The Internet of Things
The Death of a Traditional Database?

CSC230 Database Technologies for Analytics

19 November 2016

Computer science

- definition of computer science in one question
  - What can be automated?
- definition of computer science in four questions
  - which questions can be answered by a computer program?
  - how to write a program that answers a given question?
  - how to select the best program that answers a given question?
  - how to be sure that a program really solves exactly a given question, every time, without error?
- $algorithms + data_structures = programs$
- what is artificial intelligence
  - Artificial Intelligence Today and Tomorrow, Kris Hammond, ComputerWorld, 10 April 2015
  - computer completes a task previously thought to require a human being
  - Turing Test
    - put a computer in one room
    - put a person in a second room
    - put human interrogators in a third room
    - interrogators send written questions to other two rooms, receive written responses (unbiased by appearances, accents)
    - if interrogators cannot determine from the responses which room has a person and which room has a computer, then they must conclude that the computer is "intelligent"

  - strong AI—machine produces same results as humans, in the same way
  - weak AI—machine produces same results as humans, not necessarily in same way
  - "in between" AI—use what is known about human beings reason and how brain works to design computers and software, but without a commitment to follow biological model faithfully
  - narrow AI—single, specific task
  - general AI—reason in many domains
  - applications...
    * robotics—machines that pick parts from a bin, assemble parts, robots that walk
    * autonomous vehicles—self-driving cars
    * machine vision
    * speech recognition, text-to-speech, and speech synthesis
    * recommend books, movies
    * pattern matching, classification, identification, diagnosis
    * machine learning—from many examples
- any research at the cutting edge?
- machines that substitute for human beings or machines and human beings working together?
- magnify human capacities?
- Google—artificial intelligence in an unexpected form?
- rate of progress in computer science
  - fast!
    * Moore's Law: double # transistors on an integrated circuit (chip) every 2 years
    * similar rate of progress with mass storage (disks)
    * double price/performance every 2 years
    * 50 years since Moore's prediction $\mapsto$ 25 doublings
    * $2^{25}\times \approx 32$ million$\times$
    * faster than in any other field
    * faster in now than in any period in history
  - slow!
    * RISC chips
      · invention in 1970s

        - Reduced Instruction Set Computers (as opposed to CISC: Complex Instruction Set Computers)
        - smaller set of simpler instructions (rather than larger set of more powerful instructions)
        - smaller set of addressing modes (ways of specifying location of operands)
        - more uniform representation of instructions (less variety of forms)
        - faster execution (execute each instruction in a single cycle)
        - load/store architecture
        - arithmetic instructions read/write registers (rather than read/write main memory)
        - widespread adoption in 1990s
      * object-oriented programming languages
        - invention in 1970s
        - class—a blueprint for the creation of objects
        - object—a bundle of related data and methods for accessing, updating, and combining that data
        - make programs easier to read, easier to write
        - avoid repeating code
        - define one class in terms of another class
        - inheritance—definition of subclass needs to include only features that distinguish the subclass from parent class (shared features come for free)
        - overriding a method—method in subclass works diffently than same method in parent class
        - polymorphism—objects belonging to different classes carry out same task in different ways
        - polymorphism—squares compute their areas diffently than do circles
        - polymorphism—computer recognizes which version of a method belongs to a given object, executes it automatically
        - classes/objects are reusable components
        - build programs with reusable components (instead of writing everything from scratch)
        - widespread adoption in 1990s
    - world spoiled/deceived by rapid rate of progress in computing?

- how does software engineering differ from other kinds of engineering?
    - less history/experience from which to learn
    - harder to study and learn from successful design

  - (compilation hides the details of a design)
  - scale models are useless
  - sudden rather than gradual failure
  - no amount of testing can guarantee that all defects have been discovered
  - most complex products that human beings have ever designed
  - software never wears out
  - unconstrained by laws of physics (danger in having the freedom to exercise too much imagination?)
- lessons that we have learned about how to create software
  - write just a little at a time
  - begin with a working example, modify just a little bit at a time
  - test frequently!
  - annotate code—explain to yourself and present and future teammates
  - recognize the need for a second pair of eyes—work with partners

Outline of article

- predicting future of information and communication technology
- expert groups/projects
  - GRIDs
  - CLOUDs
  - service-oriented architectures
  - quantum computing
  - bio-computing
  - new materials
  - human-computer interaction
  - cognitive technology
- Internet of Things is a strong theme
- e-Infrastructure in Europe, Cyberinfrastructure in US
- database researchers not well represented
  - surprising!
  - future requires...
    * interoperation with existing database technology (at least)

        * evolutionary or revolutionary technology (more likely)

- database research
  - semi-structured data—processing, managing of data streams
  - schema matching, mapping for interoperation, domain ontologies
  - Web-database interfaces
  - modeling and systems development
  - performance, query optimization with new algorithms
  - optimized storage architecture—P2P (peer–to–peer)
- researchers challenged to match/contribute to advances in...
  - social networking
  - content creation and repurposing
  - game
  - sensor systems
  - robotics, autonomous systems
  - visualization
  - user interaction
  - systems and software development
  - service-oriented architecture
- vision of Europe and the world 20 years from now...
  - always-on, always with us devices for connecting to the Internet
  - invisible infrastructure, optimizes performance, reliability, cost, security
  - sense, detect, record, curate everything
  - default universal sharing of data (with protection of ownership and privacy)
  - at home, in industry, in social services
  - embedded subsystems
    * agriculture
    * transportation/vehicles
    * medicine
    * generation/distribution of power
- vision of Europe and the world 20 years from now: implications...
  - need for smaller, faster, cheaper, more energy-efficient devices
    * less heat

        * biologically inspired
        * quantum computing
    - intelligent materials, Internet-ready
        * agricultural products
        * manufactured products
    - open availability & physical access produce demands for...
        * increased performance
        * reduced latency
        * greater scalability
        * greater reliability
        * more self-management
    - middleware's responsibility...
        * self-* characteristics
            · self-managing
            · self-tuning
            · self-repairing
        * security/access/trust
            · identification
            · authorization
            · trust
            · security
            · privacy
            · access control
- infrastructure stack (bottom to top)
    - e-infrastructure—communication
    - i-infrastructure—processing
        * collect
        * structure
        * manage
        * describe
        * manipulate
    - k-infrastructure—knowledge
        * semantics
        * extract knowledge from information—deduction, induction
        * codify, store knowledge
        * API—interface to application layer
- SOKU: Service-Oriented Knowledge Utilities

  - discoverable
  - composable
  - dynamically tunable
  - metadata (to make discovery, composition, tuning possible)
  - content
    * massive, includes...
      · structured, verified
      · streams of data from detectors (sensors)
      · personally authored
      · education
      · entertainment

- send software to data, rather than data to software (because of volume of software)

- keys to development of SOKUs
  - well-defined interface to e-infrastructure
  - use of off-the-shelf, tested components
  - rapid development

- markets/domains
  - B2C—business to consumer
  - B2B—business to business
  - E2E—enterprise to enterprise (departments within an organization serving one another?)
  - R & D

- how decisions made by...
  - deduction
  - induction
  - simulation

- non-functional characteristics of devices
  - performance
  - security
  - use-conditions

- devices
  - 'intelligent'

  - 'learn' (behavior changes/improves over time)
  - end-user does not known (or want to know) location
  - service level agreements negotiated by agents
- kinds of applications
  - general (large, diverse audiences)—precomposed
  - specialized (specific industries, markets, social groups)—dynamically constructed
- seamless integration
  - planning travels
  - managing projects
    - define tasks
    - order tasks, specify dependencies/prerequisites
    - establish benchmarks/milestones
    - track/report progress
  - collaborating with teammates
  - scheduling meetings
- some (or all) of Information and Communication Services outsourced/placed in cloud...
  - "virtualization"
  - IaaS: Infrastructure as a Service
  - PaaS: Platform as a Service
  - AaaS: Application as a Service
  - EaaS: Enterprise as a Service
- what kind of R & D is required?
  - metadata a part of all priorities
  - formal syntax, declared semantics
  - metadata to facilitate mobility of software services
    - send software to node where it is needed
  - metadata to describe sources of data
    - structured, semi-structured, unstructured
    - temporal properties
    - degrees of certainty (propabilities)
  - metadata to manage agents (?)
- replacement of established database technologies

  - composition, orchestration of SOKUs
  - intelligent dialogues between SOKU and (end-user) SOKU agents

- challenges in research
  - distinguish between data and metadata
  - categorize metadata
  - measure, maintain state on millions of nodes
  - describe data types, attributes
  - describe precision and accuracy of data
  - describe source, trustworthiness of data
  - describe duration of relationships
  - describe certainty of relationships
  - trade performance for price
  - declare, enforce, monitor policies for trust, security, and privacy
  - cleanly separate services (processes), data (information and knowledge), agents (roles, consumers)

- special challenge in detail: management of state
  - conventional (relational) databases
    * ACID: Atomic, Consistent, Isolated, Durable
    * goal: maintain state in nearly real-time
    * locking to prevent transactions from overlapping
    * duration of locks depends upon...
      · number of instructions executed in each update
      · number of tables updated
    * rollbacks reschedule locked-out transactions
    * compensation restores state during rollbacks
  - distributed databases
    * geographically distributed data
    * two-phase commit protocols
    * compensation (restoration of state) as in non-distributed databases
    * microseconds to minutes to execute protocols
    * does not work with...
      · millions of nodes
      · frequent, automated updates from sensors (audio, video, etc.)
      · frequent updates from numerous, geographically separated human clients
    * solutions?

        - settle for maintenance of consistency of states in local neighborhood
        - use lazy methods ("eventual consistency?")
        - rethink meaning/importance of state, transactions
- special challenge in detail: representation of data
    - multitude of types requires labeling, descriptions of data to make it useable
    - metadata
        - describe attributes
        - describe location
        - describe ownership, permissions
    - restriction to hierarchical organizations of data too restrictive
    - real world does not map to hierarchies
    - relationships change over time
    - need for something like Apple's Time Machine: view and review relationships over time
- theoretical (mathematical) foundations
    - relational calculus
    - fuzzy logic (reasoning with probabilities/uncertainty)
    - graph theory
- conclusions
    - dataspaces (people + machines)—people solve problems with help of machines
    - relational database technology is 40 years old
    - databases are an important component of Web applications
    - computer scientists invented the Web over the last 20 years
    - databases technologies neglected by researchers during that time
    - people will invent ways of using new technologies to behave badly

References

[1] Keith G. Jeffery, *The Internet of Things: The Death of a Traditional Database?*, IETE Technical Review, Volume 26, Issue 5, Sep–Oct 2009, pages 313–319