Module Role of Software in Complex Systems

by Gerrit Muller       Buskerud University College

 e-mail: gaudisite@gmail.com

 www.gaudisite.nl

Abstract

This module addresses the role of software in complex systems
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

The Role of Software in Systems

version: 1.3
October 22, 2014
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

**mismatch!**

**focus of software discipline**

software technologies, such as:
  - programming languages
  - data bases
  - operating systems
  - component technologies
  - engineering practices
Control Hierarchy along Technology axis

- human user
- application SW
- control SW
- digital electronics
- analog or power electronics
- mechanical device
- optical device
- sensor

Legend:
- local automation or safety
Quality Attributes annotated with SW relation

usable
- usability
- attractiveness
- responsiveness
- image quality
- wearability
- storability
- transportability

interoperable
- connectivity
- 3rd party extendible

liable
- liability
- testability
- traceability
- standards compliance

dependable
- safety
- security
- reliability
- robustness
- integrity
- availability

efficient
- resource utilization
- cost of ownership

effective
- throughput or productivity

consistent
- reproducibility
- predictability

serviceable
- serviceability
- configurability
- installability

logistics friendly
- manufacturability
- logistics flexibility
- lead-time

ecological
- ecological footprint
- contamination
- noise
- disposability

down-to-earth attributes
- cost price
- power consumption
- consumption rate
  (water, air, chemicals, etc.)
- size, weight
- accuracy

legend
- weak SW relation
- strong SW relation

The Role of Software in Systems
version: 1.3
October 22, 2014
RSWqualities
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
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
Describe the SW in a complex product, from different viewpoints for instance:

- Give an indication of the size/complexity
- Outline the SW architecture
- Identify the top 3 critical characteristics
- Identify potential improvements
- Process
- Development environment
Role of Software

Integration technology captures application functionality and defines a lot of system behavior. It determines how much of the potential system performance is achieved and acts as a director.

Focus of software discipline includes software technologies, such as:
- Programming languages
- Databases
- Operating systems
- Component technologies
- Engineering practices

Mismatch!

Control Hierarchy

Discipline Characteristics

Concrete, tangible, mature, production lead-time, material cost.

Abstract, intangible, immature, flexible?

Mechanics, analogue/power electronics, digital electronics, software.

Intentionally left blank.