The Role of Software in Systems

by Gerrit Muller  USN-SE

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

Software is a dominating factor in the development of complex systems. It plays a crucial role in the performance of the final product at the one hand, while it contributes significant to the development cost and elapsed time of development. This paper will discuss the role of software in the broader system context. An improved understanding of the role of software enables the system architect, and the other stakeholders of the product creation process, to integrate the software development better. In this way hardware-software tradeoffs can be made, balancing performance, costs and risks.
Relative Contribution of SW

- Electronics
- Mechanics
- Physics/Chemistry, etc.

The Role of Software in Systems

version: 1.3
March 27, 2021
RSWRelativeEffort
Mismatch between Role and Discipline

**role of software**
integration technology
captures *application* functionality
defines lot of *system* behavior
determines how much of potential *system* performance is achieved
acts as director

**focus of software discipline**
software technologies, such as:
  - programming languages
  - data bases
  - operating systems
  - component technologies
  - engineering practices

mismatch!
Control Hierarchy along Technology axis

- **Sensor**: collect data from the environment
- **Optical device**: capture images or measure distances
- **Mechanical device**: perform physical tasks
- **Human user**: interacts with the system
- **Application software (SW)**: processes user input and controls the system
- **Control software (SW)**: manages the application software
- **Digital electronics**: processes data from sensors
- **Analog or power electronics**: controls mechanical devices
- **Local automation or safety**: ensures system operation and safety

Legend:
- Local automation or safety
Characterization of disciplines

Mechanics  Analogue / power Electronics  Digital Electronics  Software

Concrete  Tangible  Mature  Production lead-time  Material cost  Abstract  Intangible  Immature  Flexible?

The Role of Software in Systems
Design Aspects related to SW

- Customer objectives
- Application
- Functional
- Conceptual
- Realization

design philosophy per quality attribute
granularity, scoping, containment, cohesion, coupling
interfaces, allocation, budgets
information model (entities, relations, operations)
identification, naming
static characteristics, dynamic behavior
system-level infrastructure
software development process, environment, repository, and tools
life cycle, configuration management, upgrades, obsolescence
feedback tools, for instance monitoring, statistics, and analysis
persistence
licensing, SW-keys
setup sequence, initialization, start-up, shutdown
technology choices
make, outsource, buy, or interoperability decisions

performance, safety, security, ...
e.g., distributed or centralized control

HAL_message_acknowledge_status versus ACK
SW Mechanisms

Customer objectives
Application Functional Conceptual Realization

error handling, exception handling, logging
processes, tasks, threads
configuration management; packages, components, files, objects, modules, interfaces
automated testing: special methods, harness, suites
signaling, messaging, callback scheduling, notification, active data, watchdogs, timeouts
locking, semaphores, transactions, checkpoints, deadlock detection, rollback
identification, naming, data model, registry, configuration database, inheritance, scoping
resource management, allocation, fragmentation prevention, garbage collection
persistence, caching, versioning, prefetching, lazy evaluation
licensing, SW-keys
bootstrap, discovery, negotiation, introspection
call graphs, message tracing, object tracing, etc.
distribution, allocation, transparency; component, client/server, multitier model