

The modern software development process is making construals?

Richard Cartwright

Computer science follows a course of simplifying the development of business logic through ever increasing abstractions from the underlying hardware. Creating abstraction layers enables domain experts to optimise mappings from logical to physical platforms. Beyond languages / compilers, networking protocols, the world wide web and virtualisation / cloud, the artefacts of modern software development are adapting to an emerging world of serverless computing and the Internet of Things (IoT). This talk presents observations of modern software development and design – a expanding universe of executable and/or animated boxes and lines – where asynchronous distributed system configuration is replacing imperative, sequential programs. With examples from Amazon Web Services, IBM's Node-RED flow-based graphical programming language and more, this talk illustrates how developers can both experiment with and communicate their ideas through models in web browsers and/or fragments of code in Read-Eval-Print Loops (REPLs). Effectively, is this a process of building construals that either fail fast or - with the addition of tests and a "dev ops" process - get deployed live?