5.3 Computational Technologies

Despite the potential for many beneficial improvements, current software development practices are relatively stable and effective, at least for building singular products[1]. In contrast, many advances in computational technology that will enable software to have greater capabilities are being explored, emerging for imminent use, or already being adopted for limited use. The implications, such as can be foreseen, for these advances on software development practices are considered here.

{address only how each category of computational technology will affect software development}

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 ("codesign") tradeoffs
- 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
- 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)

[1] Not withstanding increasing excursions into machine learning and extended reality

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

- Signal processing (i.e., Analog-to-digital and digital-to-analog analysis and conversion of electromagnetic signals)
- Sensory processing (i.e., sensing and producing environmental phenomena)
- Media processing (i.e., audio/video encode/decode/transcode/accelerate)
- 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)
- Neural net processing, using generalized parallel processing of AI logic for data analysis and machine learning (e.g., for processing and analysis of data, images, and speech)

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)}

{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: cyber-physical systems for managing/controlling a monitored environment)

edge devices as autonomous and as nodes in cyber-physical systems (connected computation and data collection by all devices, data correlation across 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 capabilities (robots, spacecraft)

challenge of time synchrony, connectivity, transmission delays/latency (including over galactic distances), and failures

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)

Spatial Computing

"Spatial computing is human interaction with a machine in which the machine retains and manipulates referents to real objects and spaces."[2] *{human interaction is not essential; the relevant point is the intermix of physical and virtualized content, with or without human intervention} {what applicability does this technology have in software-based product development?}*

[2] S. Greenwold, *Spatial Computing* (graduale thesis), Massachusetts Institute of Technology, Cambridge, MA, June 2003. < https://acg.media.mit.edu/people/simong/thesis/SpatialComputing.pdf >

Quantum Computing

The emerging science and technology of quantum computing is likely leading to the ability to solve problems that are infeasible or inefficient to solve conventionally (e.g., related to encryption, machine learning, simulation of complex systems, or analyses of alternatives). As quantum computing capabilities are developed, questions regard how best to develop software that defines computational behavior using this technology.

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 will be conceived 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.

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

Biologic Computing

Organic (chemical and biologic) and analog mechanisms can also be viewed as performing a computational process[3]. 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.

Viewing genomics from a computational perspective, DNA is the encoding of genetic instructions that determine the capabilities and potential behavior of an organic system. DNA can in principle be expressed or modified so as to define a system having

[3] https://en.wikipedia.org/wiki/Biological_computing ; https://en.wikipedia.org/wiki/DNA_computing

prescribed biologic capabilities. Experimentally, particular realizations of DNA have been achieved for performing massively parallel computations. Although the ability to implement DNA-based computation raises unresolved ethical questions and potential risks, its similarities to other forms of computation is of theoretic interest.

https://en.wikipedia.org/wiki/Genomics

https://en.wikipedia.org/wiki/Real_computation

https://en.wikipedia.org/wiki/Analog_computer

https://en.wikipedia.org/wiki/Computer#Digital_computers