

# Module Role of Software in Complex Systems

by *Gerrit Muller* Embedded Systems Institute  
e-mail: `gerrit.muller@embeddedsystems.nl`  
`www.gaudisite.nl`

## Abstract

This module addresses the role of software in complex systems

### Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

February 11, 2012  
status: preliminary  
draft  
version: 1.0



Embedded Systems  
INSTITUTE

# The Role of Software in Systems

by *Gerrit Muller* Buskerud University College

e-mail: `gerrit.muller@embeddedsystems.nl`

`www.gaudisite.nl`

## Abstract

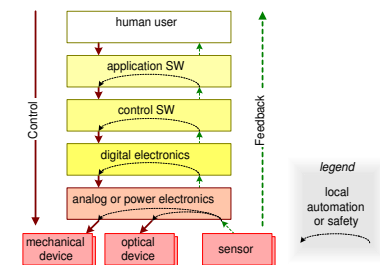
Software is a dominating factor in the development of complex systems. It plays a crucial role in the performance of the final product at the one hand, while it contributes significant to the development cost and elapsed time of development.

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.

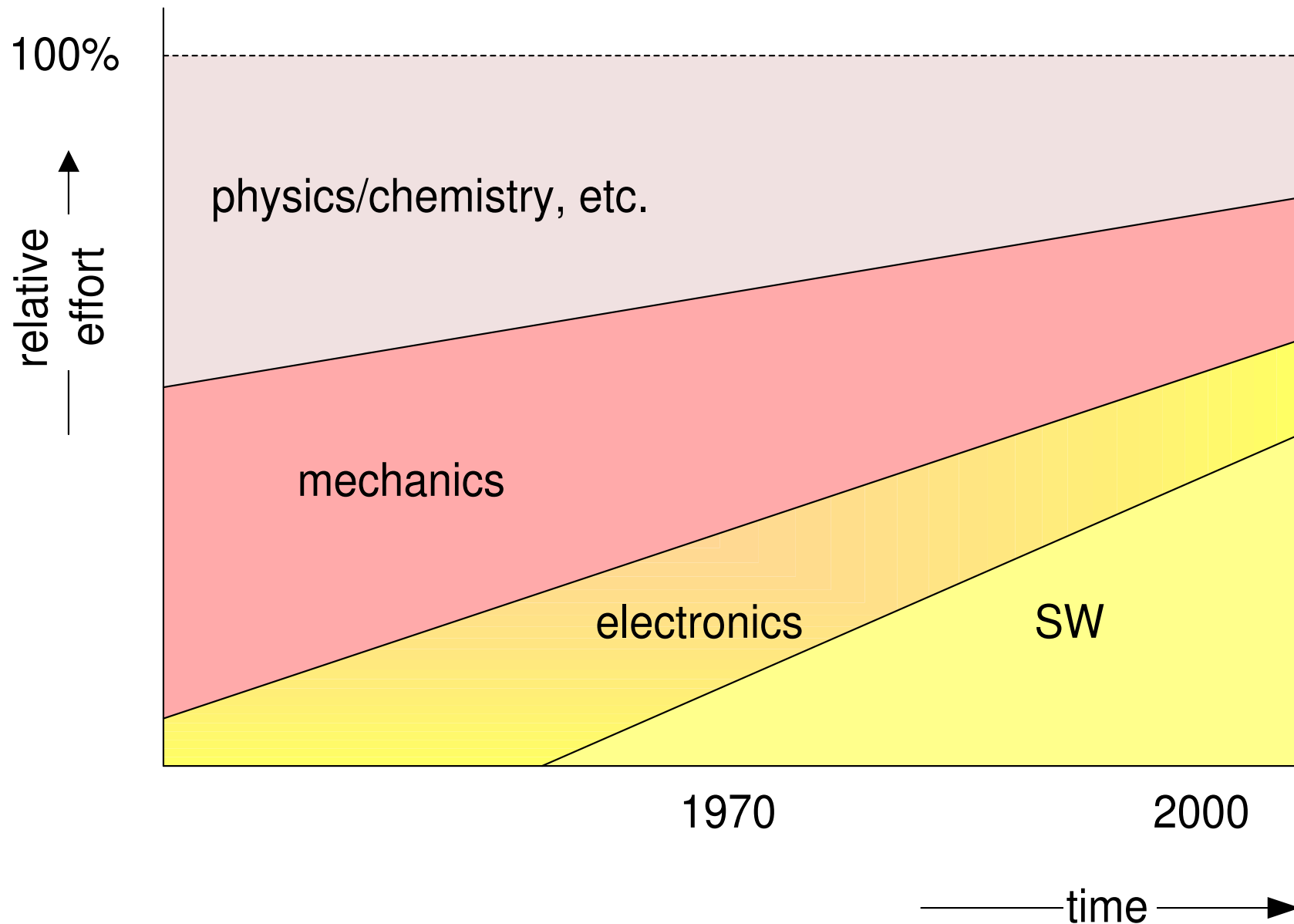
## Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

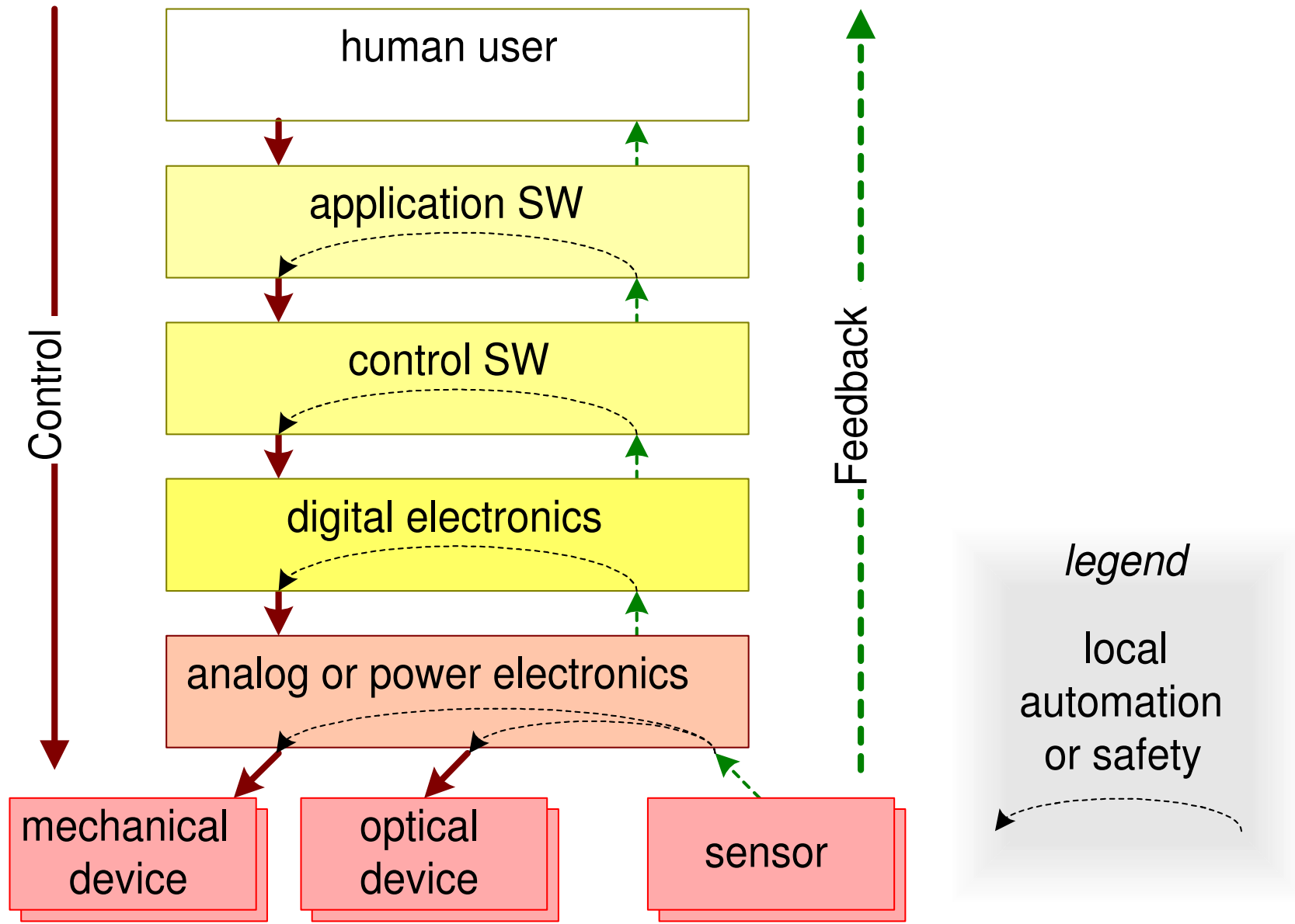
February 11, 2012  
status: concept  
version: 1.2



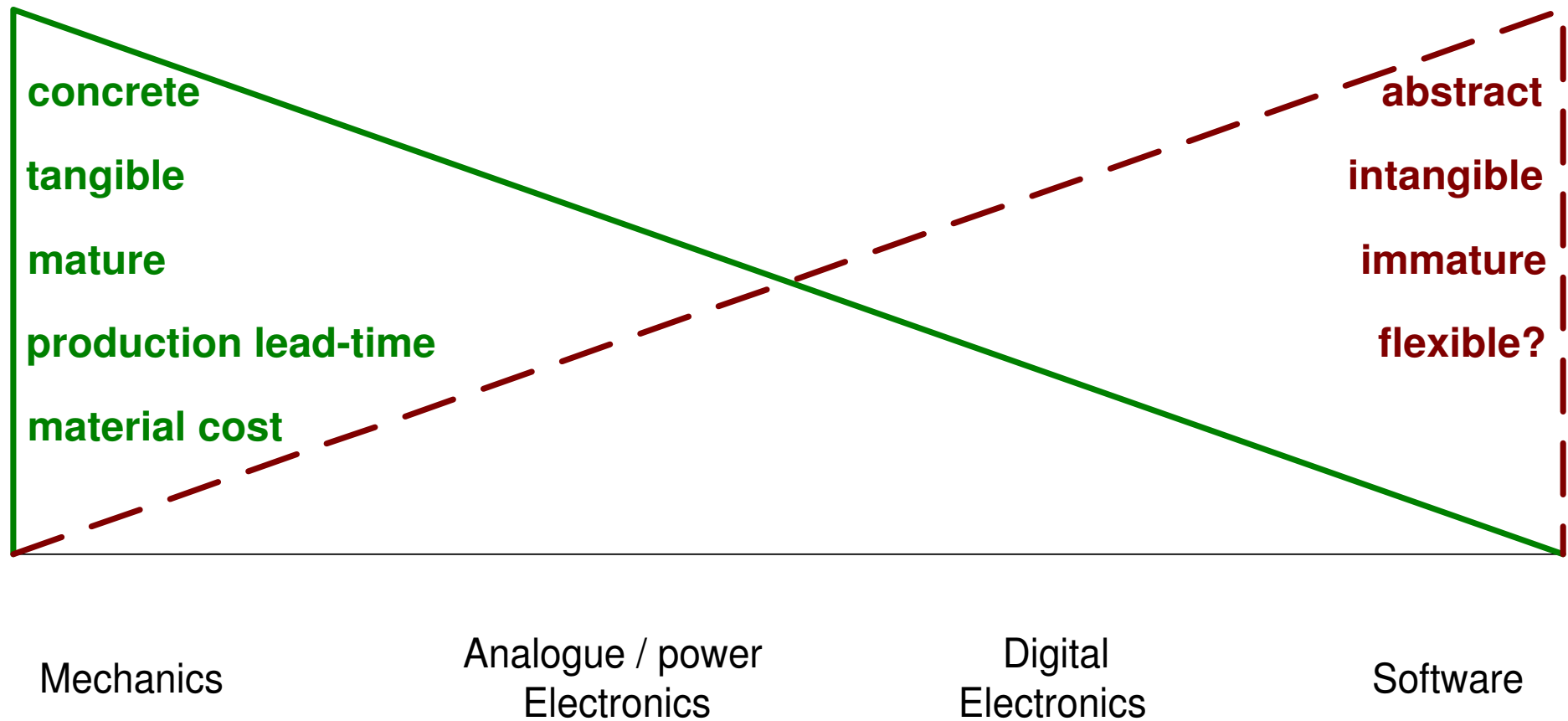
# Relative Contribution of SW



# Control Hierarchy along Technology axis



# Characterization of disciplines



# Quality Attributes annotated with SW relation

## usable

usability  
attractiveness  
responsiveness  
image quality

wearability  
storability  
transportability

## dependable

safety  
security  
reliability  
robustness  
integrity  
availability

## effective

throughput or  
productivity

## interoperable

connectivity  
3<sup>rd</sup> party extendible

## liable

liability  
testability  
traceability  
standards compliance

## efficient

resource utilization  
cost of ownership

## consistent

reproducibility  
predictability

## serviceable

serviceability  
configurability  
installability

## future proof

evolvability  
portability  
upgradability  
extendibility  
maintainability

## logistics friendly

manufacturability  
logistics flexibility  
lead-time

## ecological

ecological footprint  
contamination  
noise  
disposability

## down-to-earth attributes

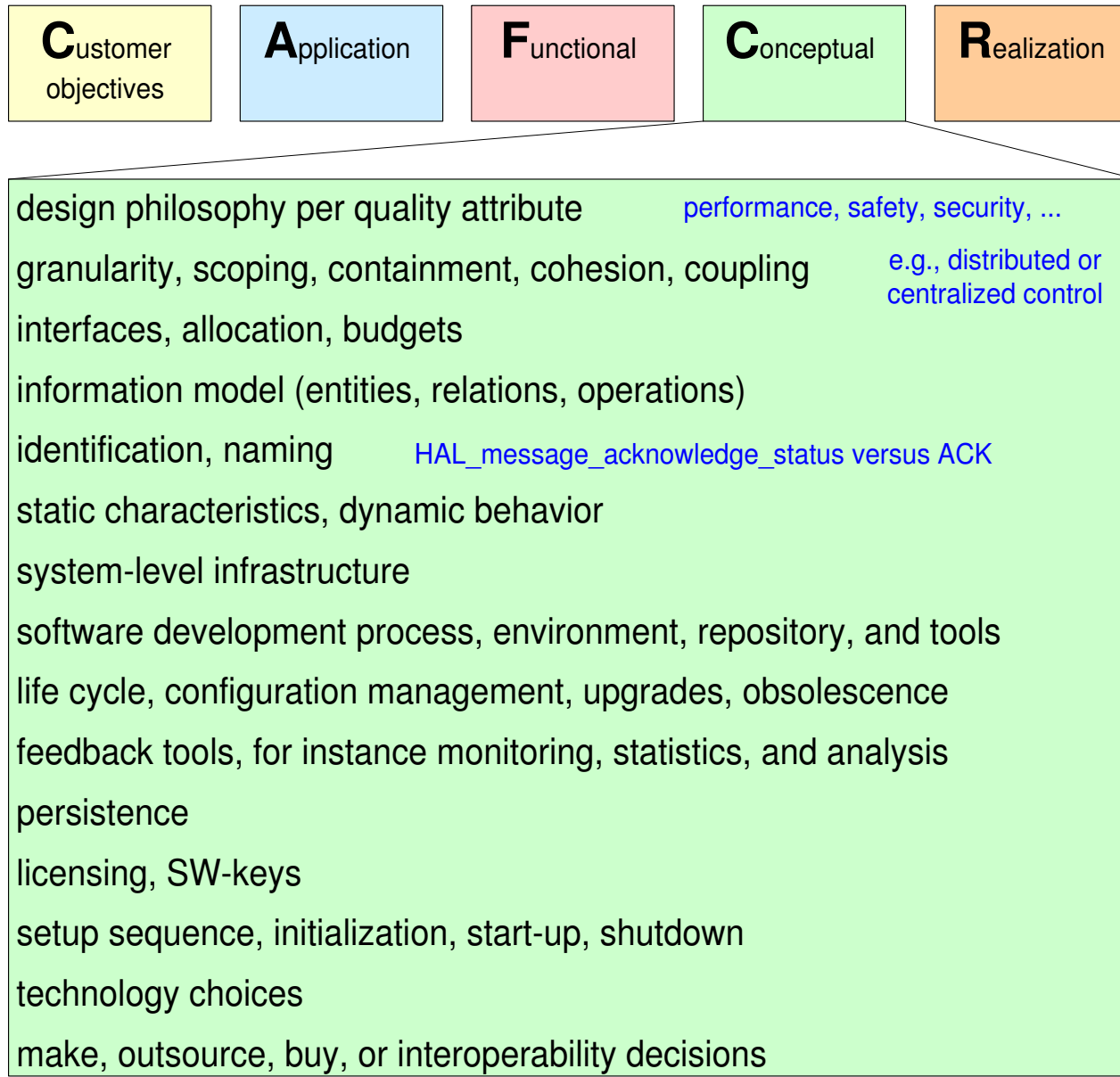
cost price  
power consumption  
consumption rate  
(water, air,  
chemicals,  
etc.)  
size, weight  
accuracy

### legend

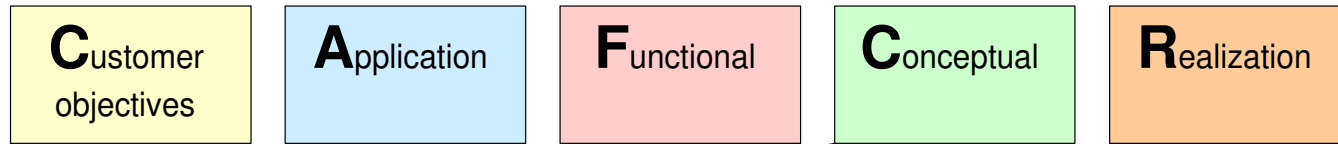
weak SW relation

strong SW relation

# Design Aspects related to SW



# SW Mechanisms



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

# Exercise Role of Software in a complex product

---

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