APPENDIX B
Knowledge Areas of the IT Field

This appendix contains the knowledge areas of the IT field. They are grouped according to the degree programs. Each knowledge area is subdivided into parts for Theory, Abstraction, Design, and Technology.

The accompanying Cross-Reference Matrix shows the page numbers of each knowledge area and how the individual degree programs draw upon the knowledge areas.
COMPUTER SCIENCE KNOWLEDGE AREAS (CS)

Top Level Areas

DISCRETE STRUCTURES
ALGORITHMS, DATA STRUCTURES, AND COMPLEXITY
PROGRAMMING
PROGRAMMING LANGUAGES & COMPILERS
ARCHITECTURE
OPERATING SYSTEMS
NETWORKS AND DISTRIBUTED COMPUTING
DATA MANAGEMENT
SOFTWARE ENGINEERING AND METHODOLOGY
INTELLIGENT SYSTEMS
GRAPHICS AND MULTIMEDIA
HUMAN-COMPUTER INTERACTION
COMPUTATIONAL SCIENCE
SYSTEM MEASUREMENT AND CAPACITY PLANNING
SOCIAL, ETHICAL, AND PROFESSIONAL ISSUES

Matrix Rows

DISCRETE STRUCTURES
Computing machines operate in discrete state spaces. Most physical phenomena are continuous. Discrete structures are models of computational spaces that help control errors when approximating physical processes and prove useful properties of algorithms.

Theory
Functions, relations, and sets
Basic logic
Proof methods
Basics of counting
Graphs and trees
Fractals, Mandelbrot sets, strange attractors, chaos
Digital representation and processing of signals

Abstraction
Flow networks
Computational grids
Finite element methods
Algorithms for computing fractals, strange attractors

Design
Variable density grids
Finite element data structures
Representing graphs for computations
Classic algorithms of discrete structures

Technology
Applications of discrete analysis in various domains
ALGORITHMS, DATA STRUCTURES, AND COMPLEXITY

Every computing machine is controlled by algorithms, which are mechanical procedures of machine-implementable instructions. The efficiency of algorithms depends on the organization and structure of the data.

Theory
- Automata theory
- Computability theory
- Complexity theory (notation; classes including P and NP)
- Concurrency theory
- Probabilistic algorithm theory
- Database theory
- Randomized algorithms
- Pattern-matching algorithms
- Graph and network algorithms
- Algebraic algorithms
- Combinatorial optimization
- Cryptography
- Proving algorithm correctness

Abstraction
- Object oriented paradigms
- Experimental studies of algorithms
- Stress testing of algorithms
- Heuristic algorithm testing
- Divide and conquer algorithms
- Greedy algorithms
- Dynamic programming
- Finite state machine interpreters
- Stack machine interpreters
- Heuristic searches
- Randomized algorithms
- Genetic algorithms
- Evolutionary algorithms
- Geometric algorithms

Design
- Top 10 algorithms of 20th century
- RSA public key system
- VLSI circuit layout and simulation systems
- Design methods such as predicate transformers
- Object oriented design

Technology
- Libraries of algorithms (e.g., mathematical, statistics)
- Object oriented design tools
PROGRAMMING
Programming is the art of designing working programs that use appropriate algorithms and data structures to solve problems. It is a central practice of information technology.

Theory
- Notations for predicates and invariants
- Proof methods for programs
- Mapping program text to execution dynamics

Abstraction
- Algorithms and problem-solving
- Basic programming constructs
- Basic data structures
- Recursion
- Abstract data types
- Object-oriented programming
- Event drive and concurrent programming
- Functional programming
- Logic programming

Design
- Tools to assist programming
- version control
- syntax directed editors
- profilers

Technology
- Modern APIs
- Class libraries
- Function libraries
- Extant programming languages

PROGRAMMING LANGUAGES & COMPILERS
This area deals with notations for virtual machines that execute algorithms and with notations for algorithms and data; the sets of strings of symbols that are generated by such notations are called languages. It also deals with efficient translations from high-level languages into machine codes.

Theory
- Acceptors, transformers, and generators (e.g.,
  - Turing machines, Post systems, Lambda-calculus,
  - Pi-calculus, Propositional logic)
- Semantics
- Types
- Code optimization

Abstraction
- Language classes (e.g., static typing, dynamic typing, functional, procedural, object-oriented, logic, message-passing, dataflow)
- Application classes (e.g., business, commerce, data processing simulation, graphics)
- Implementation models (run-time environments, e.g., imperative object-oriented, logic, constraint, concurrent, distributed)
- Type systems
Design
Declarations, modularity, and storage management
Methods of implementing models (e.g., static execution, dynamic execution, storage and register allocation, compilers, cross-compilers, interpreters, parallelism-finders).
Methods of debugging and testing
Compiler generators
Production-quality compilers
Plug-ins and add-ons (e.g., equations for documents)

Technology
Programming languages (e.g., C, C++, Pascal, Fortran, Lisp)
Compiler generators (e.g., YACC, LEX)
Syntax directed editors

ARCHITECTURE
This area deals with methods of organizing hardware (and associated software) into efficient, reliable systems.

Theory
Digital logic and digital systems
Boolean algebra
Coding theory
Finite-state machine theory
Finite arithmetic
Error propagation

Abstraction
Finite-state machine models
Machine level representation of data
Assembly-level machine organization
Optimal instruction sets for given workloads
Hardware reliability
VLSI space-time tradeoffs
Information hiding
Levels of abstraction
Simulation of machines
Memory system organization
Shared memory models
I/O and communication
CPU implementation
Cache optimization strategies
Bus sharing strategies
RISC versus CISC
SIMD and MIMD models
Interrupt systems
Procedure call systems

Design
Machine types (e.g., von Neumann, functional, dataflow, pipeline, hypercube, vector, supercomputer)
Parts-based system synthesesation
Methods to optimize instruction sets
Function units
Memory
I/O (A/D and D/A converters, output transducers, DMA)
Attachment cards (e.g., graphics, network, ethernet)
VLIW
CAD methods for integrated circuits
Logic simulators

Technology
Intel chips (x86)
Motorola chips
Chip foundries
Silicon compilers
Case studies of machines of each manufacturer

OPERATING SYSTEMS
This area deals with control mechanisms that allow multiple resources to be efficiently coordinated in computations distributed over many computer systems connected by local and wide-area networks.

Theory
Concurrency theory (synchronization, determinacy, and deadlocks)
Scheduling theory
Program behavior and memory management theory
Network flow theory
Performance modeling and analysis
Cryptographic protocols

Abstraction
Abstraction and information-hiding principles
Naming and binding
Binding user-defined objects to internal computational structures
Interrupt systems
Procedure call models
Process and thread management
Memory management
Job scheduling
Secondary storage and file management
Device management
Files and directories
Shells and GUIs
Performance analysis
Experimental validation of performance models
Distributed computation
Remote procedure calls
Real-time systems
Access and flow models
Secure computing

Design
Time sharing systems
Automatic storage allocators
Multilevel schedulers
Memory managers
Hierarchical file systems
Technology
   Existing systems (e.g., Unix, MS/DOS, Windows, Mach, MacOS)
   Utilities (e.g., editors, document formatters, compilers, linkers, device drivers)
   Queueing network modeling packages

NETWORKS AND DISTRIBUTED COMPUTING
This area deals with the architecture of communication networks used for data communication, distributed computing, and collaboration; and with their use in computing and communicating.

Theory
   Network flow theory
   Spanning trees
   Network queueing and congestion theory
   Network-based cryptographic protocol proofs

Abstraction
   Layered protocols
   Internet protocols
   Cryptographic protocols
   Naming
   Remote resource usage
   Help services
   Dynamic routing protocols
   Local network routing protocols (e.g., token-passing and shared buses)
   GPS (global positioning systems)
   Client-server computing
   The Web
   Distributed object systems
   Distributed operating systems
   Distributed systems
   Checkpoint and recovery

Design
   Network architectures (e.g., Ethernet, FDDI, token ring, ATM, Frame Relay)
   Internet protocols (e.g., TCP/IP, HTTP, HTTPS, SSL)
   Conferencing protocols
   Security protocols
   Client-server protocols
   Authentication protocols

Technology
   Specific protocol implementations
   GPS
DATA MANAGEMENT
This area deals with the organization of large sets of persistent, shared data for efficient query and update.

Theory
- Relational algebra
- Relational calculus
- Concurrency theory
- Serializable transactions
- Deadlock prevention
- Synchronized updates
- Statistical inference
- Rule-based inference
- Sorting
- Searching
- Indexing
- Cryptographic sealing and authentication

Abstraction
- Data models (e.g., object based, record based, object-relational)
- Storing files for fast retrieval
- Access methods
- Query optimization
- Concurrency control and recovery
- Integrity
- Security and privacy
- Virtual machine interpreters for query languages
- Hypertext and multimedia integration
- Archiving and media-migration methods

Design
- Database architectures (e.g., relational, hierarchical, network, distributed, and retrieval systems)
- Multilevel secure database systems

Technology
- Commercial database systems (e.g., Ingres, Oracle, System R, dBase, Sybase)
- Commercial retrieval systems (e.g., Lexis, Osiris, Medline)
- Commercial hypertext systems (e.g., NLS, NoteCards, HyperCard, SuperCard, Intermedia, Xanadu)
SOFTWARE ENGINEERING AND METHODOLOGY
This area deals with the design of programs and large software systems that meet specifications and are safe, secure, reliable, and dependable.

Theory
  Program verification and proof
  Temporal logic
  Reliability theory

Abstraction
  I/O specifications of systems
  Software reliability
  Measurement and evaluation of programs
  Software tools and environments
  Software testing and maintenance
  Megaprogramming (aka programming in the large)

Design
  Software project management
  Software construction processes
  Software systems development processes
  Matching software systems with machine architectures
  Quality assurance
  Secure computing

Technology
  Specification languages
  Version tracking systems
  Syntax directed editors
  Tools (e.g., program development, measurement, profiling
text formatting, debugging, CASE)

INTELLIGENT SYSTEMS
This area deals with the modeling of animal and human cognition, with the ultimate intention of building machine components that mimic or augment them.

Theory
  Logic systems for reasoning (e.g., first order logic, fuzzy
  logic, temporal logic, probabilistic logic, deduction,
  induction)
  Knowledge representation
  Formal models for knowledge representation
  Search methods (e.g., branch-and-bound, alpha-beta, tree pruning
  genetic algorithms)
  Theories of learning
  Neural networks
  Pattern recognition
  Computer vision
  Speech recognition and understanding
  Natural language translation
  Robot systems
Abstraction
Knowledge representation models
Problem-solving models
Search heuristics
Learning models
Agents
Language understanding models (natural language processing)
Speech models
Pattern recognition models
Vision models
Neural network models
Genetic algorithms
Models of human memory
Knowledge robots (e.g., "knowbots" or "bots")

Design
Logic programming
Expert systems
Knowledge engineering environments
Natural-language problem-solving systems
Games of strategy (esp., chess, checkers)
Neural network models
Fuzzy logic modules
Speech synthesis and recognition
Robots

Technology
Lisp, Prolog
Dendral, etc.
Margie, SHRDLU
Chess playing programs
Music composition programs (e.g., EMI)
Dragon systems, ViaVoice
Genetic algorithms

GRAPHICS AND MULTIMEDIA
Graphics is concerned with processes for representing physical and conceptual objects and their motions visually on a 2D computer screen or in a 3D hologram.

Theory
Computational geometry
Projecting 2D images of 3D objects
Chaos theory
Fractal theory
Graphics theory
Sampling theory
Color theory

Abstraction
Drawing algorithms (e.g., rendering, smoothing, shading, hidden line removal, ray tracing, hidden surfaces, translucent surfaces, shadows, lighting, edges, color maps, splines, textures, antialiasing, fractals, animation, object hierarchies)
Geometric modeling
Visualization
Animation
Virtual reality
Interactive simulation
Cognitive psychology
Medical imaging
Multimedia objects
Compression and decompression algorithms

Design
Graphics libraries
Video editors
Scientific visualization
Color models
Web pages
Multimedia applications
Content authoring
Multimedia servers and file systems
Streaming video

Technology
Graphics libraries
Graphics standards
Printer languages (PostScript, PDF)
CAD systems
Medical imaging systems
Multimedia data technologies
Data compression and storage standards (e.g., MP3, PICT, JPEG, GIF)

HUMAN-COMPUTER INTERACTION
HCI is concerned with the efficient coordination of action and transfer of information between humans and machines via various human-like sensors and motors, and with information structures that reflect human conceptualizations.

Theory
Cognitive psychology

Abstraction
Modeling the user
Interaction
Analysis of risks to humans
Models of collaborative work
Business processes
Event-driven interface model
Desktop metaphor

Design
Human factors
Window management
Desktop metaphors
Other possible interface metaphors
Help systems
Displays with high interpretation accuracy
Usability engineering
Collaborative work systems
Technology
A/D and D/A converters
Output transducers
WIMMP user interfaces
Pen-based user interfaces
Flight simulators
Distributed interactive simulation (DIS)

COMPUTATIONAL SCIENCE
This area deals with explorations in science and engineering that cannot proceed without high-performance computation and communications.

Theory
Number theory
Linear algebra
Symbolic computation
Discipline dependent mathematical models

Abstraction
Discrete approximations
Backward error propagation and stability
Fast Fourier Transform
Poisson Solvers
Finite element models
Iterative methods and convergence
Parallel algorithms and architectures
Automatic grid generation and refinement
Scientific visualization
Symbolic integration and differentiation
Hypercubes and grids

Design
Scientific function packages (domain dependent)
Grand challenge problems
Programming for parallel architectures

Technology
Hypercubes
Chem, Web, Linpack, Elpack, Ellpack, Macsyma, Mathematica, Maple, and Reduce
Solutions to grand challenge problems

SYSTEM MEASUREMENT AND CAPACITY PLANNING
This area deals with the performance analysis of systems, especially for throughput and response time; with the calculation of resource capacities needed to meet stated performance objectives; and with forecasts of performance.

Theory
Queueing theory
Operational analysis
Transform analysis
Mean Value analysis
Convolution analysis
Abstraction
Queueing network models
Computational algorithms
Approximation algorithms
Workload models
Customer models
Extracting models from event traces
Benchmarking
Simulation

Design
Capacity planning
Workload characterization

Technology
Best/1, QN solvers

SOCIAL, ETHICAL, AND PROFESSIONAL ISSUES
The human context in which professionals work. Professional responsibility and conduct. (TADT does not work in this case.)

History of computing
Social contexts of computing
Methods and tools of analysis
Professional and ethical responsibilities
Risks and liabilities of safety-critical systems
Intellectual property
Privacy and civil liberties
Social implications of Internet
Computer Crime
Information warfare
Economic issues in computing
Philosophical foundations of ethics
COMPUTER SYSTEMS ENGINEERING KNOWLEDGE AREAS (CSE)

Top Level Areas

ELECTRONICS
DIGITAL DEVICE TECHNOLOGY
DIGITAL DESIGN
SIGNALS AND SYSTEMS
COMPUTER ARCHITECTURE (See CS - Architecture)
COMPUTER INTERFACING (See CS - I/O Architecture, HCI)

Matrix Rows

ELECTRONICS
This area deals with the basic elements of electrical circuits including resistors, capacitors, inductors, and active devices such as diodes and transistors. Analog and digital circuits consist of these elements along with appropriate voltage and current power supplies.

Theory
Ohm's law
Kirchoff laws for voltage and current summation
Power laws
Energy conservation laws
Series and parallel laws for components
Time varying currents and impedance
Instrumentation
Frequency domain representation of circuits

Abstraction
Linear circuits model
First order and second order circuits
Nonlinear elements
Frequency response and amplifier gain curves
Idealization of components

Design
Networks of active and passive components
Time varying functions and response

Technology
Laboratory measurements
DIGITAL DEVICE TECHNOLOGY
This area deals with basic physics and technology underlying the fabrication of electronic digital circuits; with physical concepts in semiconductor devices; with design of logic devices; and with different families of logic and their characteristics.

Theory
Semiconductor physics
p-n junctions
Metal-Oxide-Semiconductor Field Effect Transistor (MOSFET)
Complimentary MOSFET (CMOS)
CMOS logic

Abstraction
Transistors as switches
Idealized models of CMOS technology
Threshold devices

Design
Pull-up and pull-down transistor design for logic gates
Common building blocks (e.g., transmission gates)

Technology
Physical level layout tools (Magic, L-EDIT)

DIGITAL DESIGN
This area deals with designing digital systems for implementing combinational and sequential operations; with arithmetic and logic unit of a computer, shift registers, state machine circuits; with structure and utilization of programmable logic structure; and with use of computer-aided design and optimization tools.

Theory
Logical minimization techniques
Finite state machines
Time dependencies
Gate arrays and programmability
Signal processing functionalities
Karnaugh maps
Turing machines
Race conditions
Arbitration problem

Abstraction
Boolean algebra
State transition diagrams
Synchronous circuits
Asynchronous circuits
Pipelined circuits
VLSI circuits
Wafer-scale circuits
Design
  Register transfer models
  Behavioral models
  Structural models
  Schematic capture
  Netlist

Technology
  Gate arrays
  CAD tools
  Different hardware description languages
  Circuit simulators

**SIGNALS AND SYSTEMS**

This area deals with analog and digital representation of signals; with frequency representation of analog signals; with sampling and quantization; with finite impulse response (FIR) and infinite impulse response (IIR) filtering; with digital signal processing; and with hardware implementation of DSP functional blocks.

Theory
  Sinusoidal signals
  Linear shift invariant systems
  Sampling theorem
  Finite word length effects
  Fourier and Laplace Transforms
  Fast Fourier transform
  Digital signal processing (DSP) algorithms
  Shannon communication theory
  Optimal message encoding
  Signal encryption
  Random processes and noise
  Error detection and correction
  Encoding digital signals for analog lines

Abstraction
  Logic circuits and Boolean expressions
  Discrete processes
  Frequency analysis
  Encryption methods and models
  Algorithms for FFT, error detection, correction
  Modems, facsimile systems

Design
  Time domain filtering
  Frequency domain filtering
  Stability analysis
  DSP chips

Technology
  Special purpose DSP chips
  Matlab signal processing software
COMPUTER ARCHITECTURE (See CS - Architecture)

COMPUTER INTERFACING (See CS - I/O Architecture, HCI)
SOFTWARE ENGINEERING KNOWLEDGE AREAS (SWE)

Top Level Areas

COMPUTING SYSTEMS
SOFTWARE REQUIREMENTS AND SPECIFICATIONS
SOFTWARE DESIGN
SOFTWARE CONSTRUCTION
USER INTERFACE DESIGN AND DEVELOPMENT
TEST, EVALUATION, AND MEASUREMENT
PROJECT MANAGEMENT
CRITICAL SOFTWARE SYSTEMS

Matrix Topics

COMPUTING SYSTEMS
This area deals with methods of organizing hardware, software mechanism for controlling hardware and software resources, and translations from high-level programming languages into executable versions.

Theory
  Acceptors and generators
  Language semantics
  Data typing
  Digital logic and digital series
  Boolean algebra
  Concurrency theory
  Scheduling theory

Abstraction
  Language classes
  Type systems
  Finite-state machine models
  Assembly language
  Levels of hardware abstraction
  Naming and binding
  Job, process and memory management
  Distributed computation

Design
  Declarations and storage
  Compiler generators
  Machine type
  Machine hardware component connections
  Time sharing systems
  Memory managers

Technology
  Programming languages
  Assembly language
  Current operating systems (Unix, Windows)
  Utilities
SOFTWARE REQUIREMENTS AND SPECIFICATIONS
Software must be developed that satisfies rigorous requirements and follows well defined mathematical principles. This area deals with descriptions of the software's expected behavior, both functional and non-functional, and formal and informal. It covers eliciting functional requirements from users and stating them in forms that are useful for designers. This basis allows software engineers to create formal models of software, leading to higher quality products.

Theory
- Logic
- Functions, relations and sets
- Proof methods
- Mathematical structures (lists, trees, graphs)
- Countability
- Statistical properties
- Psychology
- Formal descriptions of software
- Safety and security

Abstraction
- Properties of software: functional behavior, reliability, usability, maintainability, safety, security, availability, conformance to standards, ...
- Object-oriented techniques
- Algorithms and strategies
- Design, specifications, and requirements
- User-level versus engineer-level requirements

Design
- Usefulness of software requirements
- Documentation
  - Mathematical modeling
  - Well known algorithms

Technology
- Measurements of the mean
- Modeling tools
- Formal specification languages
- Model checking and formal verification

SOFTWARE DESIGN
Software design produces a solution for software requirements that will solve the software-related aspects of the software in some understandable description. It covers high level, intermediate level, and low level design. This is the major step from functional descriptions of "what" the software must do to process-oriented descriptions of "how" the software does it, and is thus the center of all software engineering activities.

Theory
- Problem solving
- General design concepts
- Key issues in software design
Abstraction
   Function-oriented design
   Data structure design
   Object-based design
   Architectural design

Design
   Integration
   Quality attributes and analysis

Technology
   Languages for expressing design
   Language tools
   Automatic code writers

SOFTWARE CONSTRUCTION
Software construction is the most concrete part of software engineering that produces the most recognizable product: the code. This area deals with advanced programming concepts and the ability to produce software that conforms to requirements and design. The emphasis is not just on "getting it to work", but on "producing a high quality product". While satisfying the requirements and design is important, reliability is only one of many quality attributes.

Theory
   Notations for invariants
   Finite state modeling
   Program logic
   Quality attributes: reliability, efficiency, maintainability, conformance.
   safety, security, usability, readability, etc.
   Handling complexity

Abstraction
   Algorithms and problem-solving
   Advanced data structures
   Programming abstraction methods: control, recursion, data structures, inheritance, polymorphism, dynamic binding, concurrency
   Programming for testability and maintainability

Design
   Appropriate use of language features
   Version control
   Profilers and debuggers

Technology
   Programming languages
   Program editors and compiling tools
   Debuggers
   Low-level test tools
USER INTERFACE DESIGN AND DEVELOPMENT
This area deals with the goals and techniques for developing software that interacts directly with the user. Design of GUIs requires careful attention to the interaction model of the software and the user. A variety of important methods and technologies assist in this process.

Theory
- Psychology of human computer interaction
- Limits of human comprehension
- Evaluation and measurement of user interfaces

Abstraction
- Models of human comprehension
- Models of software execution

Design
- Command languages
- Menus and forms
- Graphical user interfaces
- Web-based interfaces

Technology
- Parsing and translation
- GUI development tools
- Programming languages for UI development
- Web software development

TEST, EVALUATION, AND MEASUREMENT
This area deals with evaluating the quality of software products. It includes execution-based testing, inspections and static evaluation, and measurement of various quality aspects. The evaluation should be done on code, requirements, designs, user interfaces, and all documentation. This is by far the most mathematical area in software engineering.

Theory
- Definitions of testing
- Test criteria, acceptors, generators, and analyzers
- Infeasibility and theoretical limitations
- Analysis of software
- Metrics for software products

Abstraction
- Graphical representations of software: control flow, data flow, program dependence, class graphs, inheritance graphs, etc.
- Regression testing
- Test levels: Unit and module, integration, system, acceptance
- Functional and non-functional testing (conformance, usability)

Design
- Inspection processes
- Test requirements
- Test case generation and measurement
- Test techniques
- Quality assurance
Technology
Work flow tools
Test tools: Generators, coverage analyzers, driver generators, regression test tools, test case database tools.

PROJECT MANAGEMENT
This is the area that most involves people. It addresses how to run a development process, interact with, evaluate, and motivate people. The software process and how to evaluate the software products is discussed. Techniques for integrating software and integrating the various software development activities are important for project managers.

Theory
Psychology of human interaction
Integration and coupling
Hierarchical and flow models of organizations
Cognitive and negotiating strategies
Decision theory

Abstraction
Inter-personal interaction
Leading and management
Management under uncertainty and change
Legal and ethical aspects

Design
Configuration and version control
Staffing and staff organization
Leadership and consensus building

Technology
Software for supporting communication and work organization

CRITICAL SOFTWARE SYSTEMS
This area deals with techniques for creating software that is used in applications where failure can result in loss of property, money, or life. Much of such software is concurrent, real-time, or distributed, and many such systems are extremely large. Critical software discusses the theory behind the applications, techniques for developing such software, and methods for evaluating it. This is one of the most rigorous areas.

Theory
Synchronization, determinacy, deadlocks
Scheduling theory
Communicating processes
Real-time constraints
Design redundancy
Data redundancy
Software safety
Formal methods
Abstraction
    Process management
    Job scheduling
    Communications among processes
    Remote procedure calls
    Decision trees

Design
    Client-server computing
    Finite-state modeling
    Modeling techniques for real-time systems
    Tolerance of software faults
    Design for testability
    Statistical testing

Technology
    Modeling languages
    Programming language facilities
    Packages for communication and process management
NETWORK ENGINEERING KNOWLEDGE AREAS (NE)

Top Level Areas

INFORMATION TECHNOLOGY
NETWORK PROTOCOLS AND SOFTWARE
DATA COMMUNICATIONS
ENCRIPTION AND COMPUTER/NETWORK SECURITY
NETWORK SYSTEM SOFTWARE
WIRELESS AND MOBILE COMMUNICATIONS
NETWORK MANAGEMENT

Matrix Rows

INFORMATION TECHNOLOGY
This area draws on basic technology and concepts from other areas, especially computer science and computer systems design. The principal knowledge areas are:

- Electronics (telecommunications systems are composed of these components)
- Digital systems (principles of digital components and logic)
- Operating systems (network protocols are managed by OSs)

NETWORK PROTOCOLS AND SOFTWARE
Modern networks are built around architectures that are implemented as a set layers called the protocol software stack. These protocols are used not only for data communication but for many aspects of distributed computing and commerce.

- Theory
  - Network flow theory
  - Spanning trees
  - Network queueing and congestion theory
  - Network-based cryptographic protocol proofs

- Abstraction
  - Layered protocols
  - Internet protocols
  - Naming
  - Remote resource usage
  - Help services
  - Dynamic routing protocols
  - Local network routing protocols (e.g., token-passing and shared buses)

- Design
  - Network architectures (e.g., Ethernet, FDDI, token ring, ATM, Frame Relay)
  - Internet protocols (e.g., TCP/IP, HTTP, HTTPS, SSL)
  - Conferencing protocols
  - Security protocols
DATA COMMUNICATIONS
This area deals with the encoding of data and information into digital signals, the transmission of those signals through various (noisy) media, and the reconstruction of the data by the receiver.

Theory
  Coding theory
  Information theory
  Error correction and reliability
  Signal and file compression
  Photonics

Abstraction
  Protocol layering
  Link layer models
  Link layer routing models
  Transport layer models
  Application layer models
  Electrical and optical signaling

Design
  Effective link performance
  Duplex channels
  Repeaters and bridges
  Shared medium access alternatives

Technology
  Physical layer encodings
  Link-layer protocols
  MAC sublayer protocols
  Bridges

ENCRYPTION AND COMPUTER/NETWORK SECURITY
This area deals with the cryptographic encoding of signals so that they can be kept secret or authenticated; protocols that use cryptography to achieve coordinated action such as agreement between sender and receiver on a session key; and other methods for controlling access to objects and information within a network.

Theory
  Security principles
  Public key cryptosystems
  Private key cryptosystems
  Key distribution principles
  Access and flow models
  Proof methods for secure systems
  Incomputability of most security properties
  Types of attacks and risks of compromise

Abstraction
  Security architectures
Protocol for agreeing on a session key
Protocol for signing a document
Protocol for authenticating a user or machine
Protocol for key distribution
Notaries and certificates
Viruses

Design
Selecting effective encryption in network design
Building real protocol stacks

Technology
Cryptographic techniques (e.g., secret key, RSA, DES)
Cryptographic chipsets
Cryptographic software (e.g., PGP)
Cyberlocator (GPS-based authentication)
Virus and worm detection and eradication
Kerberos

NETWORK SYSTEM SOFTWARE
Distributed and networked systems rely on specialized software that is not designed by the same principles as applications software.

Theory
Protocol layering
Functions of network software components
Connectivity models (and routing algorithms)
Naming models (e.g., domains)
Performance and congestion models of networks

Abstraction
Network architectures
Naming services
Client/server architectures

Design
Protocol design
Server software design
Performance versus reliability tradeoffs
Effective combination of network components

Technology
Internet protocol families
Other open protocols (OSI, IEEE)
Proprietary commercial software and protocols (e.g. Cisco, Novell)
Multimedia networking
**WIRELESS AND MOBILE COMMUNICATIONS**

This area deals with the protocols and signaling methods used to provide networking via radio communications to mobile units.

**Theory**
- Wireless transmission theory
- Cells and packet radio
- Performance and congestion models of networks

**Abstraction**
- Signaling systems for mobile telephony (e.g., CDMA)
- Dynamic roaming
- Architectures for wireless/mobile access

**Design**
- Integrating wireless/mobile technologies in networks
- Selecting wireless technologies to meet network requirements

**Technology**
- Digital satellite links
- Cellular network standards
- Infrared and microcell techniques
- IEEE wireless LAN protocols

**NETWORK MANAGEMENT**

This area deals with the methods of managing routing and routers so as to achieve performance and reliability goals.

**Theory**
- Monitoring
- Control
- Network cost models
- Network performance models

**Abstraction**
- Management by exception
- Traffic loading and flow

**Design**
- Strategies for optimizing cost
- Strategies for optimizing performance
- Cost/performance tradeoffs

**Technology**
- OSI CMIP protocols
- IETF SNMP protocols
- Network status display software
INFORMATION SYSTEMS KNOWLEDGE AREAS (IS)

Top Level Areas

INFORMATION TECHNOLOGY
ORGANIZATIONAL AND MANAGEMENT CONCEPTS
SYSTEM DEVELOPMENT
MANAGEMENT INFORMATION SYSTEMS

Matrix Rows

INFORMATION TECHNOLOGY
Deals with basic elements of information technology from the perspective of how they process, store, retrieve, and present information for human uses. These elements are primary knowledge areas of other specialties of IT, notably Computer Science.

Theory
Basic concepts and mathematics associated with the main areas listed under Abstraction

Abstraction
Computer architectures
Algorithms and data structures
Programming languages
Operating systems
Telecommunications
Database
Artificial intelligence
Human computer interaction

Design
Design principles associated with the main areas listed under Abstraction

Technology
Technologies deployed in the marketplace in the main areas listed under Abstraction

ORGANIZATIONAL AND MANAGEMENT CONCEPTS
Deals with the theory and practice of organizations considered as networks of commitments supported by information systems.

Theory
Hierarchical and flow models of organizations
Organizational work groups
Organizational span: user, group, team, enterprise, world
Strategic, tactical, operational roles of IS in enterprise
Interaction between IS and organizational structures of centralized, decentralized, matrix
Issues of software use in organizations
Decision theory
Organizational behavior
Job design theory
Cognitive styles
Negotiating styles

Abstraction
Information systems management
Computer operations management
Managing IS as a business
Performance evaluation of IS as service function
Decisions under uncertainty
Cost and value of information, competitive value
Group decision processes
Change management
Legal and ethical aspects of IS
Professionalism

Design
IS planning
Staffing and HR management
CIO and staff functions
Backup, disaster planning, recovery
Managing emerging technologies
Security and control, viruses, system integrity
Group dynamics, teamwork, leadership
Power and politics
Consensus building
Personal and interpersonal skills

Technology
Specific systems for supporting the above

SYSTEM DEVELOPMENT
Deals with the engineering and management issues of designing, building, testing, operating, and maintaining information systems in organizations.

Theory
Systems and information concepts
Information theory
System control (feedback, loops, measurement, quality)
System development lifecycle models

Abstraction
Measures and metrics of lifecycle models
Organizational and software process modeling
Data modeling (entity-relationship, normalization)
Data, process, behavior, object oriented methodologies
Risks and risk management
Projects and project management
Information and business analysis
Strategies for testing and implementation
Systems operation and maintenance
Systems development for standard types of systems (e.g., transaction processing, MIS, group support, decision
support, expert systems, executive support, office, collaborative, image, workflow, functional support, interorganizational)

Design
Evaluating and selecting a development approach
Software engineering process
Application planning (infrastructure, architecture, operations, measurement, bottlenecks, complexity
Planning for security, privacy, control of complexity
Information systems design (methods, creativity, cognitive styles, HCI, software development)

Technology
System development tools (CASE, JAD, IDEF, GDSS)
Software tools (data dictionary, repository, application generator, reuse, version tracking, program generators)
Risk assessment tools
Project management tools (PERT, Gantt, etc)

MANAGEMENT INFORMATION SYSTEMS
This area deals with management and organizational practice in the context of international information systems.

Theory
Economic theory for MIS (transaction and agency costs)
Behavioral theories (sociology, cultural, political)
Electronic markets
Interorganizational systems
Virtual organizations
Implicit and explicit knowledge
Return on investment analysis
Software ergonomics
Predictor variables for user satisfaction
International IT infrastructure
Cultural issues in IT diffusion
Transnational IT systems
Global business drivers and IT strategy
Legal (property rights, liability, accountability, due process)
Ethics (Kant, Descarte, etc)
Net present value
Profitability index
Cost benefit ratio

Abstraction
Information architecture
Change management
Managing dispersed teams
Value chain model
Competitive forces model
Knowledge markets and knowledge transfer
Return on investment models for cable modem, DSL, ISDN, etc
Internet business models
E-commerce models
Information analysis
Assessing value of knowledge assets
Risk assessment models
Software metrics
Data quality audits
System performance audits
User interface models
Role of PTT (domestic, exporter, multinational franchises)
Data mining
Capital (monetary, intellectual, moral, human)
Function point analysis

Design
Project management
Supply chain management
ERP
Knowledge value measurement
Intranets
OSI model
Reverse engineering
IT portfolio evaluation methodologies
Repetitive stress injury and carpal tunnel syndrome
Structured analysis
Structured programming
Total Quality Management (TQM)
Trusted systems
Capital budgeting models

Technology
Internet
Intranet
Enterprise Resource Planning (ERP)
Executive support systems
Decision support systems
Group decision support systems
Knowledge management enabling technologies
Expert systems
Intelligent agents
Joint application design (JAD)
O-O tools
CASE
Human factors check sheets
Voice recognition
Virtual private networks
INFORMATION SECURITY KNOWLEDGE AREAS (SEC)

Top level areas

INFORMATION TECHNOLOGY
CRYPTOGRAPHIC ALGORITHMS AND PROTOCOLS
SECURITY POLICY
INSTRUSION DETECTION
SECURITY ARCHITECTURES
SECURITY ASSURANCE, RISK, AND SAFETY ANALYSIS

Matrix Topics

INFORMATION TECHNOLOGY
Deals with basic elements of information technology from the perspective of how they support security and privacy policies and mechanisms. These elements are primary knowledge areas of other specialties of IT, notably Computer Science.

Theory
  Basic concepts and mathematics associated with the main areas listed under Abstraction

Abstraction
  Computer architectures
  Algorithms and data structures
  Operating systems
  Telecommunications
  Database
  Artificial intelligence
  Human computer interaction

Design
  Design principles associated with the main areas listed under Abstraction

Technology
  Technologies deployed in the marketplace in the main areas listed under Abstraction

CRYPTOGRAPHIC ALGORITHMS AND PROTOCOLS
This area deals with algorithms for enciphering and deciphering and protocols that enable cooperating processes to achieve a level of trust that enables their ongoing communications.

Theory
  Block and stream ciphers
  Perfect secrecy
  Cryptanalysis
  Information theory
  Modular arithmetic
  Factorization
Discrete logarithms
Field and group theory
Elliptic curves
Proof methods
Zero-knowledge theorems

Abstraction
Service abstractions
Authentication
Integrity
Non-repudiation
Signatures
Confidentiality
Common design flaws
Establishing a communication session
Establishing trust among network servers and clients

Design
Confusion
Diffusion
Use of nonces
Prevention of common attacks
Separate keys for separate purposes
Efficiency of protocols

Technology
DES, 3DES, AES, RC2, RC4, Blowfish, IDEA, IPSEC,
SSL, Kerberos, Diffie-Hellman, SET, S/MIME, MD5, SHA

SECURITY POLICY
This area deals with methods of specifying policies, such as
access, flow, and privacy policies, as distinct from the system
mechanisms that implement them.

Theory
Inference
Non-interference
Formal models: HRU, take-grant, SPM, TAM
Safety analysis: decidability and complexity
Impossibility of sealing covert channels
Impossibility of protecting against all viruses

Abstraction
Access matrix model
Flow models (Bell-Lapadula, Lattice)
Statistical inference models
Covert channels
Mutually suspicious systems

Design
Discretionary access control
Mandatory access control
Role-based access control
OM-AM framework (access review)
Principle of least privilege
Principle of separation of duties
Principle of abstract privileges
Principle of policy-mechanism separation
Anti-virus architectures and strategies

Technology
Unix
Windows NT
Oracle and other DBMS
CORBA

INSTRUCTION DETECTION
This area deals with systems that monitor activities in other systems to detect patterns of activity characteristic of intruders rather than authorized users.

Theory
Fundamental limitations
Statistical methods
Neural nets
Heuristic methods

Abstraction
Misuse detection
Anomaly detection
Survivability
Recovery and response

Design
Signature based design
Learning based design
Protecting the IDS system itself
Usability criteria
Effectiveness criteria

Technology
COPS
Tripwire
SATAN
ISS
Symantec
Packet sniffers
IDES
NIDES
Haystack
Emerald

SECURITY ARCHITECTURES
Security is fundamental to systems and must be part of their design. This area deals with system design principles to assure security.

Theory
Formal models of secure systems
Trust and topology
Fault tolerance
Abstraction
User-pull
Server-pull
Proxy
Agent-based architectures
Access control lists
Capabilities
Tokens
Digital certificates
Firewalls
Guards

Design
Principle of least privilege
Capability based architectures
Object oriented architectures
Service based architectures
Agent based architectures
Multi-organization architectures

Technology
Schumman SAM
Kerberos
Windows NT
Java Virtual Machine
espeak
DASCOM

SECURITY ASSURANCE, RISK, AND SAFETY ANALYSIS
This area deals with the process of assuring that a system is secure and of assessing the risks of security failures.

Theory
Errors versus faults
Software versus hardware errors
Verification and testing
Limits of formal models
Risk formalization and analysis
Probabilistic models of failure and decay
Models of software error accumulation with system age

Abstraction
N-version programming
Operational assurance
Design assurance
Development assurance
Classes of security properties
Classes of viruses and worms
Timing and storage covert channels

Design
Reference monitor
Security kernels
Border devices
Shared resource matrices
Fuzzy time
Technology
Gypsy
Ina Joe
Common Criteria
The Pump
Starlight
Checkpoint firewall
ELECTRONIC COMMERCE KNOWLEDGE AREAS (EC)

Top Level Areas

BUSINESS SYSTEMS AND PROCESSES
CUSTOMER BEHAVIOR MODELS
PERFORMANCE EVALUATION

Matrix Rows

BUSINESS SYSTEMS AND PROCESSES
Deals with the mechanisms by which businesses achieve their goals and fulfill promises to their customers and business partners.

Theory
  - Economic principles
  - Information economy principles
  - Dynamic pricing
  - Marketing principles

Abstraction
  - Electronic market places
  - Online auction models
  - Business processes
  - Workflows
  - Electronic distribution channels
  - Desintermediation
  - Reintermediation
  - E-communities
  - B2B models
  - B2C models
  - C2C models

Design
  - Order fulfillment systems
  - Online configurators
  - Supply-chain management systems
  - E-advertisement systems
  - Dynamic customization systems

Technology
  - W3C's Platform for Privacy Preferences Project (P3P)

CUSTOMER BEHAVIOR MODELS
Aimed at understanding how users interact with e-business sites through the analysis of site logs.

Theory
  - Regression analysis
  - Clustering techniques
  - Analysis of variance
Contingency tables
Neural networks
Genetic algorithms
Pattern matching

Abstraction
Data mining
Data warehousing
Online analytical processing (OLAP)
Real-time OLAP
Customer Behavior Models Graphs (CBMGs)
Customer Visit Models (CVMs)

Design
Customer behavior analysis and prediction

Technology
HTTP log analysis tools

PERFORMANCE EVALUATION
Deals with the analysis and prediction of the performance characteristics of computer systems.

Theory
Queuing theory
Markov chains
Operational laws
Product form queuing networks
Simulation models
Confidence intervals
Mean Value Analysis (MVA)
Approximate MVA
Bounds on performance
Clustering analysis

Abstraction
Queuing models of computer systems
Open, closed, and mixed queuing network models
Client/server Interaction Diagrams
Workload models

Design
Building and solving performance models of computer systems
Scalability analysis of computer systems
Benchmarking computer systems
Workload characterization techniques
Capacity planning methodologies

Technology
Queuing network solvers
Capacity planning tools
Simulation packages
Workload generators
Load testing tools
## EDUCATIONAL TECHNOLOGY KNOWLEDGE AREAS (ET)

### Top Level Areas

DISTANCE LEARNING  
INSTRUCTIONAL DESIGN

### Matrix Rows

#### DISTANCE LEARNING

Deals with institutional, technological, pedagogical, evaluation, online support, and resource support of distance learning environments.

**Theory**
- Andragogy (adult learning theory)
- Distributed Learning

**Abstraction**
- Learning Environments  
- Online Learning
- Asynchronous Learning Environments
- Synchronous Learning Environments
- Communities of Practice
- Web-Based Learning Features (Hypermedia, Multimedia)

**Design**
- Web-Based Learning Frameworks and Models
- Khan Framework  
- Levels Framework
- Methods, Strategies and Activities Framework

**Technology**
- Web-Based Course Management Tools
- Web-Based Course Authoring Tools
- Bulletin Boards  
- Discussion Boards
- MOOs, MUDs

#### INSTRUCTIONAL DESIGN

Deals with the analysis, design, development, implementation, and evaluation of instructional and training systems.

**Theory**
- Learning Theory
- Systems Theory
- Cognitive Information Processing
- Situated Learning
- Problem-based learning
- Cognitive Apprenticehips
Abstraction
   Instructional Systems models
   Gagne Briggs Model
   Kemp's model
   Dick and Carey's model
   The ADDIE model

Design
   Front End Analysis
   Performance Objectives
   Task Analysis
   Instructional Strategies
   Learning Strategies
   Design Documents
   Storyboarding
   Assessment
   Project Management

Technology
   Computer-Based Instruction
   Web-Based Instruction
   Digital Audio and Video
   Tools for Visual Design
   Project Management Tools
   Scripting and Programming
   Animation