5.3 Emerging Computational Technologies

Despite the potential for many beneficial improvements, current software development practices are relatively stable[1]. In contrast, many advances in computational technology are being explored, emerging for eminent use, or already being adopted for limited use. This section considers the implications, such as can be foreseen, for these advances on software development practices.

{focus more clearly on how each of these affect software-based development practices}

Specialized Processors

Advances in computational capabilities are often first realized as complex software built to operate on existing general-purpose computational devices. As these advances become better understood, refined, and standardized, specialized computational devices are built that both simplify the software and allow it to operate more efficiently. Purpose-built hardware elements can be the basis for specialized computational devices, subsequently when warranted to be integrated into general-purpose computing hardware.

The effect of this progression on software development practices is to evolve specialized software to services that encapsulate use of specialized hardware capabilities:

- A natural result of hardware-software collaborative design tradeoffs (i.e., "codesign")
- Software-first development that can target software to operate on a platform of the most suitable computational devices
- Software-enabled devices that embed (general-purpose or specialized) computational hardware for enhanced hybrid realizations of virtualized capabilities

---

[1] Not withstanding promising excursions into machine learning and extra-realities (see section 5.4)

- Flexibility to enhance hardware purpose with software—no tension between specialization and integration of broader functionality in a hardware device (e.g., extension of graphics processor capabilities to support other parallel processing)

Examples of this sort of device, providing hardware-augmentation of initially developed software capabilities, are:

- Environmental processing (i.e., sensing and producing sensory phenomena)
- Signal processing (i.e., Analog-to-digital and digital-to-analog analysis and conversion of electromagnetic spectrum signals)
- Sensory processing (i.e., converting signals between analog and digital forms corresponding to human and extra-human senses)
- Media processing (i.e., audio/video encode/decode/transcode/accelerate)
- Optimized neural processing, specializing generalized parallel processing supportive of AI/neural net logic for data analysis and machine learning (e.g., for data, image, and speech processing and analysis)
- Optimization of time-sensitive functionality (e.g., motion tracking, identity, security processing, image processing, remote sensing, speech/language translation)
- Parallel processing of complex mathematical operations to synchronously transform a homogeneous data set (e.g., process graphical content for visual display; processing of highly parallel operations for data mining/analysis and machine learning)

Remote and Virtualized Computing Resources

{high-perf/parallel computing (local multi-core/multi-processor node controls network process; complex predictive models/simulations (mega-data process analysis: weather, ocean dynamics)}

{effect on software development practices: build assuming virtualized computational resources disregarding location while accounting for latency; build to defer selection of most appropriate resources}

{consider augmented/virtual reality in context of product verification}

Cyber-Physical Computing Devices

(motivation: cyberphysical systems for managing/controlling monitored environment)

edge devices as autonomous and as nodes in CyberPhysical systems (connected computation and data collection by all devices)

(enabled by size- and cost-reduction in sensor-effector tech, distributed/remote computing capabilities; distributed monitoring, data collection, and control of larger environmental space; autonomously-operating devices heavily replicated & embedded in natural/artificial elements, remotely managed and controlled)

Distributed Computing

{anywhere-anytime augmented computing => robotics}

Distributed processing (communications, data storage and sharing)

Mobile computing (devices, vehicles, drones)

Autonomous computing

Manufacturing Technology

hardware-software codesign

fabrication of hardware from software-specified models

mass customization for manufacture of customized hardware

additive (3-D printing) and robotics manufacturing techniques (for devices and structures)

Quantum Computing

As quantum computing capabilities are developed, questions arise concerning how best to develop software that defines computational behavior using this technology. It has been advanced that such computation lends itself to solving problems that are infeasible or inefficient to solve conventionally.

It may be that existing forms of software specification can be applied to express and be translated into a form that defines the expected behavior of a quantum computing engine. Alternatively, it may be that new forms of expression may be more appropriate for specifying quantum processing capabilities. The ideal, from a software engineering perspective, would be to have an expressive form for software behavior that can be targeted to either digital or quantum processing technology, depending on accessibility to that technology and feasibility for solving the problem, with the type of hardware on which it is employed to be decided, and changed, as circumstances warrant.

{characterize type of problems expected, to include factoring large numbers/decryption and machine learning}

In software development practice medium-term, quantum computing is likely to be realized by means of a suitable aggregation of conventional and specialized processors having particular capabilities that are infeasible or inefficient with only conventional computing. During early product development, it may be that quantum computations can be approximated (simulated) on conventional processors for purposes of initial verification of non-quantum aspects of product behavior.

Biologic Computing

{https://en.wikipedia.org/wiki/Biological_computing}
{https://en.wikipedia.org/wiki/Genomics}

{molecular, organic, genetic, DNA, proteins}

Viewing genomics from a computational perspective, DNA can be viewed as an encoding of genetic instructions for a living organism. The operations of those instructions can be viewed in theory as a computation that determines the functional capabilities of that organism. DNA can in principle be expressed or modified so as to define an organism having different capabilities. Experimentally, particular realizations of DNA have been achieved for performing massively parallel computations *(https://en.wikipedia.org/wiki/DNA_computing)*.

As with quantum computing, practical biologic computing would most likely take the form of a specialized computing engine for solving problems that are not well suited to conventional digital or analog computing capabilities *(https://en.wikipedia.org/wiki/Real_computation)*.

{https://en.wikipedia.org/wiki/Analog_computer
https://en.wikipedia.org/wiki/Computer#Digital_computers}