

5.4 Data Science, Artificial Intelligence, and Machine Learning

One of the themes of early work in artificial intelligence (AI) was referred to as “automatic programming”. The essential diversity and complexity of software and the underlying tacit and difficult to precisely express technical and domain-specific knowledge required to build software are still beyond the feasibility of current techniques.

While the more mundane aspects of software development have been susceptible to greater automation, the more creative aspects have been satisfied, in an industry focused on low-investment/short-term economics, with what continues to be labor-intensive practices. Although AI efforts have not supplanted how software has been traditionally developed in practice, advances in AI, including machine learning based on neural networks, offer opportunities for potential improvements in how software is developed.

Derive Component Families from Instances

Analyze a set of previously developed components based on “similarity” among them, resulting in identifying component families of instances that have similar and equivalent behaviors (functionality and properties); identify associated decisions that are sufficient to distinguish similar instances; derive an adaptable component for each family by which instances can be derived by resolving associated decisions.

Establish a Products Taxonomy

Analyze a set of previously developed products in order to formulate a taxonomy of software-based products based on similarities in capabilities that characterize each product; create a capabilities-based taxonomy of products into which future products can be localized and a set of “closest” match products identified as the base for its efficient derivation.

Architecture Refinement for Quality Improvements

Derive alternative candidate product architectures based on analyses of alternative combinations of behavioral quality factors and alternative candidate software components corresponding to those combinations.

Delineate Behavioral Quality Factors at the Component Level

Analyze the contribution that each component in a product architecture needs to make to the various behavioral quality factors; evaluate candidate component instances in terms of fit to this contribution.

Optimize Software for Hardware Mapping and Configuration

Define a computational platform family whose instances are characterized and selectable in terms of specified software capabilities to be supported; if computational capabilities are constrained, build software to fit an available computational platform configuration that is a sufficiently close fit; estimate how behavioral quality factors are affected by alternative computational platforms.

Analyze How Alternative Decisions Affect a Resulting Product

Given an appropriate decision-based characterization of the instances of an envisioned product family, determine how alternative decision resolutions affect behavioral quality factors of the corresponding instance products.

Code Derivation from Hardware Behavioral Specifications

Derive alternative software emulations of hardware devices based on a software specification of its intended behavior and an analysis of these alternatives based on tradeoffs among behavioral quality factors; encapsulate hardware functionality within software that extends a device's capabilities with enhanced behavior (e.g., health monitoring for repair or replacement based on diagnostics and prognostics, data logging, data stream prediction and interpolation)

Build Introspective Software

Determine how to build efficient self-aware software capabilities that can analyze, determine rationale for its behavior, and provide explanations by which developers can evaluate correct behavior and detect flaws; determine how to provide similar capabilities by which a product can instruct, monitor, and improve its effective use; create capabilities by which software can monitor and evaluate specified hardware behavior so as to detect or predict failures and guide diagnostic- and prognostic-based remediations.