Towards a Standardized Representation of Syllabi to Facilitate Sharing and Personalization of Digital Library Content

Part of the research project: Personalization of Content: Bridging the gap between NSDL and its users.

Funded by the National Science Foundation, USA.

Objectives

- A syllabus defines a course offering
- If we could standardize information contained in syllabi, it could be used for various applications:
  - Personalizing digital library content
  - Comparing and creating new syllabi
  - Mobile access

Syllabus Contents

Title, Description, Instructor, Teaching Assistants, Prerequisites, Topics, Knowledge Units, Learning Objectives, Calendar, Readings, Books, Book Chapters, Articles, Papers, Instructor’s Notes, Slides, Assignments

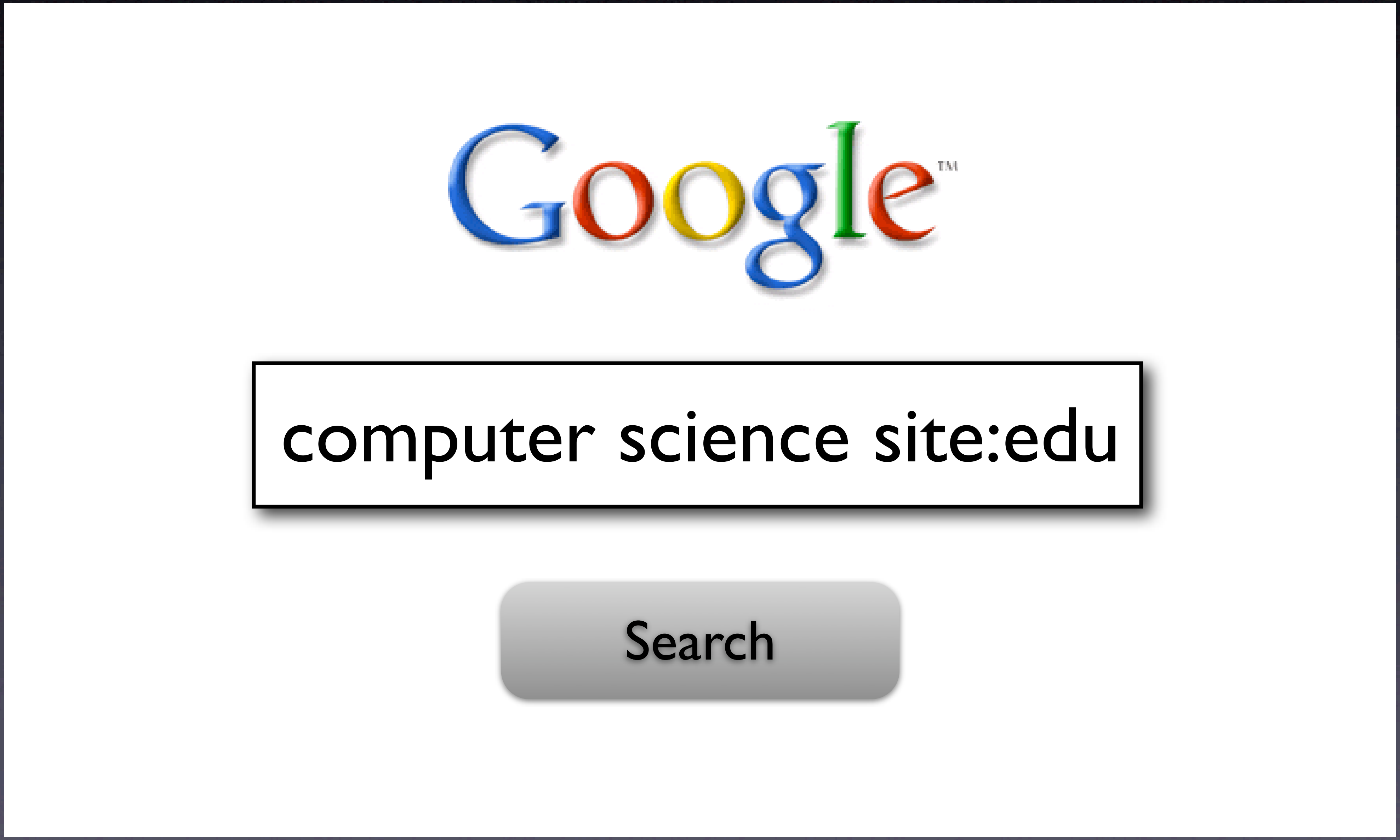
Syllabi published today

- Published via:
  - Instructor's Web site
  - Course management systems
  - University course catalogs (summary)
- Format: HTML, PDF
- Access: Closed or open to public

The Adoption Problem

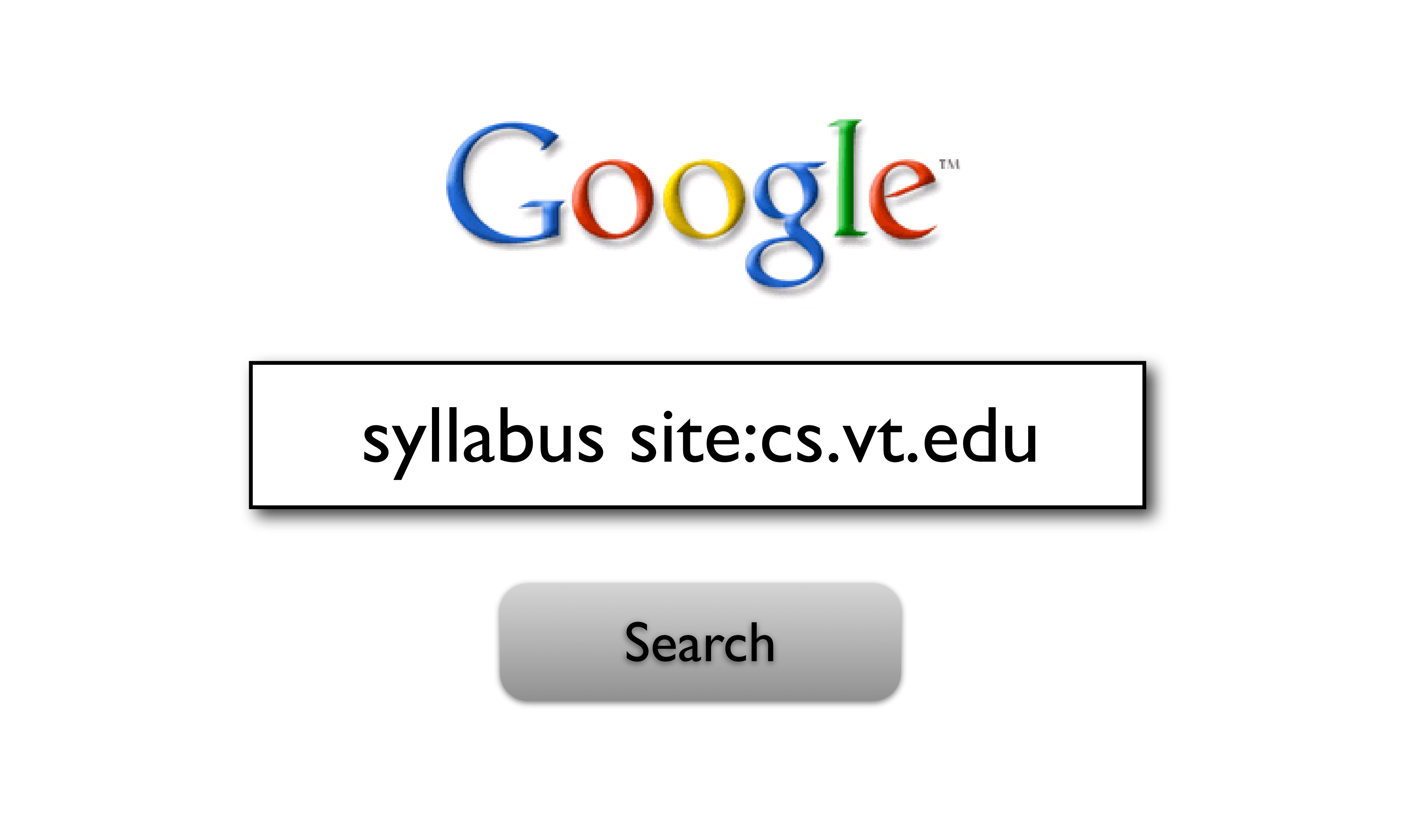
- Chicken-and-egg situation
- Solution: Crawl and parse existing syllabi
  - Develop a schema
  - Store in a repository
  - Develop tools & applications
  - Encourage creation of new syllabi conforming to schema

Obtaining Syllabi (step 1)



computer science site:edu

Obtaining Syllabi (step 2)



syllabus site:cs.vt.edu

Categorization

- Full syllabus
- Partial syllabus
- Syllabus entry page
- Noise

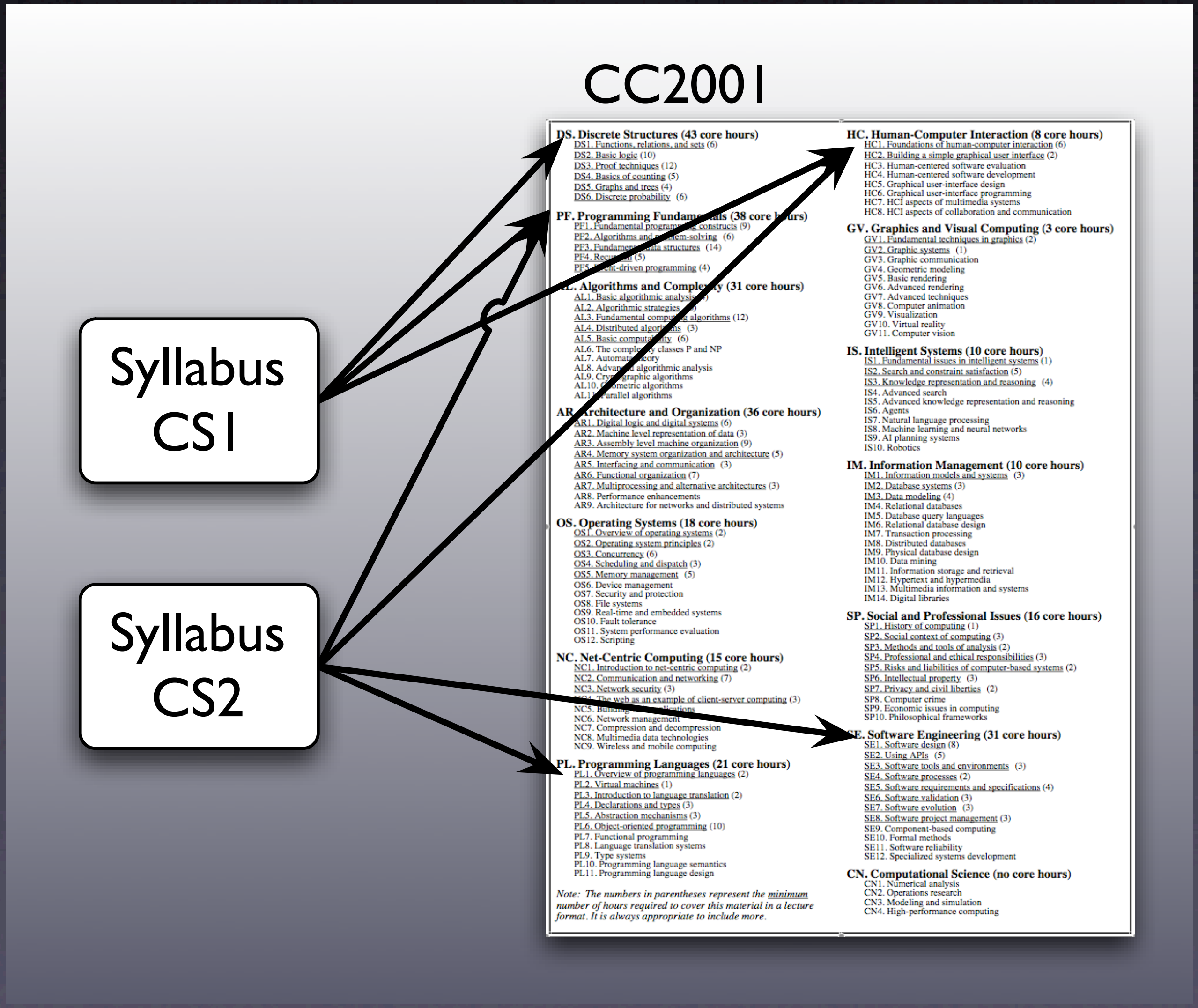
- Labeled 1000 documents as a training set

Tools

- Syllabus creators / editors
- Syllabus repository
- Community-assisted classification and error correction
- Linking to Computing Curricula 2001

Comparing Syllabi

HCI SE Curriculum



DS. Discrete Structures (43 core hours)
DS1. Functions, relations, and sets (6)
DS2. Basic logic (10)
DS3. Proof techniques (12)
DS4. Basics of counting (5)
DS5. Graphs and trees (4)
DS6. Discrete probability (6)

PF. Programming Fundamentals (38 core hours)
PF1. Fundamental programming constructs (9)
PF2. Algorithms and problem-solving (6)
PF3. Fundamental data structures (14)
PF4. Recursion (5)
PF5. Event-driven programming (4)

AL. Algorithms and Complexity (31 core hours)
AL1. Basic algorithmic analysis (4)
AL2. Algorithmic strategies (6)
AL3. Fundamental computing algorithms (12)
AL4. Distributed algorithms (3)
AL5. Basic computability (6)
AL6. The complexity classes P and NP
AL7. Automata theory
AL8. Advanced algorithmic analysis
AL9. Cryptographic algorithms
AL10. Geometric algorithms
AL11. Parallel algorithms

AR. Architecture and Organization (36 core hours)
AR1. Digital logic and digital systems (6)
AR2. Machine level representation of data (3)
AR3. Assembly level machine organization (9)
AR4. Memory system organization and architecture (5)
AR5. Interfacing and communication (3)
AR6. Functional organization (7)
AR7. Multiprocessing and alternative architectures (3)
AR8. Performance enhancements
AR9. Architecture for networks and distributed systems

OS. Operating Systems (18 core hours)
OS1. Overview of operating systems (2)
OS2. Operating system principles (2)
OS3. Concurrency (6)
OS4. Scheduling and dispatch (3)
OS5. Memory management (5)
OS6. Device management
OS7. Security and protection
OS8. File systems
OS9. Real-time and embedded systems
OS10. Fault tolerance
OS11. System performance evaluation
OS12. Scripting

NC. Net-Centric Computing (15 core hours)
NC1. Introduction to net-centric computing (2)
NC2. Communication and networking (7)
NC3. Network security (3)
NC4. The web as an example of client-server computing (3)
NC5. Building web applications
NC6. Network management
NC7. Compression and decompression
NC8. Multimedia data technologies
NC9. Wireless and mobile computing

PL. Programming Languages (21 core hours)
PL1. Overview of programming languages (2)
PL2. Virtual machines (1)
PL3. Introduction to language translation (2)
PL4. Declarations and types (3)
PL5. Abstraction mechanisms (3)
PL6. Object-oriented programming (10)
PL7. Functional programming
PL8. Language translation systems
PL9. Type systems
PL10. Programming language semantics
PL11. Programming language design

Note: The numbers in parentheses represent the minimum number of hours required to cover this material in a lecture format. It is always appropriate to include more.

HC. Human-Computer Interaction (8 core hours)
HC1. Foundations of human-computer interaction (6)
HC2. Building a simple graphical user interface (2)
HC3. Human-centered software evaluation
HC4. Human-centered software development
HC5. Graphical user-interface design
HC6. Graphical user-interface programming
HC7. HCI aspects of multimedia systems
HC8. HCI aspects of collaboration and communication

GV. Graphics and Visual Computing (3 core hours)
GV1. Fundamental techniques in graphics (2)
GV2. Graphic systems (1)
GV3. Graphic communication
GV4. Geometric modeling
GV5. Basic rendering
GV6. Advanced rendering
GV7. Advanced techniques
GV8. Computer animation
GV9. Visualization
GV10. Virtual reality
GV11. Computer vision

IS. Intelligent Systems (10 core hours)
IS1. Fundamental issues in intelligent systems (1)
IS2. Search and constraint satisfaction (5)
IS3. Knowledge representation and reasoning (4)
IS4. Advanced search
IS5. Advanced knowledge representation and reasoning
IS6. Agents
IS7. Natural language processing
IS8. Machine learning and neural networks
IS9. AI planning systems
IS10. Robotics

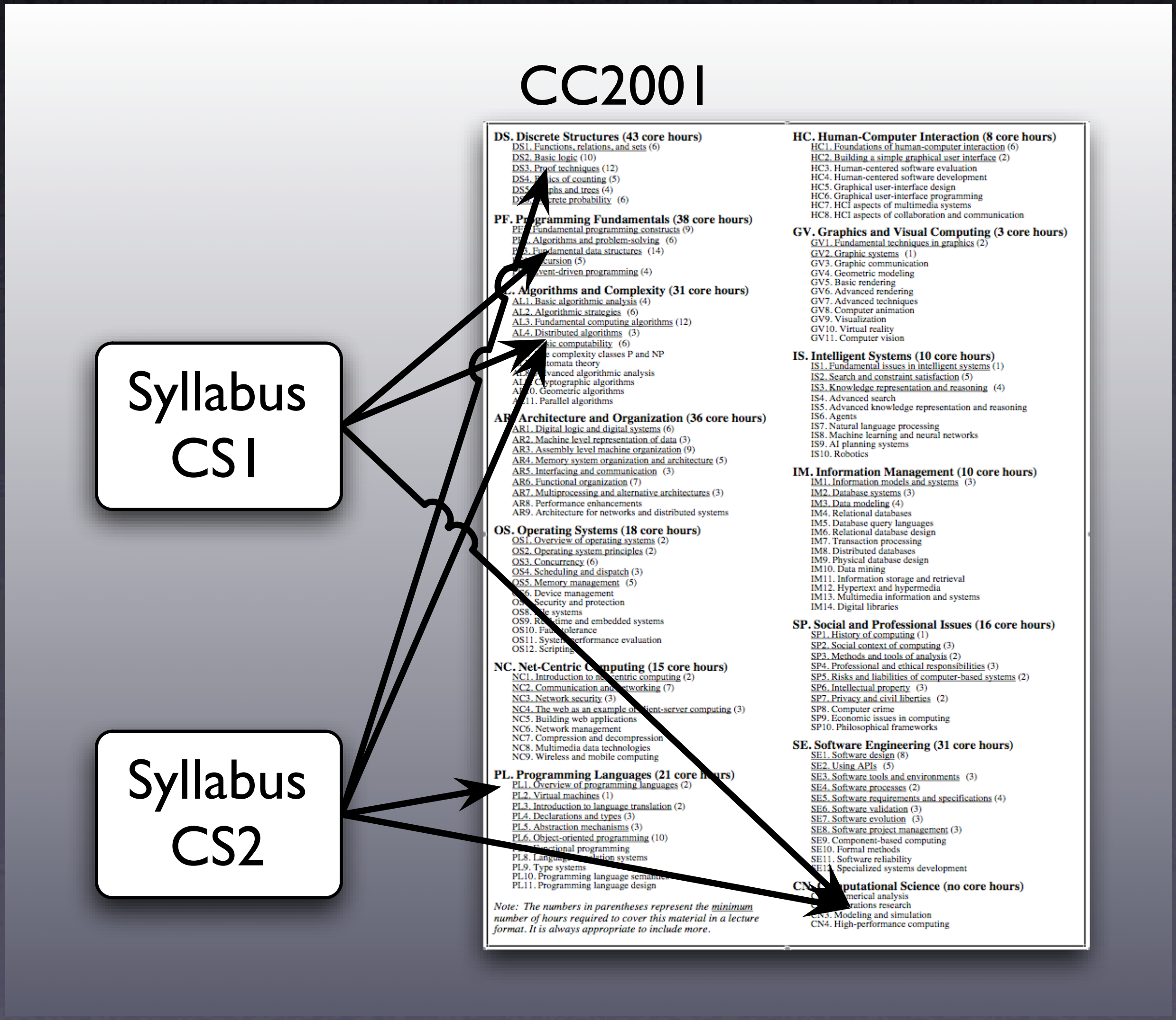
IM. Information Management (10 core hours)
IM1. Information models and systems (3)
IM2. Database systems (3)
IM3. Data modeling (4)
IM4. Relational databases
IM5. Database query languages
IM6. Relational database design
IM7. Transaction processing
IM8. Distributed databases
IM9. Physical database design
IM10. Data mining
IM11. Information storage and retrieval
IM12. Hypertext and hypermedia
IM13. Multimedia information and systems
IM14. Digital libraries

SP. Social and Professional Issues (16 core hours)
SP1. History of computing (1)
SP2. Social context of computing (3)
SP3. Methods and tools of analysis (2)
SP4. Professional and ethical responsibilities (3)
SP5. Risks and liabilities of computer-based systems (2)
SP6. Intellectual property (3)
SP7. Privacy and civil liberties (2)
SP8. Computer crime
SP9. Economic issues in computing
SP10. Philosophical frameworks

SE. Software Engineering (31 core hours)
SE1. Software design (8)
SE2. Using APIs (5)
SE3. Software tools and environments (3)
SE4. Software processes (2)
SE5. Software requirements and specifications (4)
SE6. Software validation (3)
SE7. Software evolution (3)
SE8. Software project management (3)
SE9. Component-based computing
SE10. Formal methods
SE11. Software reliability
SE12. Specialized systems development

CN. Computational Science (no core hours)
CN1. Numerical analysis
CN2. Operations research
CN3. Modeling and simulation
CN4. High-performance computing

Theory Curriculum



DS. Discrete Structures (43 core hours)
DS1. Functions, relations, and sets (6)
DS2. Basic logic (10)
DS3. Proof techniques (12)
DS4. Basics of counting (5)
DS5. Graphs and trees (4)
DS6. Discrete probability (6)

PF. Programming Fundamentals (38 core hours)
PF1. Fundamental programming constructs (9)
PF2. Algorithms and problem-solving (6)
PF3. Fundamental data structures (14)
PF4. Recursion (5)
PF5. Event-driven programming (4)

AL. Algorithms and Complexity (31 core hours)
AL1. Basic algorithmic analysis (4)
AL2. Algorithmic strategies (6)
AL3. Fundamental computing algorithms (12)
AL4. Distributed algorithms (3)
AL5. Basic computability (6)
AL6. The complexity classes P and NP
AL7. Automata theory
AL8. Advanced algorithmic analysis
AL9. Cryptographic algorithms
AL10. Geometric algorithms
AL11. Parallel algorithms

AR. Architecture and Organization (36 core hours)
AR1. Digital logic and digital systems (6)
AR2. Machine level representation of data (3)
AR3. Assembly level machine organization (9)
AR4. Memory system organization and architecture (5)
AR5. Interfacing and communication (3)
AR6. Functional organization (7)
AR7. Multiprocessing and alternative architectures (3)
AR8. Performance enhancements
AR9. Architecture for networks and distributed systems

OS. Operating Systems (18 core hours)
OS1. Overview of operating systems (2)
OS2. Operating system principles (2)
OS3. Concurrency (6)
OS4. Scheduling and dispatch (3)
OS5. Memory management (5)
OS6. Device management
OS7. Security and protection
OS8. File systems
OS9. Real-time and embedded systems
OS10. Fault tolerance
OS11. System performance evaluation
OS12. Scripting

NC. Net-Centric Computing (15 core hours)
NC1. Introduction to net-centric computing (2)
NC2. Communication and networking (7)
NC3. Network security (3)
NC4. The web as an example of client-server computing (3)
NC5. Building web applications
NC6. Network management
NC7. Compression and decompression
NC8. Multimedia data technologies
NC9. Wireless and mobile computing

PL. Programming Languages (21 core hours)
PL1. Overview of programming languages (2)
PL2. Virtual machines (1)
PL3. Introduction to language translation (2)
PL4. Declarations and types (3)
PL5. Abstraction mechanisms (3)
PL6. Object-oriented programming (10)
PL7. Functional programming
PL8. Language translation systems
PL9. Type systems
PL10. Programming language semantics
PL11. Programming language design

Note: The numbers in parentheses represent the minimum number of hours required to cover this material in a lecture format. It is always appropriate to include more.

HC. Human-Computer Interaction (8 core hours)
HC1. Foundations of human-computer interaction (6)
HC2. Building a simple graphical user interface (2)
HC3. Human-centered software evaluation
HC4. Human-centered software development
HC5. Graphical user-interface design
HC6. Graphical user-interface programming
HC7. HCI aspects of multimedia systems
HC8. HCI aspects of collaboration and communication

GV. Graphics and Visual Computing (3 core hours)
GV1. Fundamental techniques in graphics (2)
GV2. Graphic systems (1)
GV3. Graphic communication
GV4. Geometric modeling
GV5. Basic rendering
GV6. Advanced rendering
GV7. Advanced techniques
GV8. Computer animation
GV9. Visualization
GV10. Virtual reality
GV11. Computer vision

IS. Intelligent Systems (10 core hours)
IS1. Fundamental issues in intelligent systems (1)
IS2. Search and constraint satisfaction (5)
IS3. Knowledge representation and reasoning (4)
IS4. Advanced search
IS5. Advanced knowledge representation and reasoning
IS6. Agents
IS7. Natural language processing
IS8. Machine learning and neural networks
IS9. AI planning systems
IS10. Robotics

IM. Information Management (10 core hours)
IM1. Information models and systems (3)
IM2. Database systems (3)
IM3. Data modeling (4)
IM4. Relational databases
IM5. Database query languages
IM6. Relational database design
IM7. Transaction processing
IM8. Distributed databases
IM9. Physical database design
IM10. Data mining
IM11. Information storage and retrieval
IM12. Hypertext and hypermedia
IM13. Multimedia information and systems
IM14. Digital libraries

SP. Social and Professional Issues (16 core hours)
SP1. History of computing (1)
SP2. Social context of computing (3)
SP3. Methods and tools of analysis (2)
SP4. Professional and ethical responsibilities (3)
SP5. Risks and liabilities of computer-based systems (2)
SP6. Intellectual property (3)
SP7. Privacy and civil liberties (2)
SP8. Computer crime
SP9. Economic issues in computing
SP10. Philosophical frameworks

SE. Software Engineering (31 core hours)
SE1. Software design (8)
SE2. Using APIs (5)
SE3. Software tools and environments (3)
SE4. Software processes (2)
SE5. Software requirements and specifications (4)
SE6. Software validation (3)
SE7. Software evolution (3)
SE8. Software project management (3)
SE9. Component-based computing
SE10. Formal methods
SE11. Software reliability
SE12. Specialized systems development

CN. Computational Science (no core hours)
CN1. Numerical analysis
CN2. Operations research
CN3. Modeling and simulation
CN4. High-performance computing

Applications

- Personalizing NSDL* content for students
- Assisting instructors creating new syllabi
- Syllabi overview for students
- Assisting curriculum design & accreditation
- Comparing programs at various schools
- Mobile access to educational resources

* National Science, Technology, Engineering and Mathematics Education Digital Library.

Conclusion

- Leverage information already available in syllabi
- Create a standardized representation
- Obtain existing syllabi and create a repository
- Build tools and applications to enable the use of syllabi beyond their current use

Questions