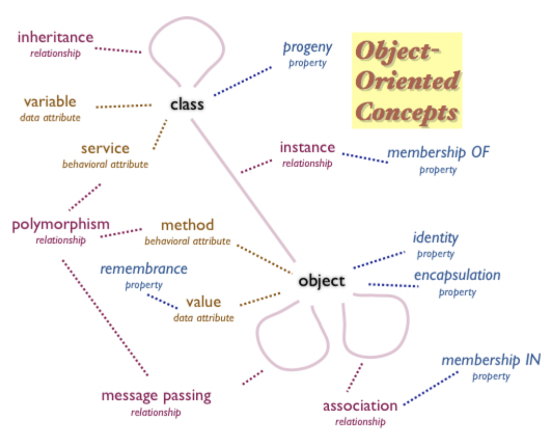
Attributes are those characteristics that are inherent to an *object*. In the object paradigm attributes define either data or behavioral characteristics - each of which has a static and dynamic form. Attributes in static form combine to define what is called the *structure* of an *object*. From inception to extinction the *structure* of an *object* is immutable.

2.1. Data Attributes

Data attributes serve to store information (data) within an *object* and implement the property of ***remembrance***. Data attributes are completely contained within an object protected by *encapsulation*. *Remembrance* is manifest statically as "what can be remembered," a ***data attribute variable***. It is manifest dynamically as a definition of "what is remembered," a particular ***data attribute value***.

2.2. Behavioral Attributes

Behavioral attributes serve to define the animate nature of an *object*. In its static form each behavioral attribute defines "*what* an object can do," usually called a ***service***. In its corresponding dynamic form this behavioral attribute defines "*how* a service is accomplished," usually called a ***method*** (or ***operation***). *Methods* define "activity" performed in an object model. A *method* may simply be access to *remembrance* inside an object or it may be complex sometimes employing the involvement of other *services* of the same or other objects to accomplish its responsibility. *Methods* reside within the *object* subject to *encapsulation* while *services* are visible at the surface of the *object* available for collaboration.



3. Classes

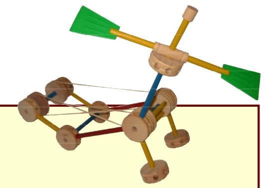
The ***class*** concept combines both a definition of *structure* and the generation of *object(s)* based on that *structure*. Every *object* is an ***instance*** of a specific *class* and shares the same static *structure* defined by that *class* with every other *object* of that *class*. The responsibility of generating *instances* that share the same *structure* is the property of ***progeny***. The *class* concept thereby fuses the existence of the *objects* to that of their *class*; *objects* cannot exist independent of their defining *class*. *Objects* are said to be ***members of*** their *class*.

©2007, Les Waguespack, Ph.D. PAGE 1

Along with the static behavioral structure of *service* defined in the *class*, the dynamic behavioral attribute, *method*, may also be defined. Defined in the *class* this dynamic behavioral attribute, "how a service is accomplished," is identical for each and every *object* generated of that *class*.

4. Relationships

Relationships in the object paradigm exist on two dimensions: structural and behavioral. The structural relationships are based primarily on the properties of *identity, remembrance* and *progeny*.



Without syntax?

Every *language* that is invented to express concepts carries with it the understanding and the biases of the inventor. Depending on his/her purpose(s) those biases simplify certain tasks performed with the language but may obscure the underlying concepts.

Programming language design must deal with the feasibility of automated translation and interoperability with other programming languages and operating systems. Compromises and assumptions are chosen to make the resulting language efficient, effective and marketable.

The goal of this description of the object-oriented paradigm is to succinctly make the concepts understandable - an ambitious task to say the least!

- Professor Waguespack

4.1. Structural Relationships

4.1.1. Inheritance

Inheritance is a relationship between *classes*. The *structure* defined in one *class* is used as the foundation of *structure* in another. By foundation it is meant that all the *structure* of the first is replicated in the second and additional *structure* in terms of *data attributes* or *services* may be added or *methods* for replicated *services* may be altered (***overridden***). The replicated *structure* defines how the two *classes* are alike. The additions or alterations define how they are different. The *class* defining all the *structure* shared between them is called the ***parent class*** (*super class*, *generalization*) while the other is called the ***child class*** (*sub class*, *specialization*). It is said that the *child class* proceeds from or is derived from the *parent class*. Successive application of *inheritance* defining related *classes* results in a ***class hierarchy***.

4.2. Behavioral Relationships

The behavioral relationships are based primarily on the property of ***membership IN***, and the capacity of *objects* to "act."

4.2.1. Association

An ***association*** is a relationship between *objects*. *Objects* are intrinsically separable by way of the *identity* property. At the same time, humans are compelled to categorize their experience of things in the physical world. Humans superimpose groupings that collect *objects* ***into*** sets (a foundation of mathematics based on human experience). *Objects* become members in a group only by designation. This property is called ***membership***. *Membership* is independent of *identity* or *attribute*. This property also permits humans to identify an *object* that is not in a set (i.e. discrimination). (*Membership in a group is discretionary and is distinct from membership of a class which is intrinsic by way of progeny.*)

Variations on *membership* derive from the intent of the relationship and generally fall into the categories of *association* and *composition*. Any designated collection of objects defines a relationship between those *objects* called ***association***. By the simple fact that they are members in the same relationship that membership defines how they relate. When the existence of the *objects* themselves is coupled with their membership; that is to say, if one (or the other or both) would not exist if it were not related to the other then the relationship is called a ***composition***.

4.2.2. Message Passing

Message passing is a relationship between *objects*. *Message passing* relies on the *identity* property and *services*. A ***message*** is a communication between a ***sender*** *object* and ***receiver*** *object* where the *sender* requests that the *receiver* render one of its *services*. The *sender* and *receiver* may be one in the same *object*. The *message* designates the *receiver*'s *identity*, the *receiver*'s *service* to be performed along with any parameters that the *service*'s protocol may require. Since the *message* is a request there are no implicit timing constraints determining when the *service* is accomplished. Unless explicitly designated a *message* results in an asynchronous activity on the part of the *receiver* without acknowledgment or returned information.

4.2.3. Polymorphism

Polymorphism results from the interplay of *message passing*, *behavioral attributes* and *classes*. A *sender* directs a *message* to a *receiver* designating a *service* of that *receiver*. A *message* does not designate a *method*. The regime that determines which *method* satisfies a service request is called ***binding***. If the *method* (corresponding to the *service*) is defined in the *class* of the *receiver object*, that *method* is invoked. If the *service* of the *receiver*'s *class* is *inherited* (and not *overridden*), the corresponding *method* defined in the nearest progenitor (*parent class*) of the receiving *object*'s *class* is invoked.