

# Module Functional View

by *Gerrit Muller* Embedded Systems Institute

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

`www.gaudisite.nl`

## Abstract

This module addresses the Functional View.

### 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: draft

version: 0

logo  
TBD

# The functional view

by *Gerrit Muller* Embedded Systems Institute

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

`www.gaudisite.nl`

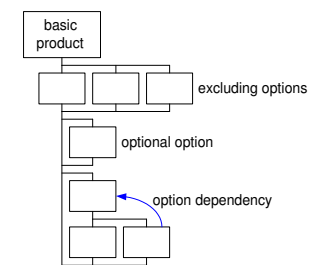
## Abstract

The purpose of the functional view is described. A number of methods or models is given to use in this view: (use) case descriptions, commercial decomposition function and feature specifications performance models and specifications, information models. The role of standards is discussed.

## 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



# Example personal video recorder use case contents

## typical use case(s)

### interaction flow (functional aspects)

- select movie via directory
- start movie
- be able to pause or stop
- be able to skip forward or backward
- set recording quality

### performance and other qualities (non-functional aspects)

- response times for start / stop
- response times for directory browsing
- end-of-movie behaviour
- relation recording quality and storage

## worst case, exceptional, or change use case(s)

### functional

- multiple inputs at the same time
- extreme long movie
- directory behaviour in case of  
extreme many short movies

### non-functional

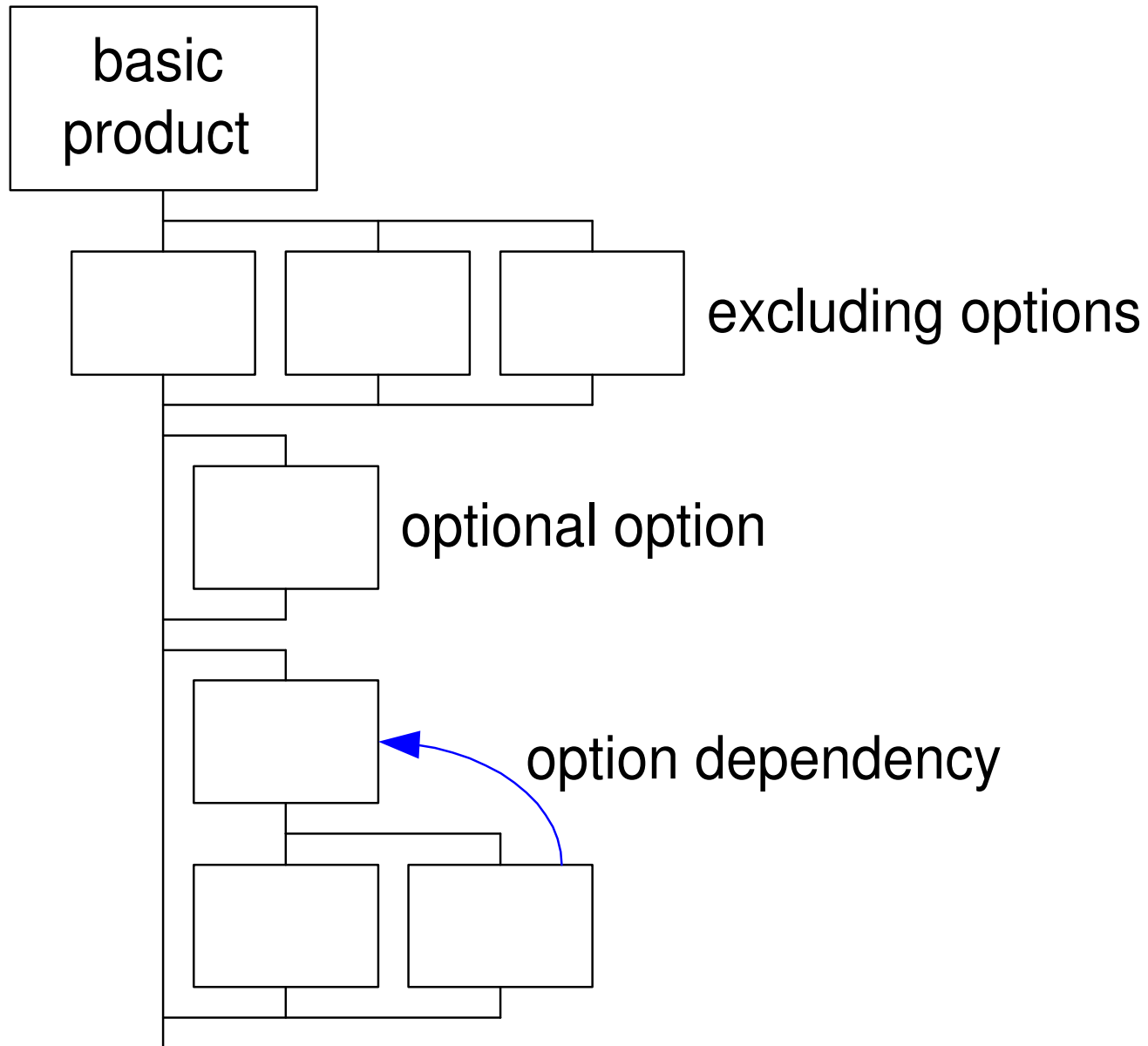
- response time with multiple inputs
- image quality with multiple inputs
- insufficient free space
- response time with many directory entries
- replay quality while HQ recording

# Recommendations for working with use cases

---

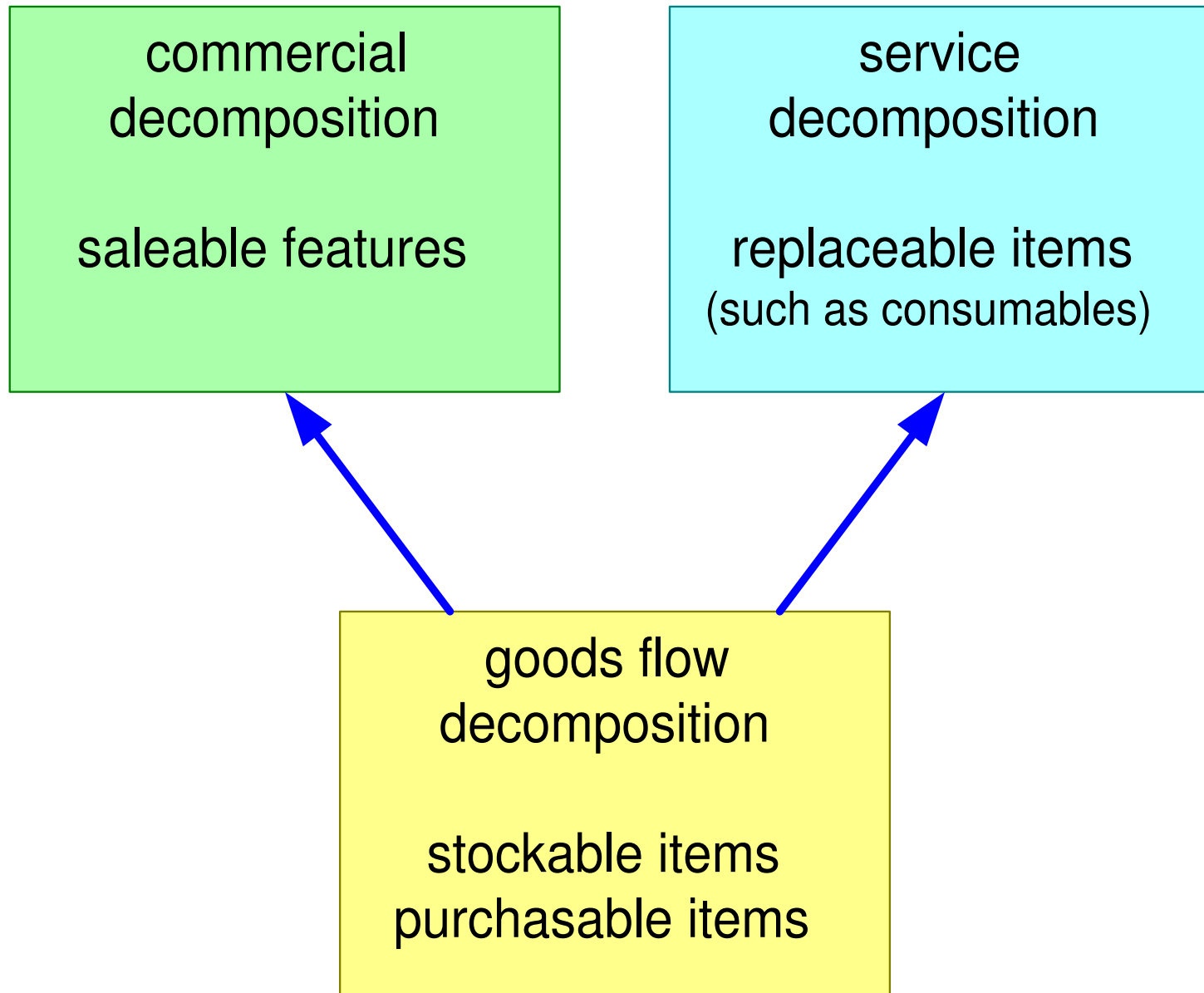
- + combine related functions in one use case
- do not make a separate use case for every function
- + include non-functional requirements in the use cases
  
- + minimise the amount of required *worst case* and *exceptional use cases*
- excessive amounts of use cases propagate to excessive implementation efforts
- + reduce the amount of these use cases in steps
- a few well chosen *worst case* use cases simplifies the design

# Commercial Decomposition



# Logistic decompositions for a product

---



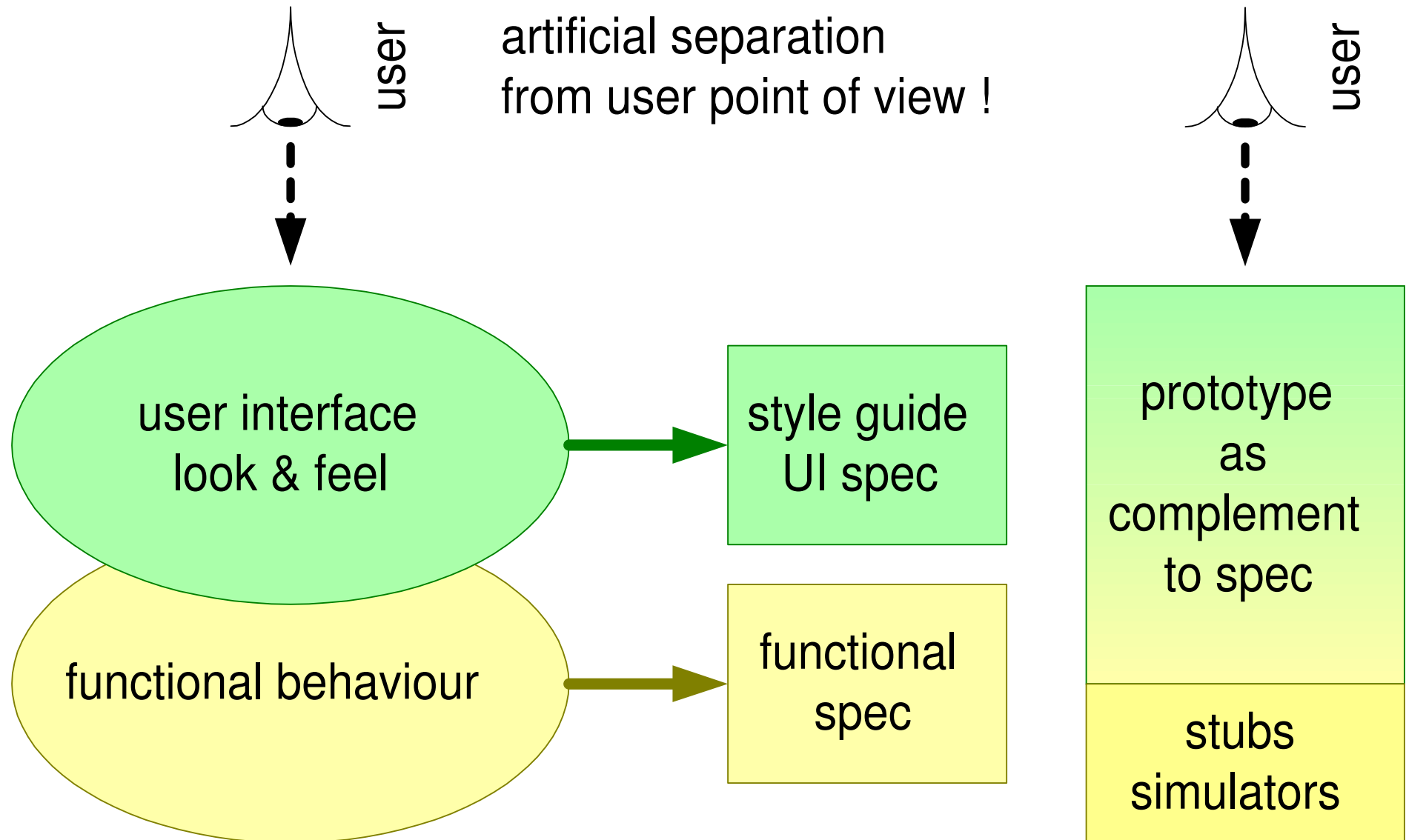
# Mapping technical functions on products

<i>technical functions</i>	<i>products</i>	home cinema system	flat screen cinema TV	bedroom TV
HD display		+	+	-
SD->HD up conversion		+	+	-
HD->SD down conversion		+	+	0
HD storage		0	-	-
SD storage		0	-	0
HD IQ improvement		+	+	-
SD IQ improvement		+	+	+
HD digital input		+	+	0
SD digital input		+	+	0
SD analog input		0	+	+
6 HQ channel audio		+	0	-
2 channel audio		-	+	+

## legend

+	present
0	optional
-	absent

# Relation between user interface and functional specification



# Layering of information definitions

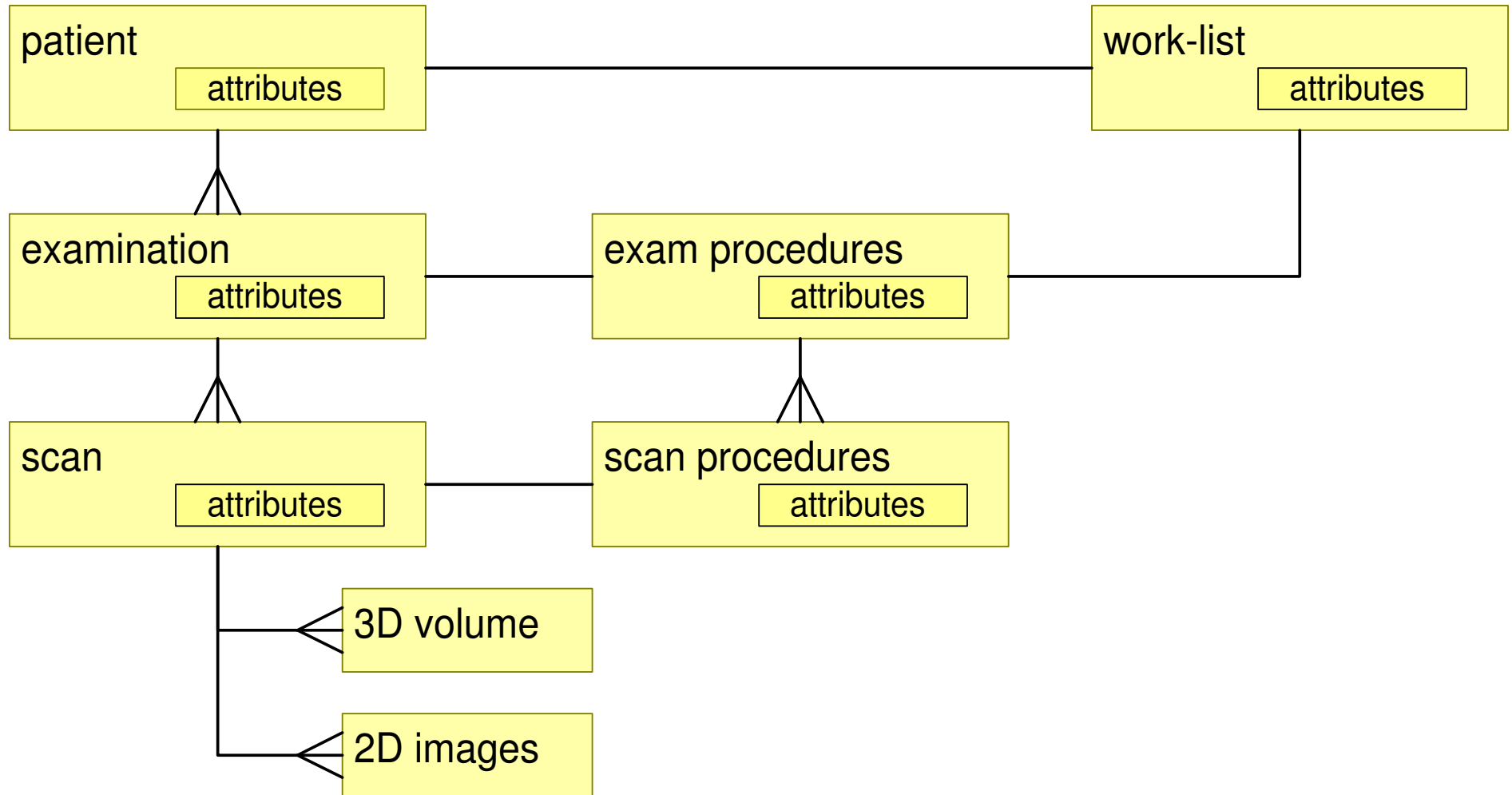
---

human understanding  
and interpretation  
of the information

information model , semantic defined in  
terms of:  
entities  
relations  
operations

data model or data dictionary  
identifiers  
types  
ranges

# Example partial internal information model



## 12 bit Image:

nx: 16 bit unsigned integer

ny: 16 bit unsigned integer

pixels[nx][ny]: 16 bit unsigned integers [0..4095]

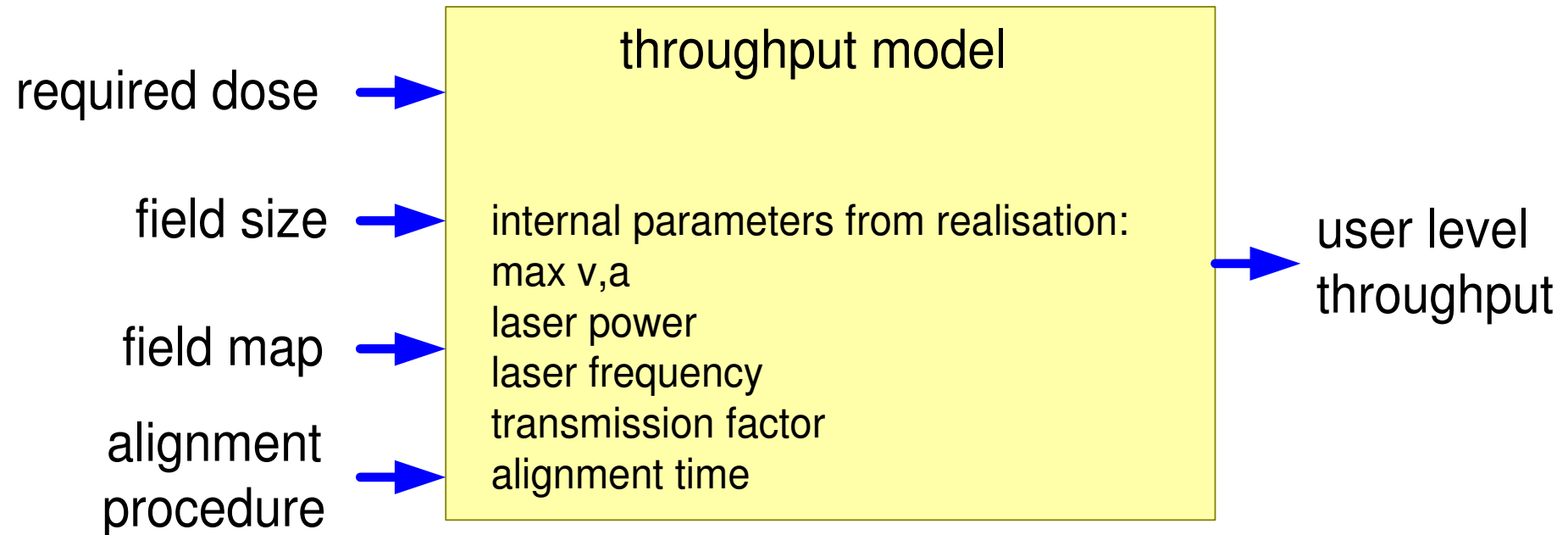
## 16 bit Image:

nx: 16 bit unsigned integer

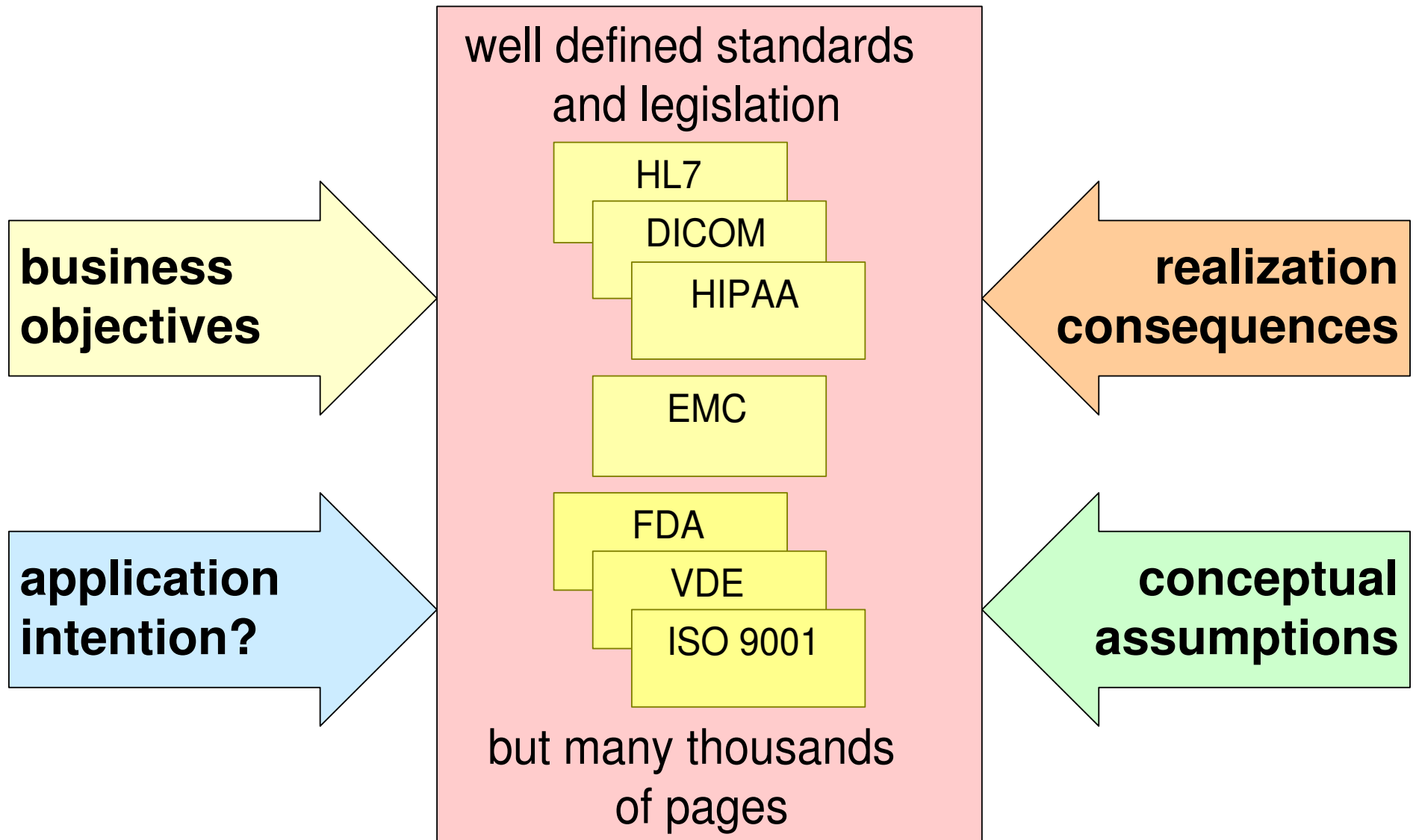
ny: 16 bit unsigned integer

pixels[nx][ny]: 16 bit unsigned integers

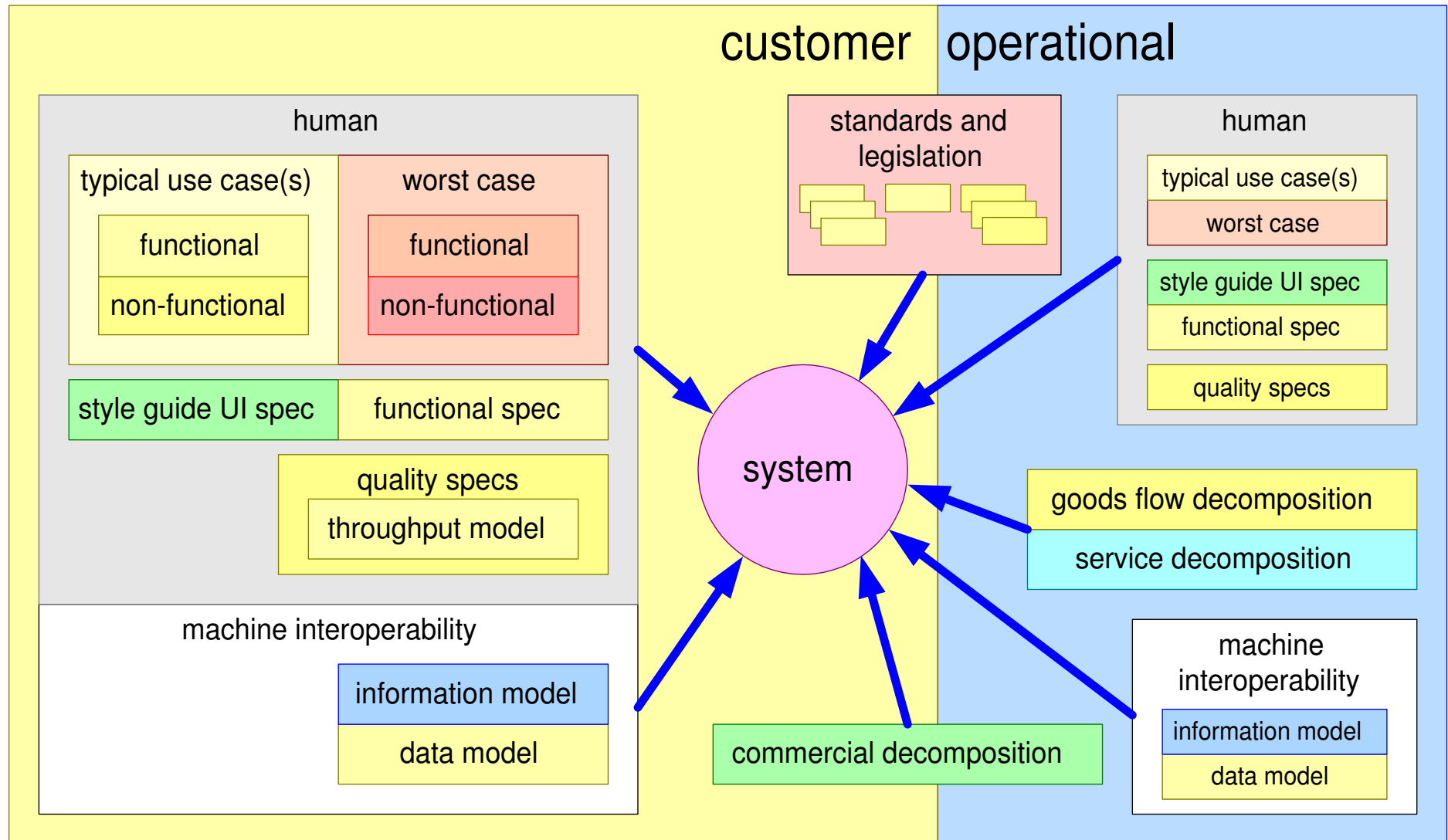
# Example of performance modelling



# The role of standards



# Functional view summary



Functional view = What: externally observable

# Exercise Functional View

---

- Make an overview of functions, performance figures, interfaces and optional features
- identify "most important" (related to CA-views)
- identify "most challenging" (related to CR-views)
- explain why "most important" or "most challenging"
- present in 5 minutes

## Goals:

- create awareness of the breadth of the specification
- share the spec with the team

- create a "living" image of the Functional view

# Exercise Functional View, second iteration

---

- Define a typical case, both functions and quantitative
- Create a single page product specification
- Define a worst case, suitable for design exploration and verification