Introduction to Civil Information Systems

Mark A. Austin

University of Maryland

austin@umd.edu
ENCE 688R, Spring Semester 2023

January 27, 2023

Overview

1. Modern Civil Infrastructure Systems
2. Near-Term Challenges (2020-2060)
3. Infrastructure Protection and Recovery
4. Transition to Information Era
5. Features of Modern Computing
6. Cyber-Physical and Digital Twin Systems
7. Engineering Sensor Systems
8. Urban and Global Applications

Part 3

Transition to Information Era

Post- Industrial Revolution (Mid-1900s)

New types of systems – planes, trains and automobiles – rely on human involvement as a means for sensing and controlling behavior, e.g.,

- Driving a car,
- Manual collection of road tolls,
- Traffic controllers at an airport,
- Manual focus of a camera.

Systems work, but:

- Humans are slow.
- Humans make mistakes.
- They also easily tire.

Transition to Information Era

Since 1990 we have been in an Information Era

Characteristics	**Stage 2** **Electrical Era**	**Stage 3** **Information Era**
Onset in the U.S.	Late 1800s.	**Late 1900s.**
Economic Focus	Manufacturing	Services
Technologies	ElectroMechanical	**Information**
Product Lifecycle	Years	Months
Living Standard	Quality of Goods	**Quality of Life**
Geographical Impact	Regional/National	**Global**

Design of Information-Age Systems

Premise of Information-Age System Design:

- Advances in computer software, sensing, and wireless networking technologies can work together to expand the functionality and performance of systems.

Trend toward Automation:

- New types of systems where human involvement for management of system functionality is replaced (or partially replaced) by software automation.

Civil Engineering Applications:

- Automated road toll collection (Rt. 200).
- Automated baggage handling systems at airports.

Transition to Information Era

Metrics of Good Engineering Design:

- A good engineering design works correctly, has good performance, and is economical.
- Functionality and performance are resilient to uncertainties.
- System can be easily upgraded to take advantage of new technologies.

Metrics of Good Systems Operation:

A well-run system has "situational awareness" and handles unexpected events:

- Sense the system state and surrounding environment,
- Look ahead and anticipate events, and
- Take action to control system behavior.

Features of Modern Computing

Key Question: **How can we use modern computing technologies to improve Civil Engineering Systems?**

Man and Machine (Traditional View)

Man	Machine
• Good at formulating solutions to problems. • Can work with incomplete data and information. • Creative. • Reasons logically, but very slow. • Performance is static. • Humans break the rules.	• Manipulates Os and 1s. • Very specific abilities. • Requires precise decriptions of problem solving procedures. • Dumb, but very fast. • Performance doubles every 18-24 months. • Machines will follow the rules.

Sensible Problem Solving Strategy

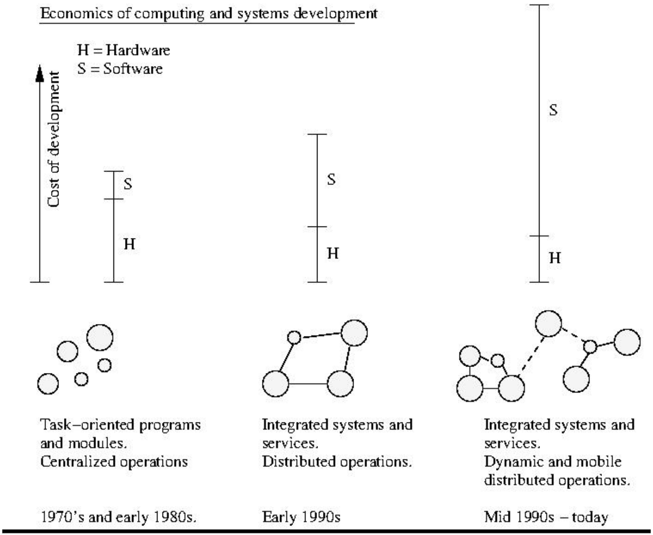
Let engineers and computers play to their strengths:

- Accelerates the solution procedure.
- Enables the analysis of problems having size and complexity beyond manual examination.
- Adds value in areas that will lead to long-term economic growth.

Getting things to work We need to:

- Describe to the computer solution procedures that are completely unambiguous.
- Look at data, organization and manipulation of data, and formal languages.

Expanding Expectations of Computing



Economics of computing and systems development

H = Hardware
S = Software

Task–oriented programs and modules.
Centralized operations

Integrated systems and services.
Distributed operations.

Integrated systems and services.
Dynamic and mobile distributed operations.

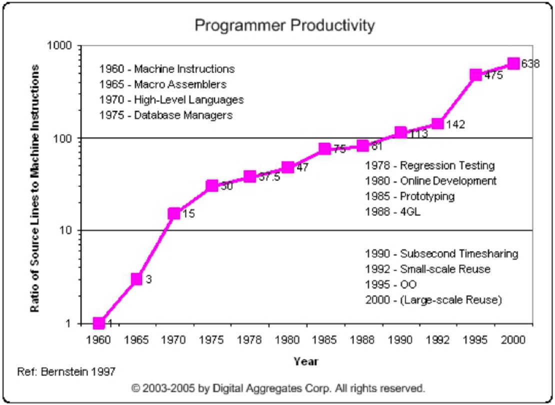
1970's and early 1980s.

Early 1990s

Mid 1990s – today

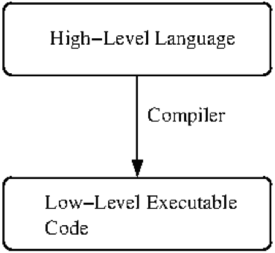
Pathway to Improved Programmer Productivity

Increasing System Complexity: Software programmers need to find ways to solve problems at high levels of abstraction.



Ref: Bernstein 1997

© 2003-2005 by Digital Aggregates Corp. All rights reserved.



High–Level Language

Compiler

Low–Level Executable Code

Evolution of Computer Languages

Computer Languages. Formal description – precise grammar – for how a problem can be solved.

Evolution. It takes about a decade for significant advances in computing to occur:

Capability	1970s	1980s	1990s
Users	Specialists	Individuals	Groups
Usage	Numerical computations	Desktop computing	E-mail, web, file transfer.
Interaction	Type at keyboard	Screen and mouse	audio/voice.
Languages	Fortran, C	MATLAB	HTML, Java

Popular Computer Languages

Tend to be designed for a specific set of purposes:

- FORTRAN (1950s – today). Stands for formula translation.
- C (early 1970s – today). New operating systems.
- C++ (early 1970s – today). Object-oriented version of C.
- MATLAB (mid 1980s – today). Stands for matrix laboratory.
- Python (1990s – today). A great scripting language.
- HTML (1990s – today). Layout of web-page content.
- Java (1994 – today). Object-Oriented language for network-based computing.
- XML (late 1990s – today). Description of data on the Web.

Post- 2000 Era

Imagine: What if COVID-19 had arrived in 2000?

- No iPhone, No iPad, No iTunes.
- No Facebook, No Instagram, No WhatsApp.
- No Google Maps, No Google Streetview.
- No Dropbox, No Zoom.

Recent Advances in Technology:

- Average internet speeds: In 2000, 0.07 Mbs; In 2009, 5-7 Mbs; In 2020, 100-200 Mbs; 5G, 1000-2000 Mbs.
- Cloud-based data storage and computational services (AWS).
- New languages: Swift → App development on iPhone/iPad.
- Many new types of sensors and methods of data collection.

Post- 2000 Era

New Computing Infrastructure → New Architectures, Languages, ...

Capability	**2000-present**	**2020-2030**
Users	Groups of people, sensors and computers.	Integration of the cyber and physical worlds.
Usage	Mobile computing. Control of physical systems. Social networking.	Embedded real-time control of physical systems.
Interaction	Touch, multi-touch, proximity.	....
Languages	XML, RDF, OWL.	New languages to support time-precise computations.

Post-2010 Era → Emergence of AI

State-of-the-Art Implementation (2020, Google, Siemens, IBM)

- AI and ML will be deeply embedded in new software and algorithms.

Artificial Intelligence:

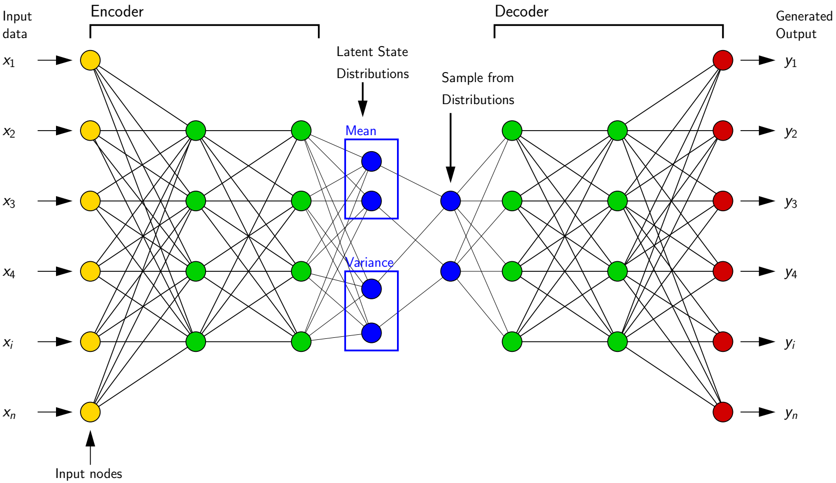
- Knowledge representation and reasoning with ontologies and rules. Semantic graphs. Executable event-based processing.

Machine Learning:

- Modern neural networks. Input-to-output prediction.
- Data mining.
- Identify objects, events, and anomalies.
- Learn structure and sequence. Remember stuff.

Post-2010 Era → Generative AI

Variational AutoEncoders (Generative Models)



Post-2010 Era → AI Generated Architecture

Convergence: Engineering-Architecture-AI

AI-generated art …

AI-generated building architecture

References

- Array of Things: See https://arrayofthings.github.io
- Austin M.A., Delgoshaei P., Coelho M. and Heidarinejad M. , Architecting Smart City Digital Twins: Combined Semantic Model and Machine Learning Approach, Journal of Management in Engineering, ASCE, Volume 36, Issue 4, July, 2020.
- Bello J.P. et al., SONYC: A System for Monitoring, Analyzing, and Mitigating Urban Noise Pollution, Communications of the ACM, 62, 2, 2019, pp. 68-77.
- Coelho M., and Browning L.S., INL Digital Engineering: Model-Based Design, Digital Threads, Digital Twins, Artificial Intelligence, and Extended Reality for Complex Energy Systems, INL/CON-22-69247, Idaho National Laboratory, Idaho Falls, Idaho 83415, September, 2022.
- Leveson N.G., A New Approach to Software Systems Safety Engineering, System Safety Engineering: Back to the Future, MIT, 2006.
- Tien J.M., Toward a Decision Informatics Paradigm: A Real-Time Information-Based Approach, to Decision Making, IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, Vol. 33, No. 1, February, 2003.