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 discusses 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.

May 29, 2023
status: concept
version: 1.3
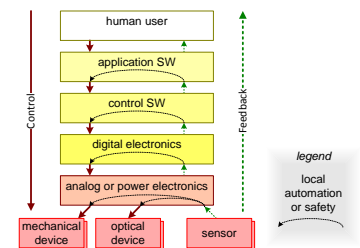
Relative Contribution of SW
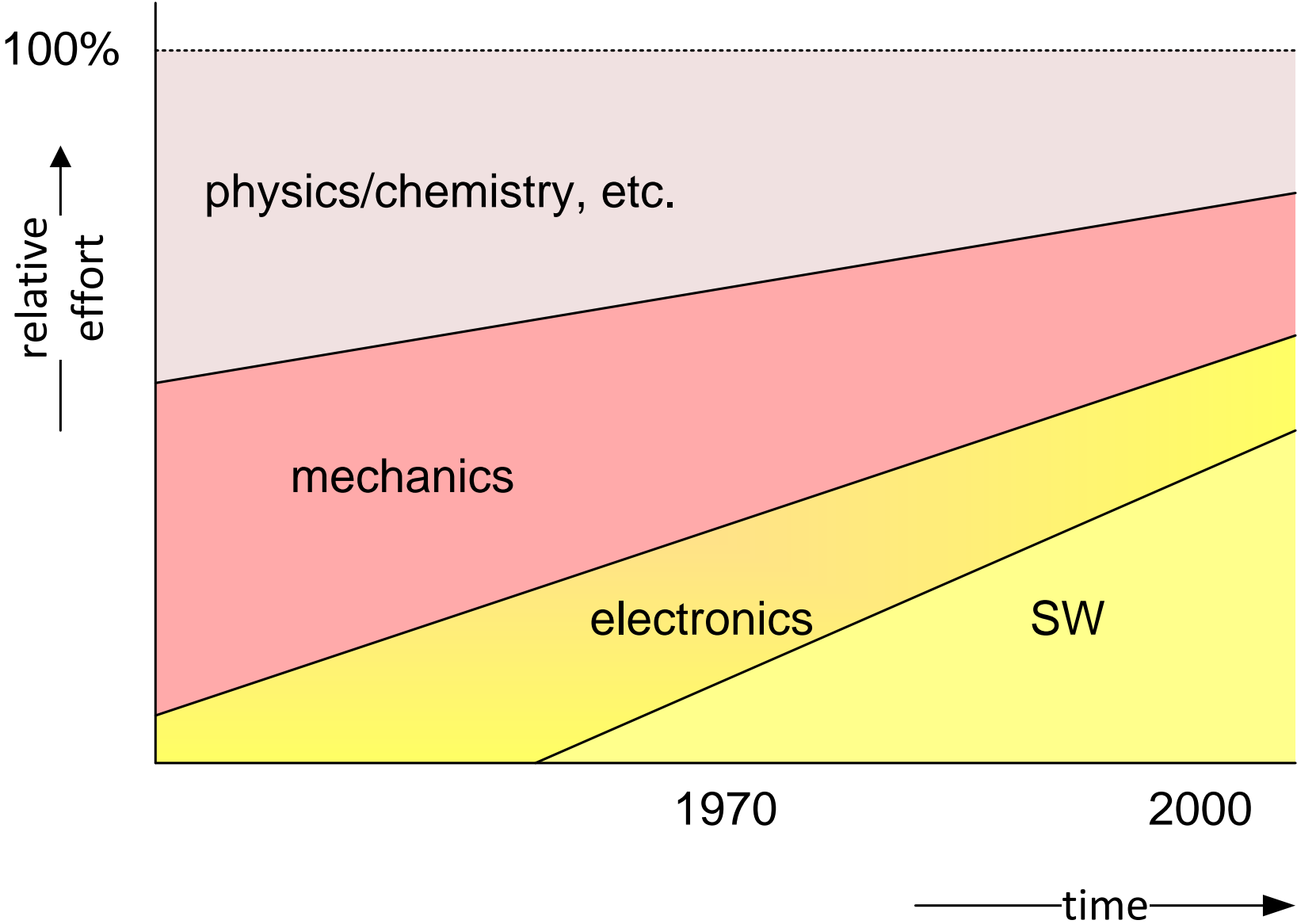
Mismatch between Role and Discipline

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

Control

Feedback

legend

local automation or safety
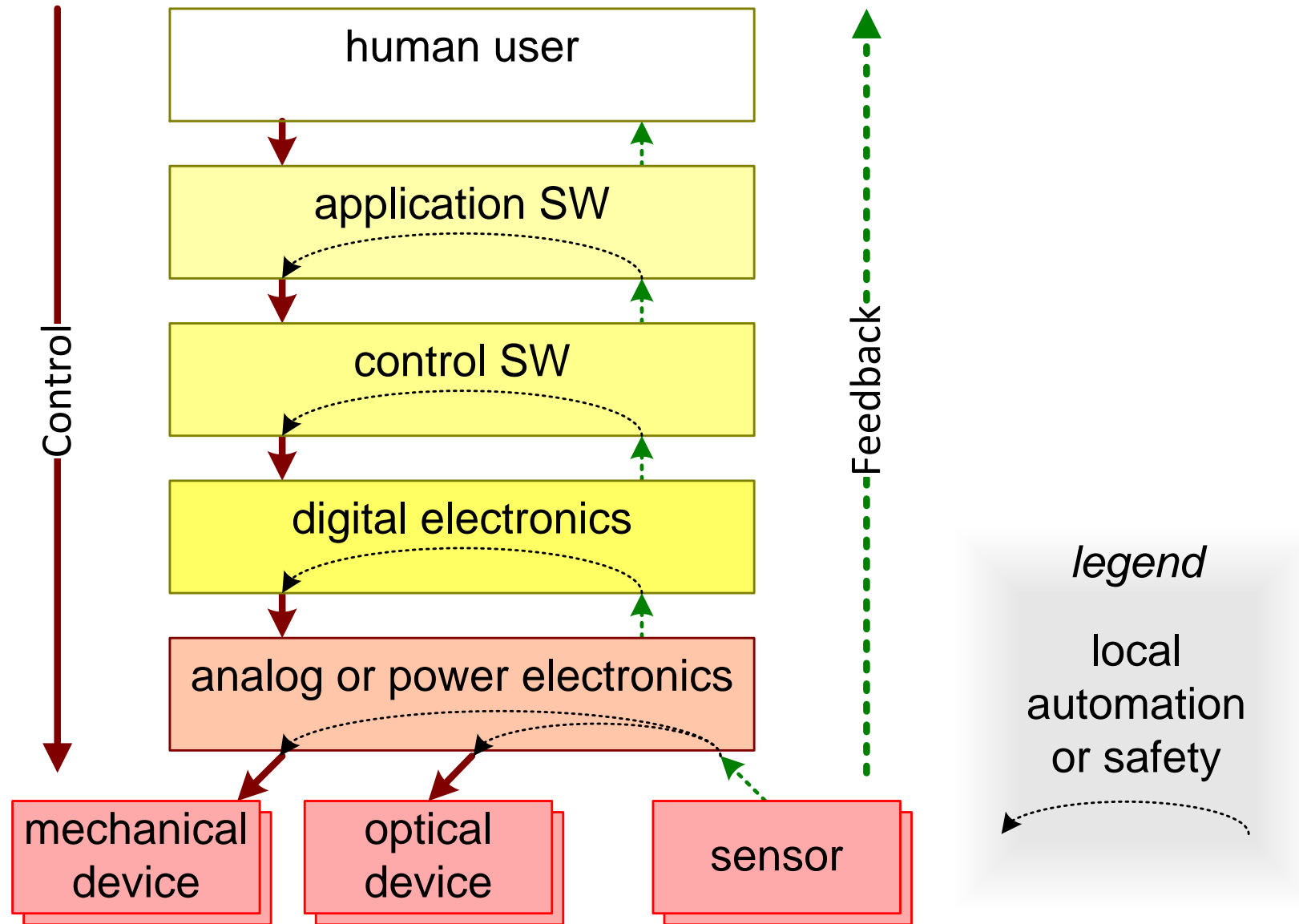
Characterization of disciplines

concrete — abstract

tangible — intangible

mature — immature

production lead-time — instanteneous deployment

material cost — flexible?

| Mechanics Optics | Analogue / power Electronics | Digital Electronics | Software Data |

USN   ESI
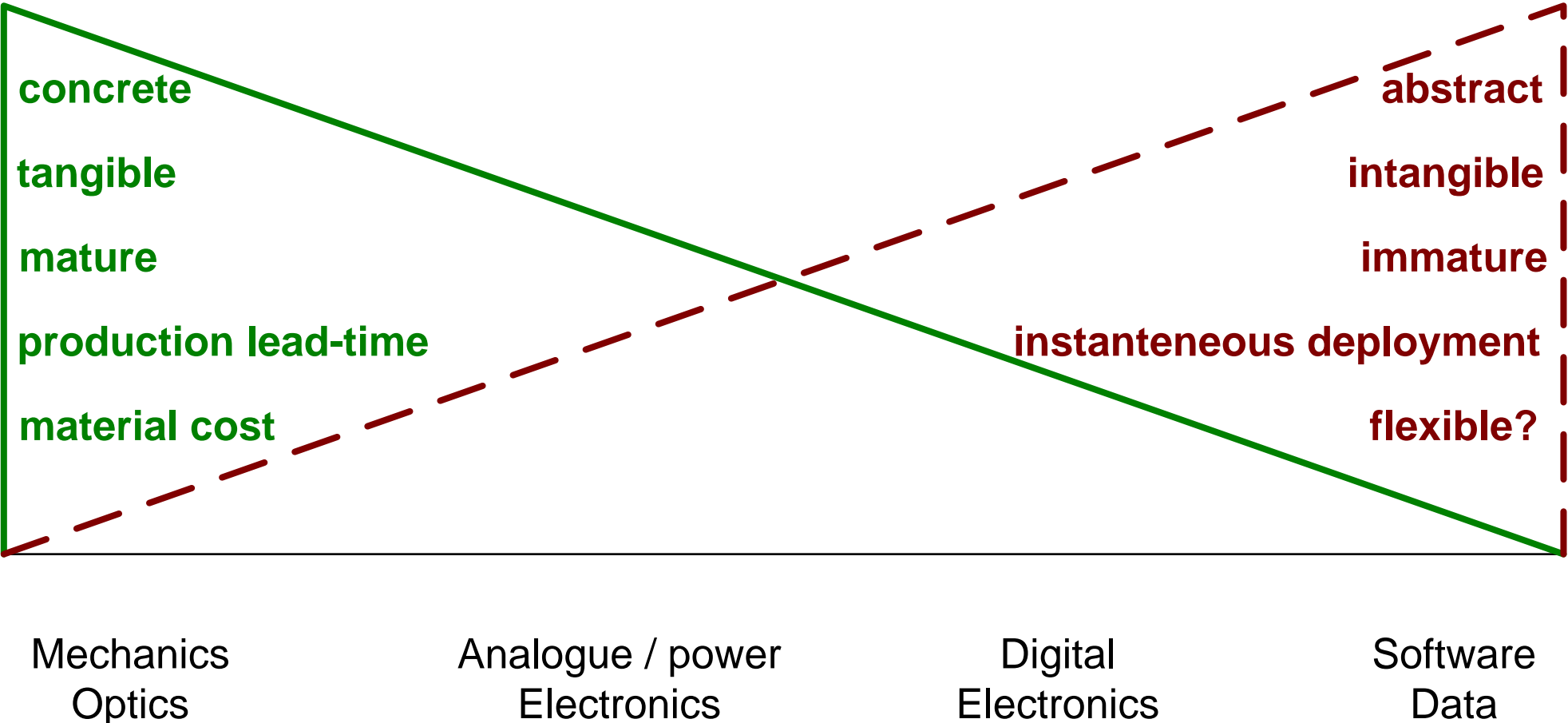
Quality Attributes annotated with SW relation

usable

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

dependable

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

effective

- throughput or productivity

interoperable

- connectivity
- 3rd 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

USN ESI

Design Aspects related to SW

Customer objectives	Application	Functional	Conceptual	Realization

design philosophy per quality attribute   performance, safety, security, ...

granularity, scoping, containment, cohesion, coupling   e.g., distributed or centralized control

interfaces, allocation, budgets

information model (entities, relations, operations)

identification, naming   HAL_message_acknowledge_status versus ACK

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

USN ESI

SW Mechanisms

Customer objectives	**A**pplication	**F**unctional	**C**onceptual	**R**ealization

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

USN ESI