A3 Architecture Overviews

Sharing Architecture Knowledge

From: Information Overload

To: Shared view of the system

P. D. Borches

University of Twente / Philips Healthcare

Outline

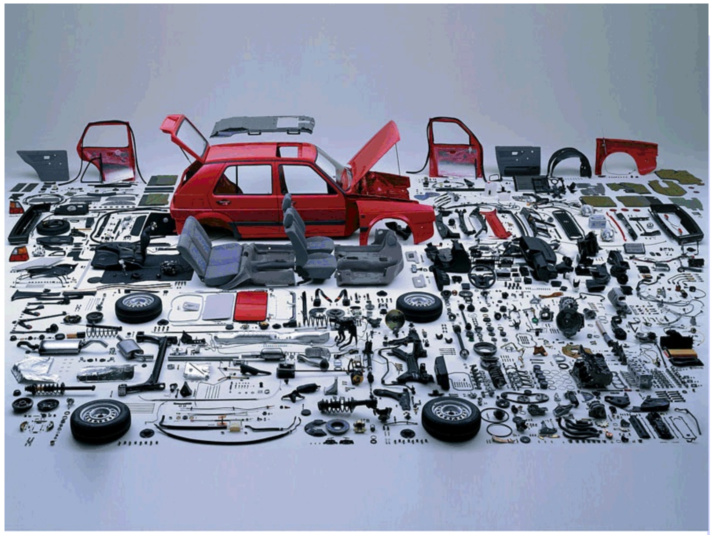
- Problem Background: Evolution of Complex Systems
- Study Case: Philips MRI System
- System Evolution Barriers (Philips' Survey)
- A3 Architecture Overview Example
- Reverse Architecting: Collect, Abstract, Present
- A3 Architecture Overviews
- Application: Philips System Design Specification (SDS) New Style
- Lessons Learned

Evolution of Complex Systems

•Designing **complex systems** from scratch is time consuming and costly -> **evolve existing systems.**

•Understanding the existing system is essential to improve it -> **reuse of knowledge**

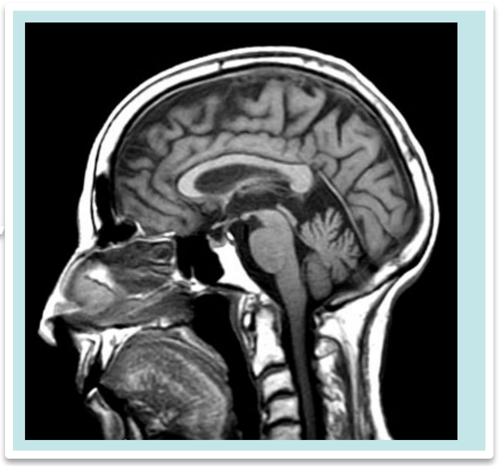
•Local changes have unforeseen system impacts -> **System perspective**



An easy to use **system representation**,
to effectively capture and share knowledge is needed

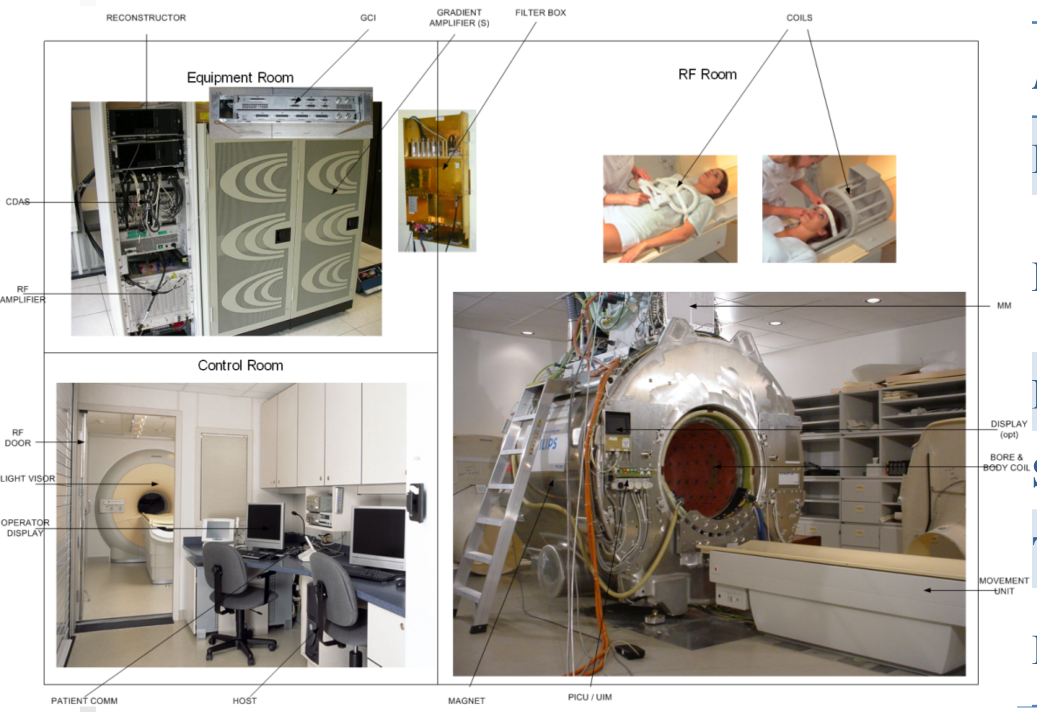
MRI System (I)

- **Magnetic resonance imaging** (**MRI**), is a medical imaging technique to visualize the structure and function of the body.
- It provides detailed images of the body in any plane.



MRI System (II)

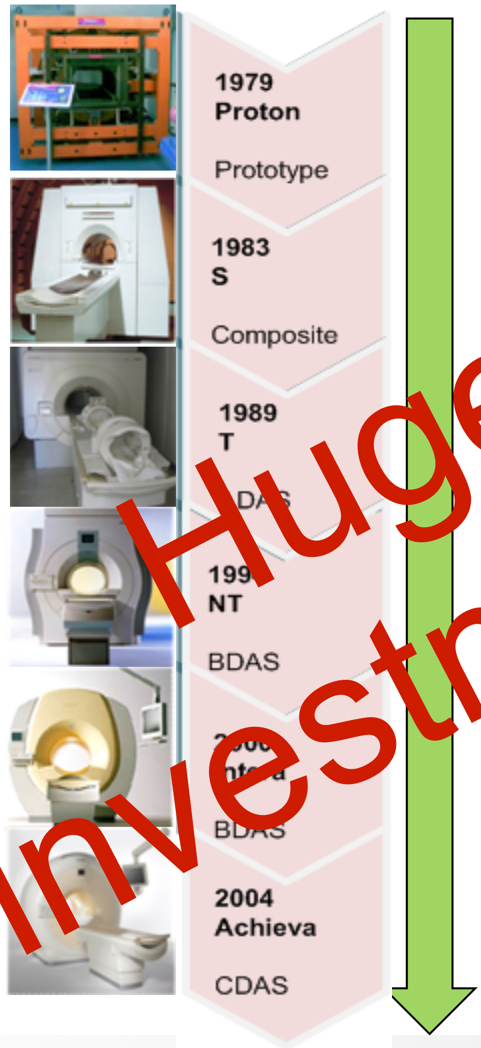
- Is the MRI a complex system?



Parameter	*Value*
Developers	~250
Disciplines	Physics, Mechanics, Electronics, Software, Medical applications etc.
Development sites	3
Subsidiary sites	All around the world
Technologies	~50
Lines of SW code	7*10^6 (~10 different languages)

Takes several years to release a new MRI in the market!

MRI Evolution



1979: first prototyp

1983: first product

1989: T-line: First i
architecture; ADAS

1994: NT-line: para n;
X-Windows UI; BDAS

2000: Intera-line: interactive

2002: WNT platform; WNT-UI (VT-emulator for Scan-UI)

2004: optimized workflow; optimized GUI; CDAS

2012: ?; DDAS

Huge investments!

Outline

- Problem Background: Evolution of Complex Systems
- Study Case: Philips MRI System
- System Evolution Barriers (Philips’ Survey)
- A3 Architecture Overview Example
- Reverse Architecting: Collect, Abstract, Present
- A3 Architecture Overviews
- Application: Philips System Design Specification (SDS) New Style
- Lessons Learned

Philips MRI Division Survey

- Identify development problems practitioners face when evolving a complex system such as the MRI:
  - **System complexity** increases in each evolution, making new developments more difficult (77%).
  - Lack of **system overview,** hard to estimate system impact of local changes (74%).
  - **Poor communication** across disciplines and departments (71%)
  - Lack of **Knowledge** Sharing lead to development errors or poor decisions (for 23% people in a monthly basis, 60% in a yearly basis)
  - Too much information, but **hard to find system information**(57%)

Outline

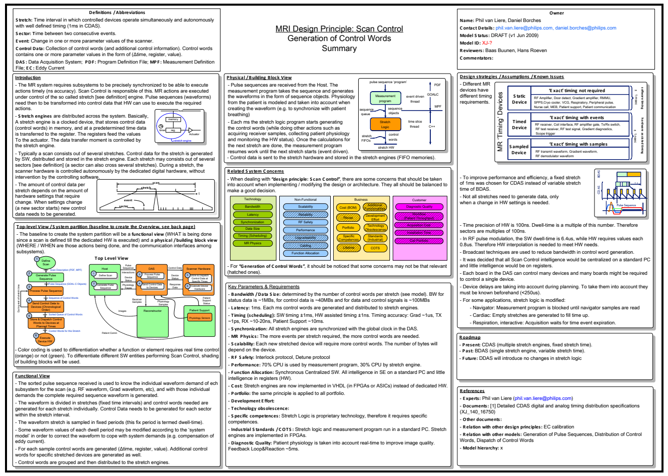
- Problem Background: Evolution of Complex Systems
- Study Case: Philips MRI System
- System Evolution Barriers (Philips' Survey)
- A3 Architecture Overviews Example
- Reverse Architecting: Collect, Abstract, Present
- A3 Architecture Overviews
- Application: Philips System Design Specification (SDS) New Style
- Lessons Learned

A3 Architecture Overviews

A3 Architecture Overviews: Manageable system description of key aspects of the system in an easy to use way.

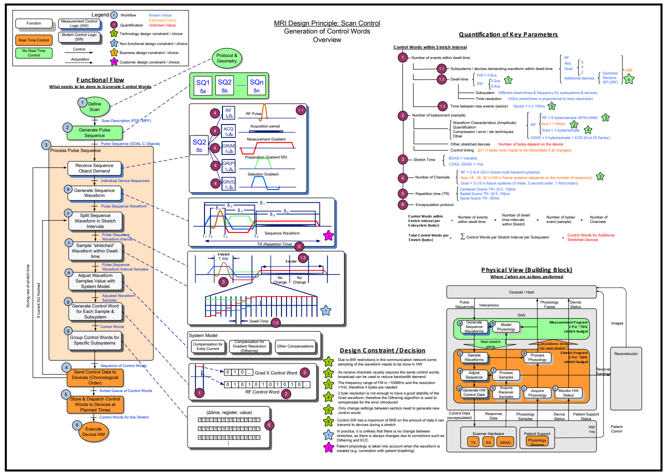
Traditional System Description

A3 Architectural Overview(s)



MRI Design Principle: Scan Control
Generation of Control Words
Summary

A3 Sheet (Text)

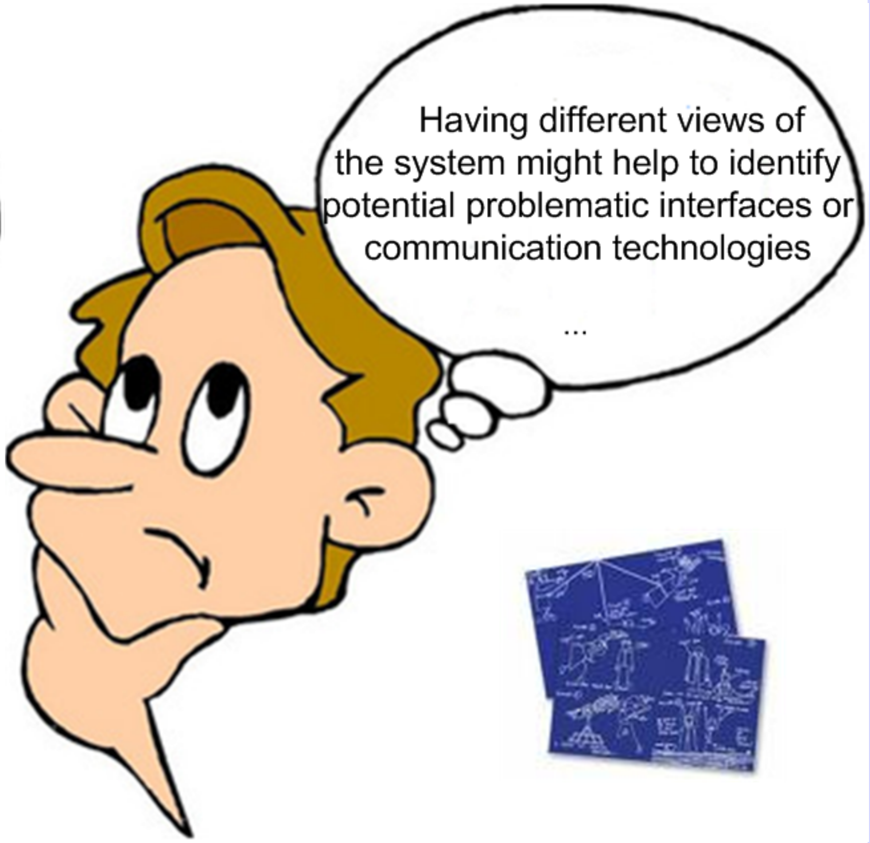


MRI Design Principle: Scan Control
Generation of Control Words
Overview
Functional Flow
Quantification of Key Parameters
Physical View (Building Block)
Design Constraint / Decision

A3 Sheet (Model)

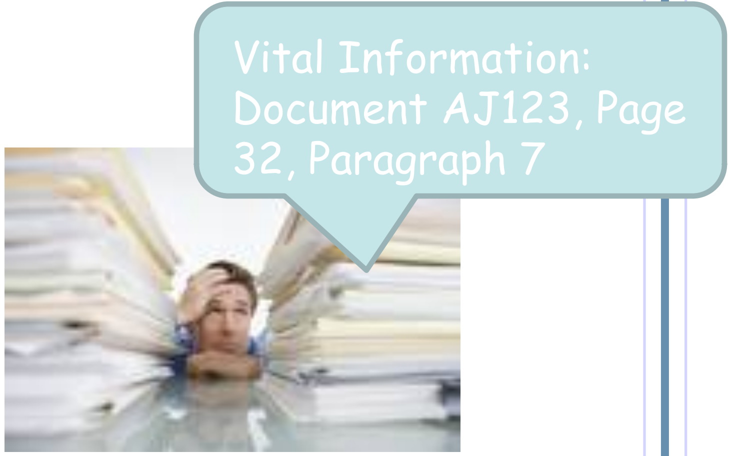
A3 Architecture Overview Motivation

How is the MRI working?
What communication is needed?
Which functions are required to perform a scan?
...


Having different views of the system might help to identify potential problematic interfaces or communication technologies
...

A3 Architecture Overview Goals

- **Systems Viewpoint:** Understand the system as a whole
- **Manage Complexity**: Provide only the relevant information, and no more (overview).
- **Effective Communication:** Simple, yet effective ways to communicate (drawings + text).
- **Integrate Multiple Views:** Benefit from having different views within the same sheet.
- **Synthesis and Visualization**: Encourage brevity by limiting space and using visual representations.



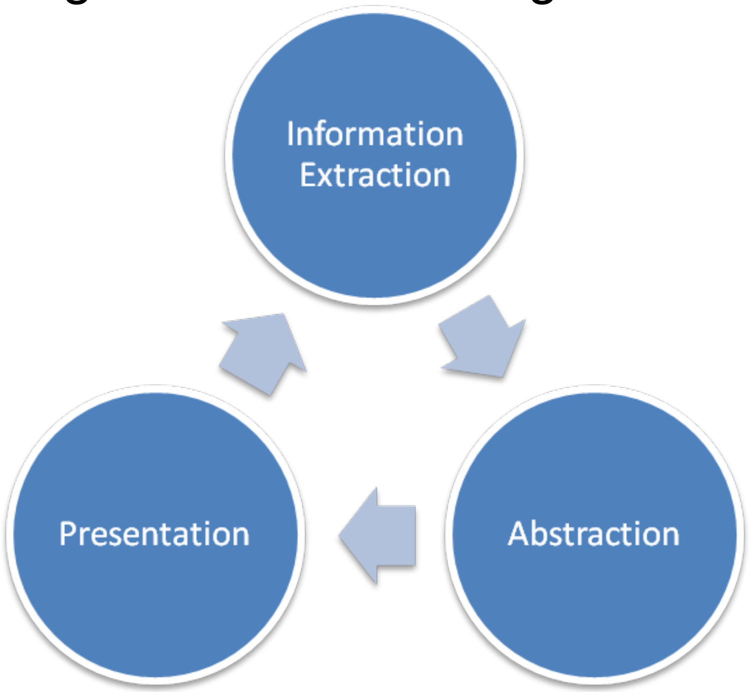
Outline

- Problem Background: Evolution of Complex Systems
- Study Case: Philips MRI System
- System Evolution Barriers (Philips' Survey)
- A3 Architecture Overview Example
- Reverse Architecting: Collect, Abstract, Present
- A3 Architecture Overviews
- Application: Philips System Design Specification (SDS) New Style
- Lessons Learned

Reverse Architecting

•Key knowledge is often not documented nor explicit but in expert’s minds.

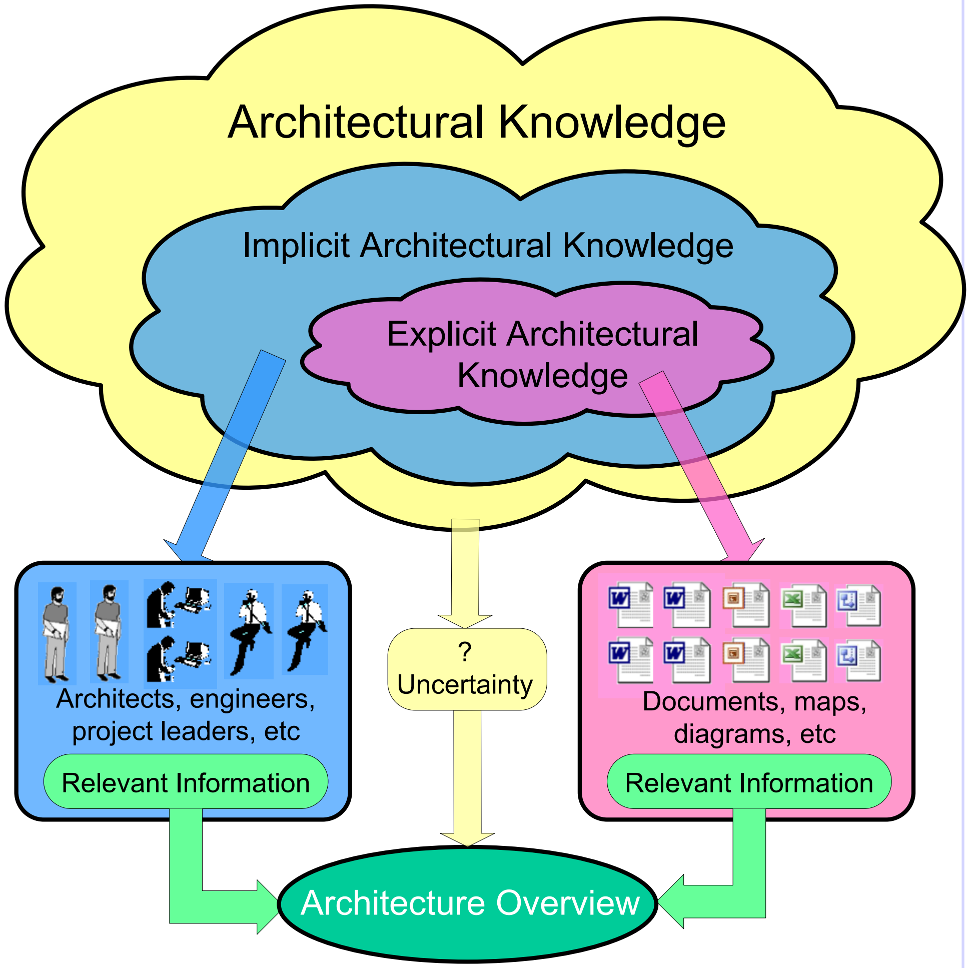
•Reverse architecting enables recovering this knowledge.



RA: Collect Information

- Challenges to Collect Information:
  - Knowledge spread within the company
  - Lost knowledge due to people leaving the company, decisions not documented, etc
  - Detailed / long documents rather than cross system concerns

An Architecture Overview provides the framework to know what/how to collect the information


Architectural Knowledge
Implicit Architectural Knowledge
Explicit Architectural Knowledge
Architects, engineers, project leaders, etc
Relevant Information
?
Uncertainty
Documents, maps, diagrams, etc
Relevant Information
Architecture Overview

RA: Abstraction

•Humans can handle limited amount of information.

•How to keep the essentials? -> A3 sheet

•Used by Toyota's Lean production system (A3 Reports)

•Reader can focus on a part, but always see the whole (Overview).

•Force synthesis and brevity of knowledge

<u>A3 Sheet Size</u>: Maximum amount of information most people is willing to read to get an overview.

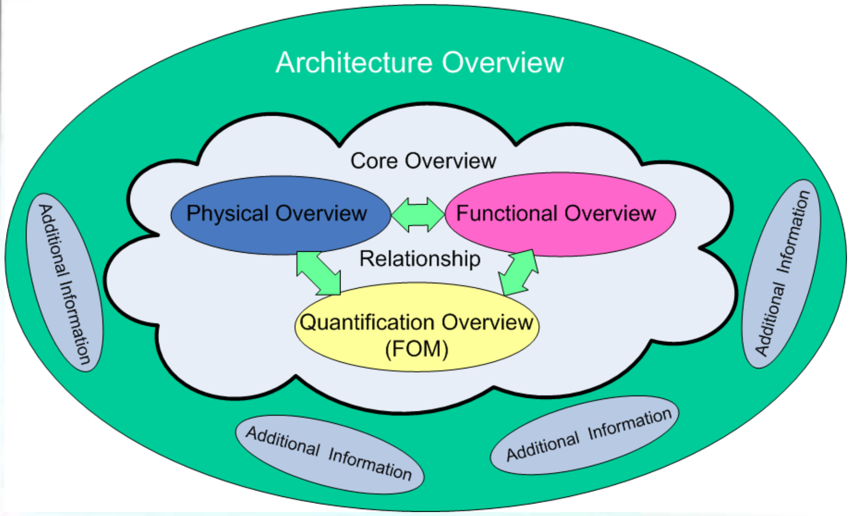
•<u>Guiding principle</u>: Include key information and eliminate everything else until only essentials remains.

RA: Presentation (I)

•What should an Architecture Overview contain?

•Different views, but not too many; physical, functional, quantification.

•Additional information; design decisions, solutions, etc

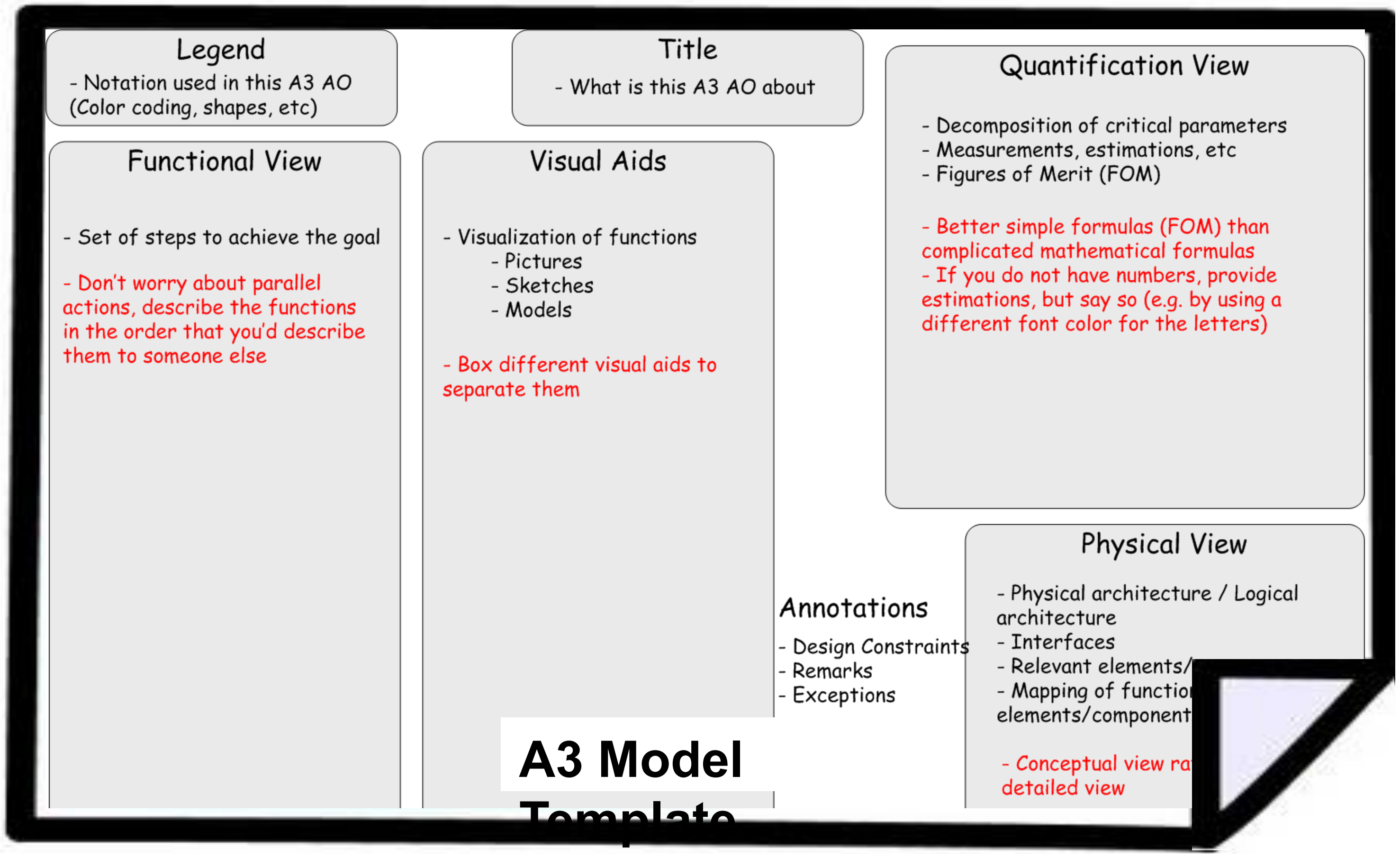


<u>Text and Model</u>

Text is necessary to support a model (prevent models to end up embedded in a document)

RA: Presentation (II)

Providing Structure to the A3 Improves Readability and Comprehension



Legend
- Notation used in this A3 AO (Color coding, shapes, etc)

Title
- What is this A3 AO about

Quantification View
- Decomposition of critical parameters
- Measurements, estimations, etc
- Figures of Merit (FOM)

- Better simple formulas (FOM) than complicated mathematical formulas
- If you do not have numbers, provide estimations, but say so (e.g. by using a different font color for the letters)

Functional View
- Set of steps to achieve the goal

- Don't worry about parallel actions, describe the functions in the order that you'd describe them to someone else

Visual Aids
- Visualization of functions
  - Pictures
  - Sketches
  - Models

- Box different visual aids to separate them

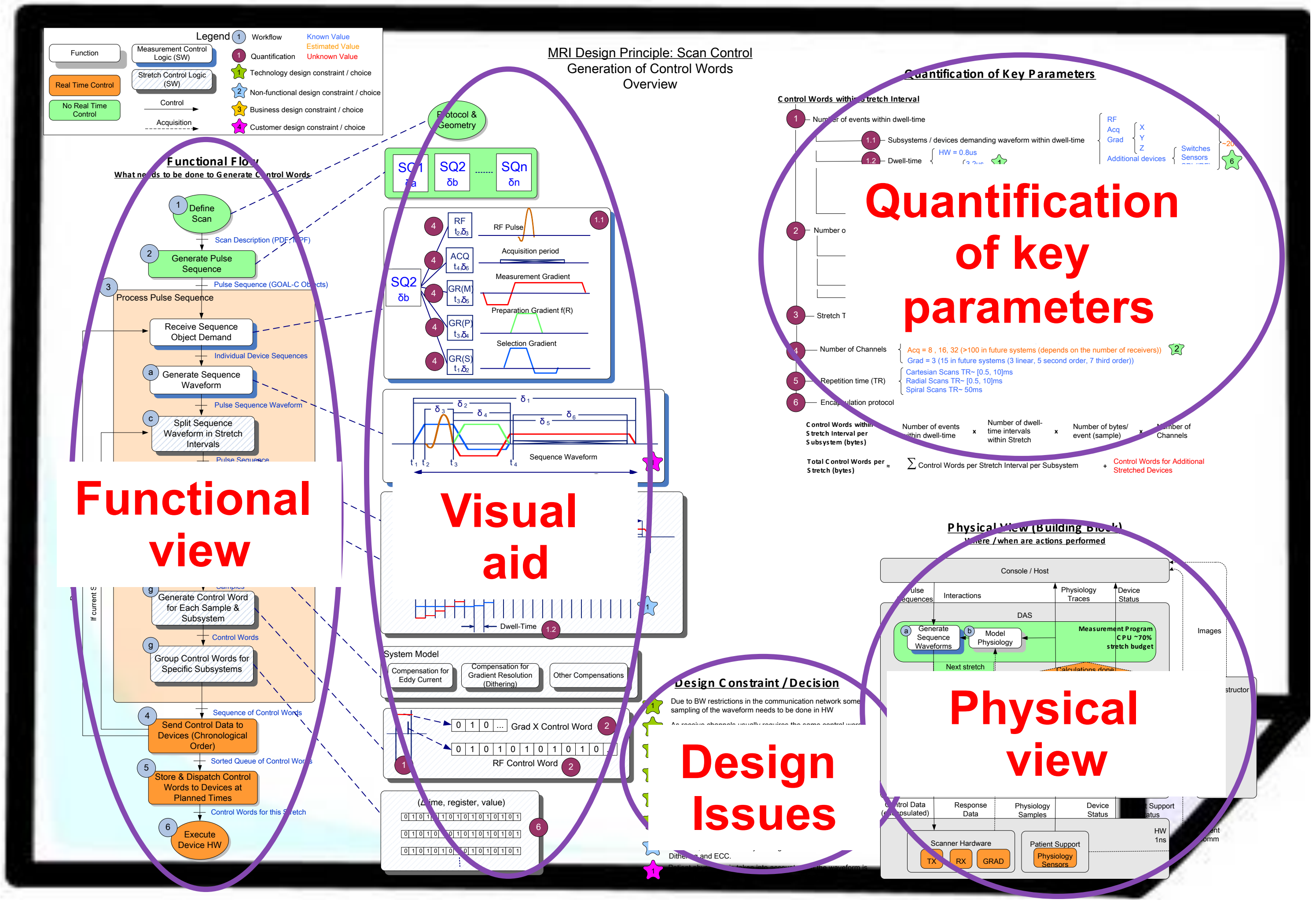
Physical View
- Physical architecture / Logical architecture
- Interfaces
- Relevant elements/
- Mapping of functio
elements/component

- Conceptual view ra
detailed view

Annotations
- Design Constraints
- Remarks
- Exceptions

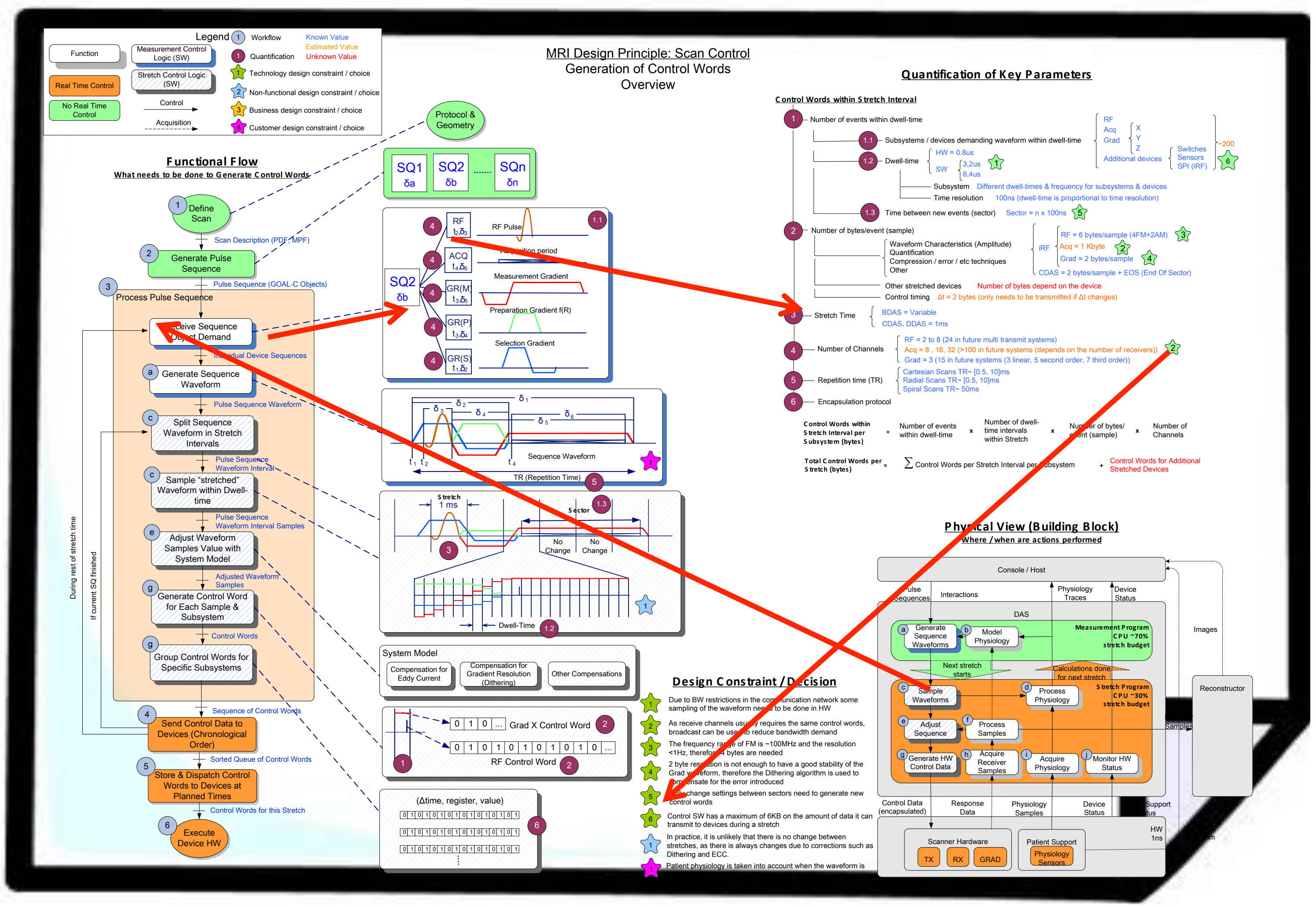
A3 Model Template

A3 Model View Example



Legend
Function
Measurement Control Logic (SW)
Real Time Control
Stretch Control Logic (SW)
No Real Time Control
Control
Acquisition
Workflow
Quantification
Technology design constraint / choice
Non-functional design constraint / choice
Business design constraint / choice
Customer design constraint / choice
Known Value
Estimated Value
Unknown Value
MRI Design Principle: Scan Control
Generation of Control Words
Overview
Functional Flow
What needs to be done to Generate Control Words
Define Scan
Generate Pulse Sequence
Process Pulse Sequence
Receive Sequence Object Demand
Generate Sequence Waveform
Split Sequence Waveform in Stretch Intervals
Generate Control Word for Each Sample & Subsystem
Group Control Words for Specific Subsystems
Send Control Data to Devices (Chronological Order)
Store & Dispatch Control Words to Devices at Planned Times
Execute Device HW
Protocol & Geometry
SQ1
SQ2
SQn
RF Pulse
Acquisition period
Measurement Gradient
Preparation Gradient f(R)
Selection Gradient
Sequence Waveform
Dwell-Time
System Model
Compensation for Eddy Current
Compensation for Gradient Resolution (Dithering)
Other Compensations
Grad X Control Word
RF Control Word
(time, register, value)
Quantification of Key Parameters
Control Words within Stretch Interval
Number of events within dwell-time
Subsystems / devices demanding waveform within dwell-time
Dwell-time
Number of Channels
Repetition time (TR)
Encapsulation protocol
Total Control Words per Stretch (bytes)
Control Words per Stretch Interval per Subsystem
Control Words for Additional Stretched Devices
Design Constraint / Decision
Physical View (Building Block)
Where / when are actions performed
Console / Host
DAS
Scanner Hardware
Patient Support
Images
Functional view
Visual aid
Quantification of key parameters
Design Issues
Physical view

A3 Model View Example



MRI Design Principle: Scan Control
Generation of Control Words
Overview
Legend
Function
Measurement Control Logic (SW)
Stretch Control Logic (SW)
Real Time Control
No Real Time Control
Control
Acquisition
Workflow
Quantification
Technology design constraint / choice
Non-functional design constraint / choice
Business design constraint / choice
Customer design constraint / choice
Known Value
Estimated Value
Unknown Value
Protocol & Geometry
Functional Flow
What needs to be done to Generate Control Words
Define Scan
Scan Description (PDF/MPF)
Generate Pulse Sequence
Pulse Sequence (GOAL-C Objects)
Process Pulse Sequence
Receive Sequence Object Demand
Individual Device Sequences
Generate Sequence Waveform
Pulse Sequence Waveform
Split Sequence Waveform in Stretch Intervals
Pulse Sequence Waveform Interval
Sample "stretched" Waveform within Dwell-time
Pulse Sequence Waveform Interval Samples
Adjust Waveform Samples Value with System Model
Adjusted Waveform Samples
Generate Control Word for Each Sample & Subsystem
Control Words
Group Control Words for Specific Subsystems
Sequence of Control Words
Send Control Data to Devices (Chronological Order)
Sorted Queue of Control Words
Store & Dispatch Control Words to Devices at Planned Times
Control Words for this Stretch
Execute Device HW
During rest of stretch time
If current SQ finished
SQ1 δa
SQ2 δb
SQn δn
RF Pulse
Acquisition period
Measurement Gradient
Preparation Gradient f(R)
Selection Gradient
Sequence Waveform
TR (Repetition Time)
Stretch 1 ms
Sector
No Change
No Change
Dwell-Time
System Model
Compensation for Eddy Current
Compensation for Gradient Resolution (Dithering)
Other Compensations
Grad X Control Word
RF Control Word
(Δtime, register, value)
Quantification of Key Parameters
Control Words within Stretch Interval
Physical View (Building Block)
Where / when are actions performed
Console / Host
Pulse Sequences
Interactions
Physiology Traces
Device Status
Images
DAS
Generate Sequence Waveforms
Model Physiology
Measurement Program CPU ~70% stretch budget
Next stretch starts
Calculations done for next stretch
Sample Waveforms
Process Physiology
Stretch Program CPU ~30% stretch budget
Adjust Sequence
Process Samples
Generate HW Control Data
Acquire Receiver Samples
Acquire Physiology
Monitor HW Status
Reconstructor
Samples
Control Data (encapsulated)
Response Data
Physiology Samples
Device Status
Support Status
HW 1ns
Scanner Hardware
TX
RX
GRAD
Patient Support
Physiology Sensors

Quantification of Key Parameters

Control Words within Stretch Interval

1. Number of events within dwell-time
   - 1.1 Subsystems / devices demanding waveform within dwell-time — RF, Acq, Grad (X, Y, Z), Additional devices (Switches, Sensors, SPI (iRF)) ~200
   - 1.2 Dwell-time — HW = 0.8us; SW 3,2us / 6,4us
     - Subsystem — Different dwell-times & frequency for subsystems & devices
     - Time resolution — 100ns (dwell-time is proportional to time resolution)
   - 1.3 Time between new events (sector) — Sector = n x 100ns
2. Number of bytes/event (sample)
   - Waveform Characteristics (Amplitude), Quantification, Compression / error / etc techniques, Other — iRF: RF = 6 bytes/sample (4FM+2AM); Acq = 1 Kbyte; Grad = 2 bytes/sample; CDAS = 2 bytes/sample + EOS (End Of Sector)
   - Other stretched devices — Number of bytes depend on the device
   - Control timing — Δt = 2 bytes (only needs to be transmitted if Δt changes)
3. Stretch Time — BDAS = Variable; CDAS, DDAS = 1ms
4. Number of Channels — RF = 2 to 8 (24 in future multi transmit systems); Acq = 8 , 16, 32 (>100 in future systems (depends on the number of receivers)); Grad = 3 (15 in future systems (3 linear, 5 second order, 7 third order))
5. Repetition time (TR) — Cartesian Scans TR~ [0.5, 10]ms; Radial Scans TR~ [0.5, 10]ms; Spiral Scans TR~ 50ms
6. Encapsulation protocol

Control Words within Stretch Interval per Subsystem (bytes) ≈ Number of events within dwell-time x Number of dwell-time intervals within Stretch x Number of bytes/ event (sample) x Number of Channels

Total Control Words per Stretch (bytes) ≈ ∑ Control Words per Stretch Interval per Subsystem + Control Words for Additional Stretched Devices

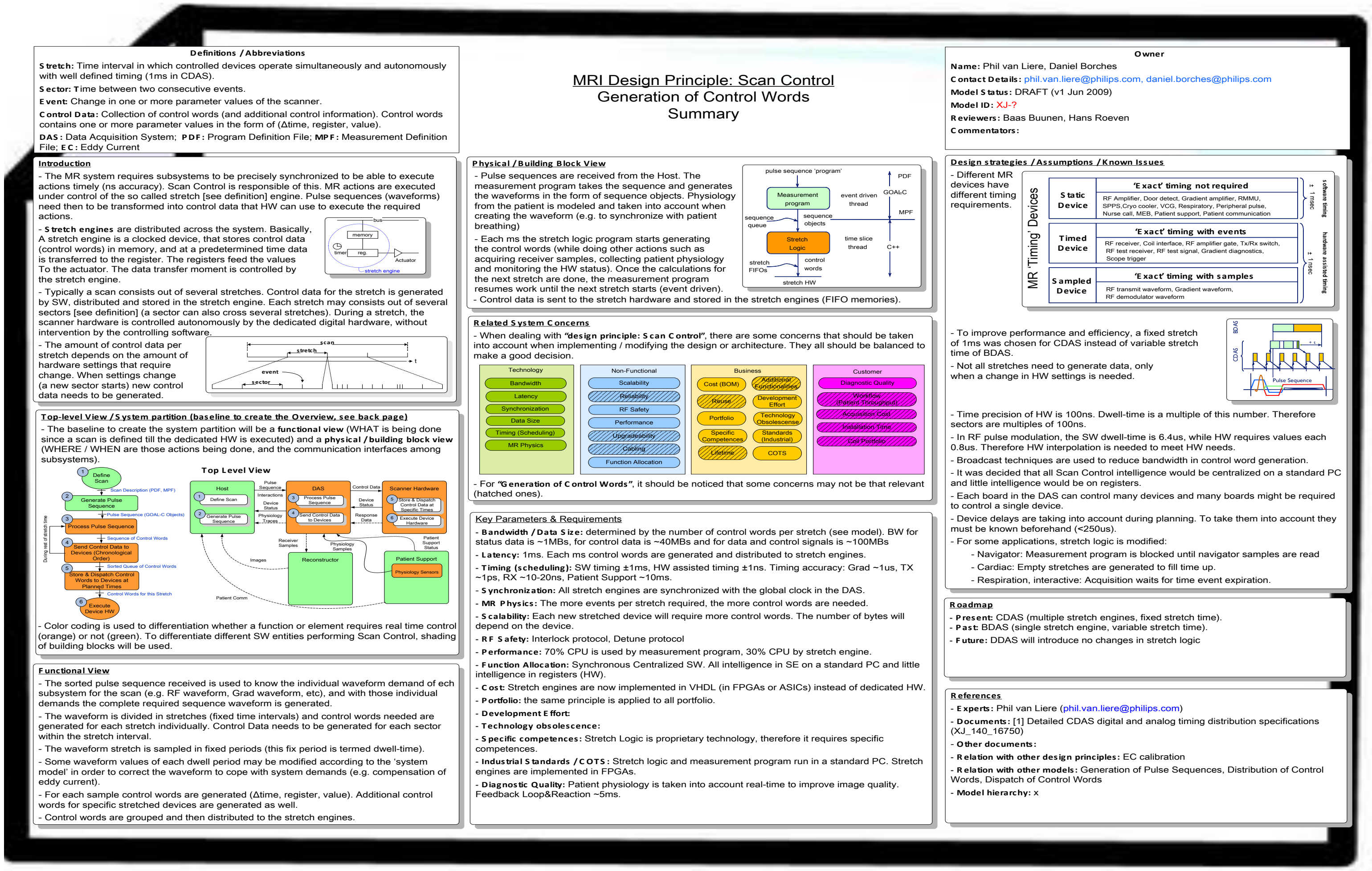
Design Constraint / Decision

1. Due to BW restrictions in the communication network some sampling of the waveform needs to be done in HW
2. As receive channels usually requires the same control words, broadcast can be used to reduce bandwidth demand
3. The frequency range of FM is ~100MHz and the resolution <1Hz, therefore 4 bytes are needed
4. 2 byte resolution is not enough to have a good stability of the Grad waveform, therefore the Dithering algorithm is used to compensate for the error introduced
5. Only change settings between sectors need to generate new control words
6. Control SW has a maximum of 6KB on the amount of data it can transmit to devices during a stretch

1. In practice, it is unlikely that there is no change between stretches, as there is always changes due to corrections such as Dithering and ECC.

1. Patient physiology is taken into account when the waveform is

A3 Text View Example



Definitions / Abbreviations

Stretch: Time interval in which controlled devices operate simultaneously and autonomously with well defined timing (1ms in CDAS).

Sector: Time between two consecutive events.

Event: Change in one or more parameter values of the scanner.

Control Data: Collection of control words (and additional control information). Control words contains one or more parameter values in the form of (Δtime, register, value).

DAS: Data Acquisition System; **PDF:** Program Definition File; **MPF:** Measurement Definition File; **EC:** Eddy Current

MRI Design Principle: Scan Control
Generation of Control Words
Summary

Owner

Name: Phil van Liere, Daniel Borches

Contact Details: phil.van.liere@philips.com, daniel.borches@philips.com

Model Status: DRAFT (v1 Jun 2009)

Model ID: XJ-?

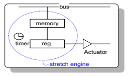
Reviewers: Baas Buunen, Hans Roeven

Commentators:

Introduction

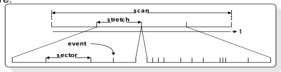
- The MR system requires subsystems to be precisely synchronized to be able to execute actions timely (ns accuracy). Scan Control is responsible of this. MR actions are executed under control of the so called stretch [see definition] engine. Pulse sequences (waveforms) need then to be transformed into control data that HW can use to execute the required actions.

- **Stretch engines** are distributed across the system. Basically, A stretch engine is a clocked device, that stores control data (control words) in memory, and at a predetermined time data is transferred to the register. The registers feed the values To the actuator. The data transfer moment is controlled by the stretch engine.



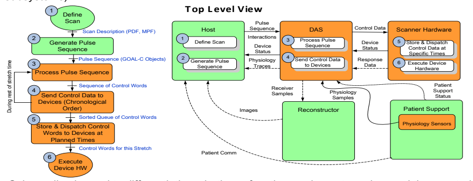
- Typically a scan consists out of several stretches. Control data for the stretch is generated by SW, distributed and stored in the stretch engine. Each stretch may consists out of several sectors [see definition] (a sector can also cross several stretches). During a stretch, the scanner hardware is controlled autonomously by the dedicated digital hardware, without intervention by the controlling software.

- The amount of control data per stretch depends on the amount of hardware settings that require change. When settings change (a new sector starts) new control data needs to be generated.



Top-level View / System partition (baseline to create the Overview, see back page)

- The baseline to create the system partition will be a **functional view** (WHAT is being done since a scan is defined till the dedicated HW is executed) and a **physical / building block view** (WHERE / WHEN are those actions being done, and the communication interfaces among subsystems).



Top Level View
1 Define Scan
Scan Description (PDF, MPF)
2 Generate Pulse Sequence
Pulse Sequence (GOAL-C Objects)
3 Process Pulse Sequence
Sequence of Control Words
4 Send Control Data to Devices (Chronological Order)
Sorted Queue of Control Words
5 Store & Dispatch Control Words to Devices at Planned Times
Control Words for this Stretch
6 Execute Device HW
During rest of stretch time
Host
1 Define Scan
2 Generate Pulse Sequence
Pulse Sequence
Interactions
Device Status
Physiology Traces
DAS
3 Process Pulse Sequence
4 Send Control Data to Devices
Control Data
Device Status
Response Data
Scanner Hardware
5 Store & Dispatch Control Data at Specific Times
6 Execute Device Hardware
Receiver Samples
Physiology Samples
Patient Support Status
Images
Reconstructor
Patient Support
Physiology Sensors
Patient Comm

- Color coding is used to differentiation whether a function or element requires real time control (orange) or not (green). To differentiate different SW entities performing Scan Control, shading of building blocks will be used.

Functional View

- The sorted pulse sequence received is used to know the individual waveform demand of ech subsystem for the scan (e.g. RF waveform, Grad waveform, etc), and with those individual demands the complete required sequence waveform is generated.

- The waveform is divided in stretches (fixed time intervals) and control words needed are generated for each stretch individually. Control Data needs to be generated for each sector within the stretch interval.

- The waveform stretch is sampled in fixed periods (this fix period is termed dwell-time).

- Some waveform values of each dwell period may be modified according to the 'system model' in order to correct the waveform to cope with system demands (e.g. compensation of eddy current).

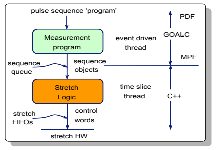
- For each sample control words are generated (Δtime, register, value). Additional control words for specific stretched devices are generated as well.

- Control words are grouped and then distributed to the stretch engines.

Physical / Building Block View

- Pulse sequences are received from the Host. The measurement program takes the sequence and generates the waveforms in the form of sequence objects. Physiology from the patient is modeled and taken into account when creating the waveform (e.g. to synchronize with patient breathing)

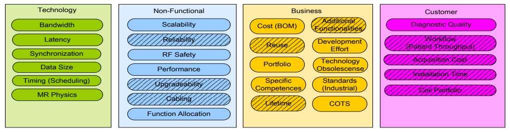
- Each ms the stretch logic program starts generating the control words (while doing other actions such as acquiring receiver samples, collecting patient physiology and monitoring the HW status). Once the calculations for the next stretch are done, the measurement program resumes work until the next stretch starts (event driven).



- Control data is sent to the stretch hardware and stored in the stretch engines (FIFO memories).

Related System Concerns

- When dealing with **"design principle: Scan Control"**, there are some concerns that should be taken into account when implementing / modifying the design or architecture. They all should be balanced to make a good decision.



- For **"Generation of Control Words"**, it should be noticed that some concerns may not be that relevant (hatched ones).

Key Parameters & Requirements

- **Bandwidth / Data Size:** determined by the number of control words per stretch (see model). BW for status data is ~1MBs, for control data is ~40MBs and for data and control signals is ~100MBs
- **Latency:** 1ms. Each ms control words are generated and distributed to stretch engines.
- **Timing (scheduling):** SW timing ±1ms, HW assisted timing ±1ns. Timing accuracy: Grad ~1us, TX ~1ps, RX ~10-20ns, Patient Support ~10ms.
- **Synchronization:** All stretch engines are synchronized with the global clock in the DAS.
- **MR Physics:** The more events per stretch required, the more control words are needed.
- **Scalability:** Each new stretched device will require more control words. The number of bytes will depend on the device.
- **RF Safety:** Interlock protocol, Detune protocol
- **Performance:** 70% CPU is used by measurement program, 30% CPU by stretch engine.
- **Function Allocation:** Synchronous Centralized SW. All intelligence in SE on a standard PC and little intelligence in registers (HW).
- **Cost:** Stretch engines are now implemented in VHDL (in FPGAs or ASICs) instead of dedicated HW.
- **Portfolio:** the same principle is applied to all portfolio.
- **Development Effort:**
- **Technology obsolescence:**
- **Specific competences:** Stretch Logic is proprietary technology, therefore it requires specific competences.
- **Industrial Standards / COTS:** Stretch logic and measurement program run in a standard PC. Stretch engines are implemented in FPGAs.
- **Diagnostic Quality:** Patient physiology is taken into account real-time to improve image quality. Feedback Loop&Reaction ~5ms.

Design strategies / Assumptions / Known Issues

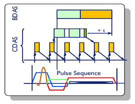
- Different MR devices have different timing requirements.

MR 'Timing' Devices

Static Device	**'Exact' timing not required** RF Amplifier, Door detect, Gradient amplifier, RMMU, SPPS,Cryo cooler, VCG, Respiratory, Peripheral pulse, Nurse call, MEB, Patient support, Patient communication	± 1 msec software timing
Timed Device	**'Exact' timing with events** RF receiver, Coil interface, RF amplifier gate, Tx/Rx switch, RF test receiver, RF test signal, Gradient diagnostics, Scope trigger	± 1 nsec hardware assisted timing
Sampled Device	**'Exact' timing with samples** RF transmit waveform, Gradient waveform, RF demodulator waveform	

- To improve performance and efficiency, a fixed stretch of 1ms was chosen for CDAS instead of variable stretch time of BDAS.

- Not all stretches need to generate data, only when a change in HW settings is needed.



- Time precision of HW is 100ns. Dwell-time is a multiple of this number. Therefore sectors are multiples of 100ns.
- In RF pulse modulation, the SW dwell-time is 6.4us, while HW requires values each 0.8us. Therefore HW interpolation is needed to meet HW needs.
- Broadcast techniques are used to reduce bandwidth in control word generation.
- It was decided that all Scan Control intelligence would be centralized on a standard PC and little intelligence would be on registers.
- Each board in the DAS can control many devices and many boards might be required to control a single device.
- Device delays are taking into account during planning. To take them into account they must be known beforehand (<250us).
- For some applications, stretch logic is modified:
  - Navigator: Measurement program is blocked until navigator samples are read
  - Cardiac: Empty stretches are generated to fill time up.
  - Respiration, interactive: Acquisition waits for time event expiration.

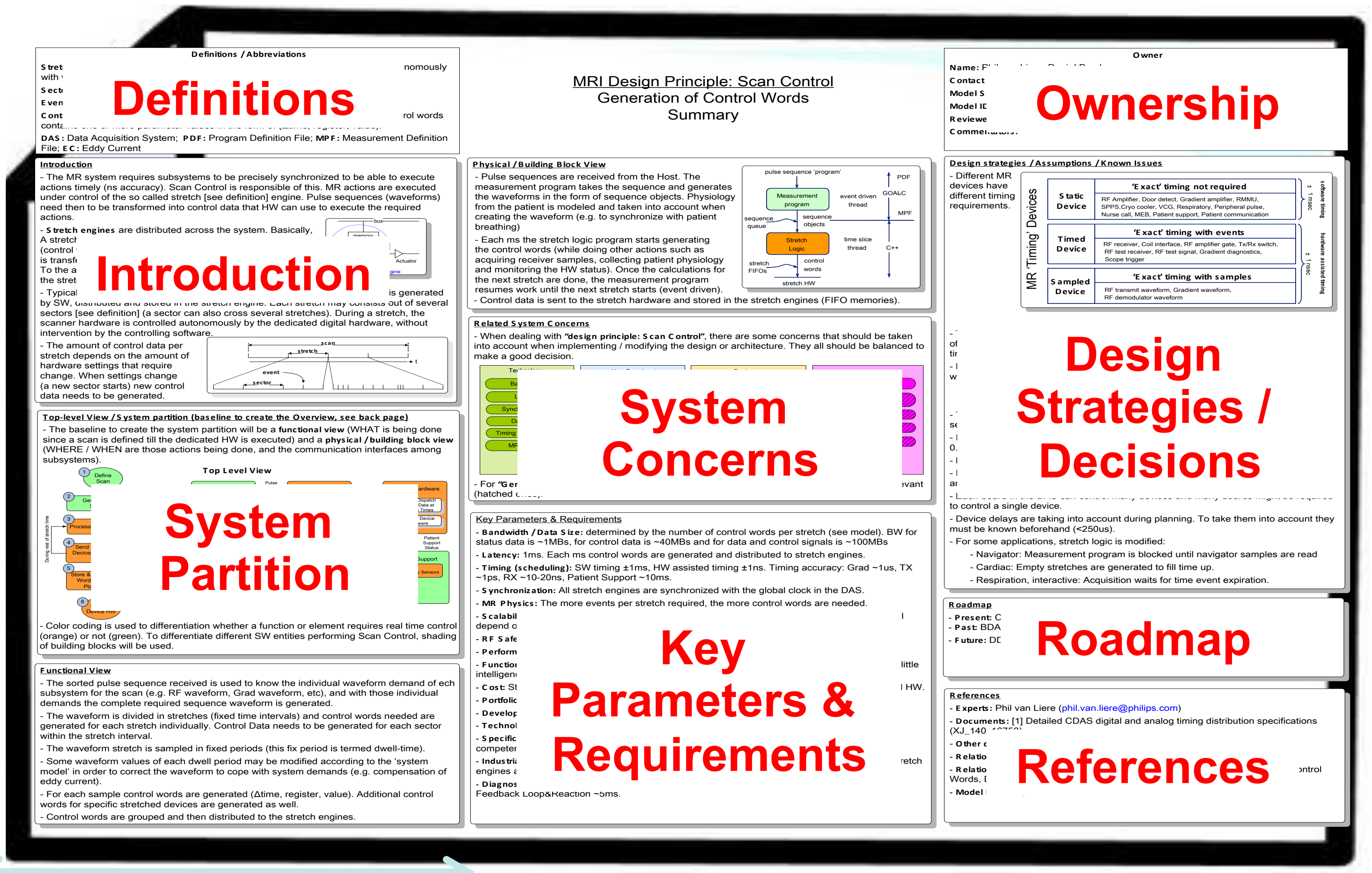
Roadmap

- **Present:** CDAS (multiple stretch engines, fixed stretch time).
- **Past:** BDAS (single stretch engine, variable stretch time).
- **Future:** DDAS will introduce no changes in stretch logic

References

- **Experts:** Phil van Liere (phil.van.liere@philips.com)
- **Documents:** [1] Detailed CDAS digital and analog timing distribution specifications (XJ_140_16750)
- **Other documents:**
- **Relation with other design principles:** EC calibration
- **Relation with other models:** Generation of Pulse Sequences, Distribution of Control Words, Dispatch of Control Words
- **Model hierarchy:** x

A3 Text View Example


Definitions
Introduction
System Partition
System Concerns
Key Parameters & Requirements
Ownership
Design Strategies / Decisions
Roadmap
References

Definitions / Abbreviations

Stret … nomously with …

Secti …

Even …

Cont … rol words conta…

DAS: Data Acquisition System; **PDF:** Program Definition File; **MPF:** Measurement Definition File; **EC:** Eddy Current

MRI Design Principle: Scan Control
Generation of Control Words
Summary

Owner

Name: …

Contact …

Model S …

Model ID …

Reviewe …

Commen …

Introduction

- The MR system requires subsystems to be precisely synchronized to be able to execute actions timely (ns accuracy). Scan Control is responsible of this. MR actions are executed under control of the so called stretch [see definition] engine. Pulse sequences (waveforms) need then to be transformed into control data that HW can use to execute the required actions.

- **Stretch engines** are distributed across the system. Basically, A stretch … (control … is transfe… To the a… the stret…

- Typical … is generated by SW, distributed and stored in the stretch engine. Each stretch may consists out of several sectors [see definition] (a sector can also cross several stretches). During a stretch, the scanner hardware is controlled autonomously by the dedicated digital hardware, without intervention by the controlling software.

- The amount of control data per stretch depends on the amount of hardware settings that require change. When settings change (a new sector starts) new control data needs to be generated.

Top-level View / System partition (baseline to create the Overview, see back page)

- The baseline to create the system partition will be a **functional view** (WHAT is being done since a scan is defined till the dedicated HW is executed) and a **physical / building block view** (WHERE / WHEN are those actions being done, and the communication interfaces among subsystems).

- Color coding is used to differentiation whether a function or element requires real time control (orange) or not (green). To differentiate different SW entities performing Scan Control, shading of building blocks will be used.

Functional View

- The sorted pulse sequence received is used to know the individual waveform demand of ech subsystem for the scan (e.g. RF waveform, Grad waveform, etc), and with those individual demands the complete required sequence waveform is generated.

- The waveform is divided in stretches (fixed time intervals) and control words needed are generated for each stretch individually. Control Data needs to be generated for each sector within the stretch interval.

- The waveform stretch is sampled in fixed periods (this fix period is termed dwell-time).

- Some waveform values of each dwell period may be modified according to the 'system model' in order to correct the waveform to cope with system demands (e.g. compensation of eddy current).

- For each sample control words are generated (Δtime, register, value). Additional control words for specific stretched devices are generated as well.

- Control words are grouped and then distributed to the stretch engines.

Physical / Building Block View

- Pulse sequences are received from the Host. The measurement program takes the sequence and generates the waveforms in the form of sequence objects. Physiology from the patient is modeled and taken into account when creating the waveform (e.g. to synchronize with patient breathing)

- Each ms the stretch logic program starts generating the control words (while doing other actions such as acquiring receiver samples, collecting patient physiology and monitoring the HW status). Once the calculations for the next stretch are done, the measurement program resumes work until the next stretch starts (event driven).

- Control data is sent to the stretch hardware and stored in the stretch engines (FIFO memories).

Related System Concerns

- When dealing with **"design principle: Scan Control"**, there are some concerns that should be taken into account when implementing / modifying the design or architecture. They all should be balanced to make a good decision.

- For **"Ger** … vant (hatched …).

Key Parameters & Requirements

- **Bandwidth / Data Size:** determined by the number of control words per stretch (see model). BW for status data is ~1MBs, for control data is ~40MBs and for data and control signals is ~100MBs

- **Latency:** 1ms. Each ms control words are generated and distributed to stretch engines.

- **Timing (scheduling):** SW timing ±1ms, HW assisted timing ±1ns. Timing accuracy: Grad ~1us, TX ~1ps, RX ~10-20ns, Patient Support ~10ms.

- **Synchronization:** All stretch engines are synchronized with the global clock in the DAS.

- **MR Physics:** The more events per stretch required, the more control words are needed.

- **Scalabil** … depend o…

- **RF Safe** …

- **Perform** …

- **Functio** … little intelligen…

- **Cost:** St… HW.

- **Portfolic** …

- **Develop** …

- **Technol** …

- **Specific** … competer…

- **Industri** … retch engines …

- **Diagnos** … Feedback Loop&Reaction ~5ms.

Design strategies / Assumptions / Known Issues

- Different MR devices have different timing requirements.

MR 'Timing' Devices			
Static Device	'Exact' timing not required	RF Amplifier, Door detect, Gradient amplifier, RMMU, SPPS,Cryo cooler, VCG, Respiratory, Peripheral pulse, Nurse call, MEB, Patient support, Patient communication	± 1 msec, software timing
Timed Device	'Exact' timing with events	RF receiver, Coil interface, RF amplifier gate, Tx/Rx switch, RF test receiver, RF test signal, Gradient diagnostics, Scope trigger	± 1 nsec, hardware assisted timing
Sampled Device	'Exact' timing with samples	RF transmit waveform, Gradient waveform, RF demodulator waveform	± 1 nsec, hardware assisted timing

- … of … tir…

- … w…

- … se…

- … 0.…

- …

- … ar…

- Each board in the DAS can control many devices and many boards might be required to control a single device.

- Device delays are taking into account during planning. To take them into account they must be known beforehand (<250us).

- For some applications, stretch logic is modified:
  - Navigator: Measurement program is blocked until navigator samples are read
  - Cardiac: Empty stretches are generated to fill time up.
  - Respiration, interactive: Acquisition waits for time event expiration.

Roadmap

- **Present:** C…
- **Past:** BDA…
- **Future:** DD…

References

- **Experts:** Phil van Liere (phil.van.liere@philips.com)

- **Documents:** [1] Detailed CDAS digital and analog timing distribution specifications (XJ_140…)

- **Other c** …

- **Relatio** …

- **Relatio** … ontrol Words, D…

- **Model** …

Variable Box Size

Outline

- Problem Background: Evolution of Complex Systems
- Study Case: Philips MRI System
- System Evolution Barriers (Philips' Survey)
- A3 Architecture Overview Example
- Reverse Architecting: Collect, Abstract, Present
- A3 Architecture Overviews
- Application: Philips System Design Specification (SDS) New Style
- Lessons Learned

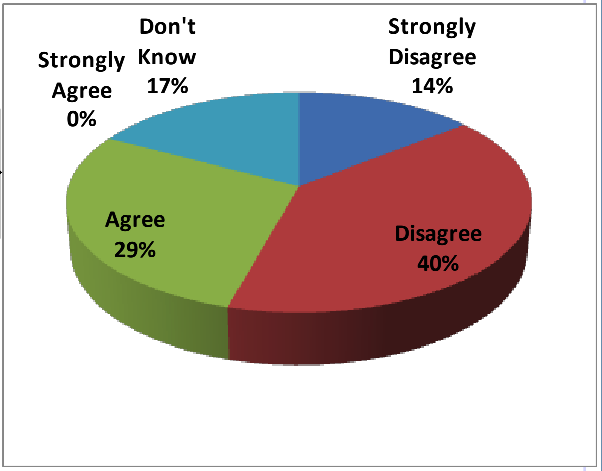
Application: Philips SDS New Style

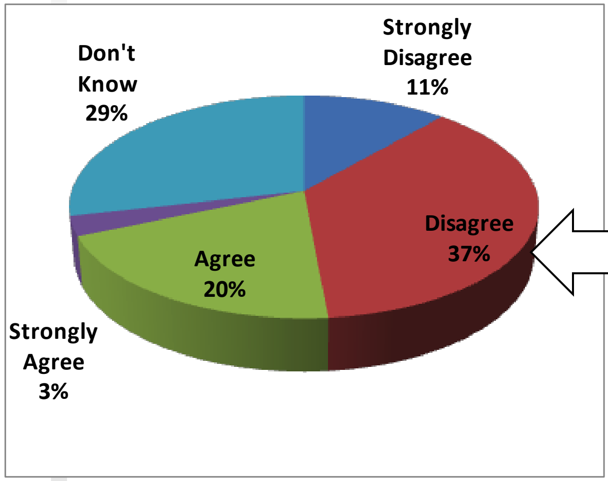
- Systems Design Specification (SDS) is usually the main description of a systems design.
- Meant to:
  - Specify how requirements are met
  - Consolidate partitioning into design entities
  - Define interfaces
  - etc
- Used by companies to:
  - Consolidate designs
  - Support ‘development memory’
  - Educational purposes

Application: Philips SDS New Style

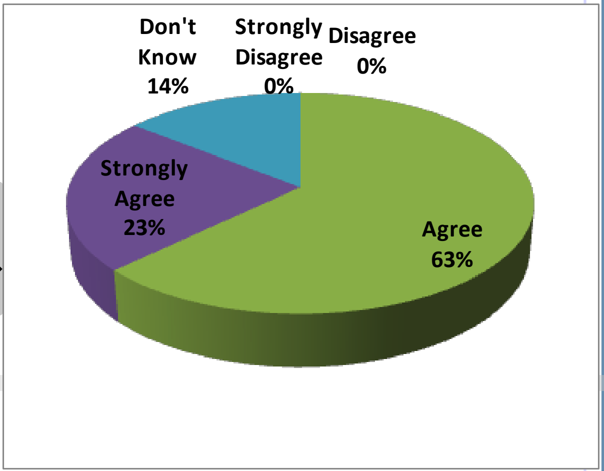
- Motivation

SDS is useful to your work





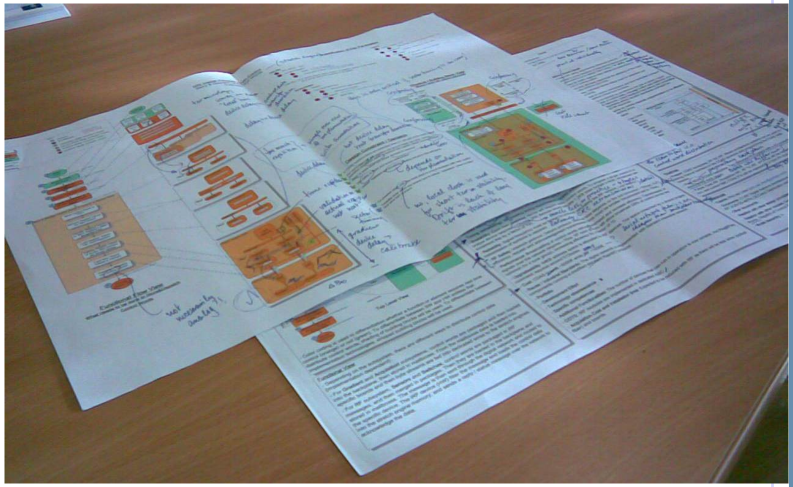
C SDS provides the necessary knowledge to understand the system outside my domain of expertise



The use of the SDS should be encouraged within the organization

Application: Philips SDS New Style

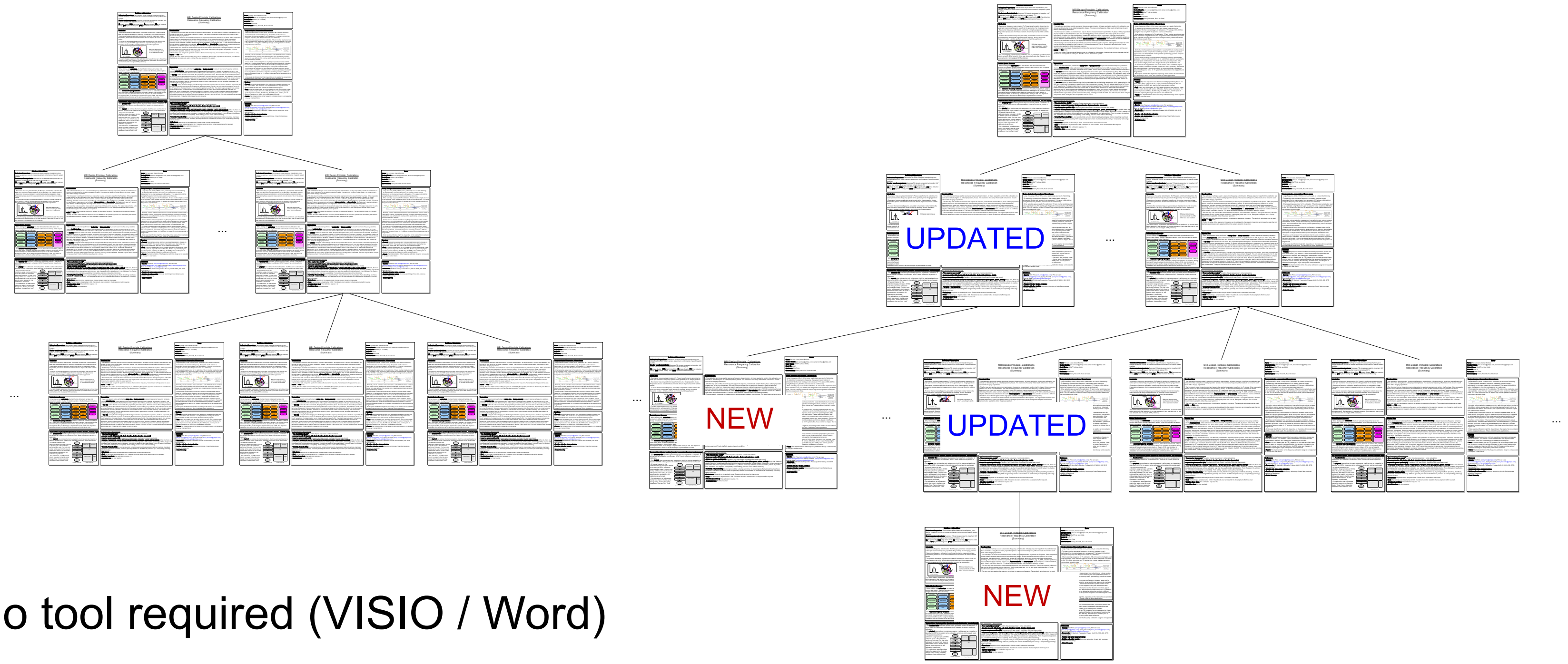
- SDS new style proof of concept was developed
  - ~10 A3 Architecture Overviews were created
    - Scan Control (4), Calibrations (2), Cooling(1), etc
- Used during running projects
  - Discussion tool at meetings
  - Learning tool for new employees
  - Consolidate design concepts



After its evaluation, Project Leaders were encouraged to create their own A3 Architecture Overviews for their projects!

Benefits

•The system description is easier to update & extend



•No tool required (VISIO / Word)

•No domain specific language (low learning curve)

•Compactness & brevity

Outline

- Problem Background: Evolution of Complex Systems
- Study Case: Philips MRI System
- System Evolution Barriers (Philips' Survey)
- A3 Architecture Overview Example
- Reverse Architecting: Collect, Abstract, Present
- A3 Architecture Overviews
- Application: Philips System Design Specification (SDS) New Style
- Lessons Learned

Lessons Learned

+

•Great discussion tool.

•Preferred to text documents.

•Different disciplines and departments can use it without much explanation.

•Enable to capture insight during discussions (

•No “I do not have time to read it” excuse.

–

•A4 addiction

•Printer problems?

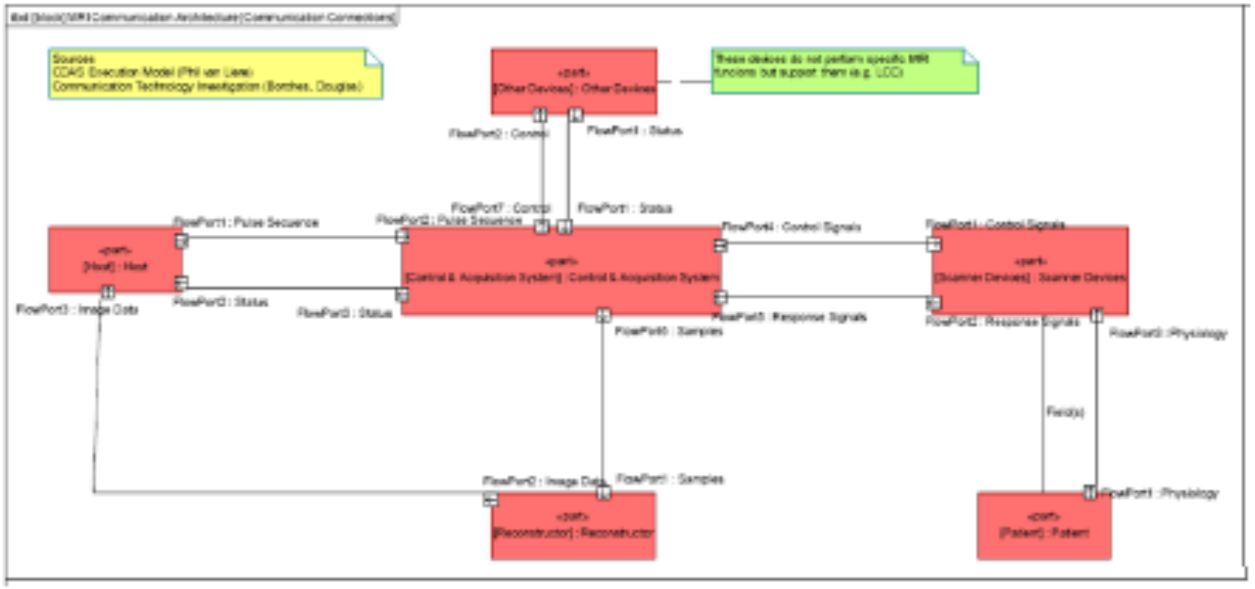
•Screen size?

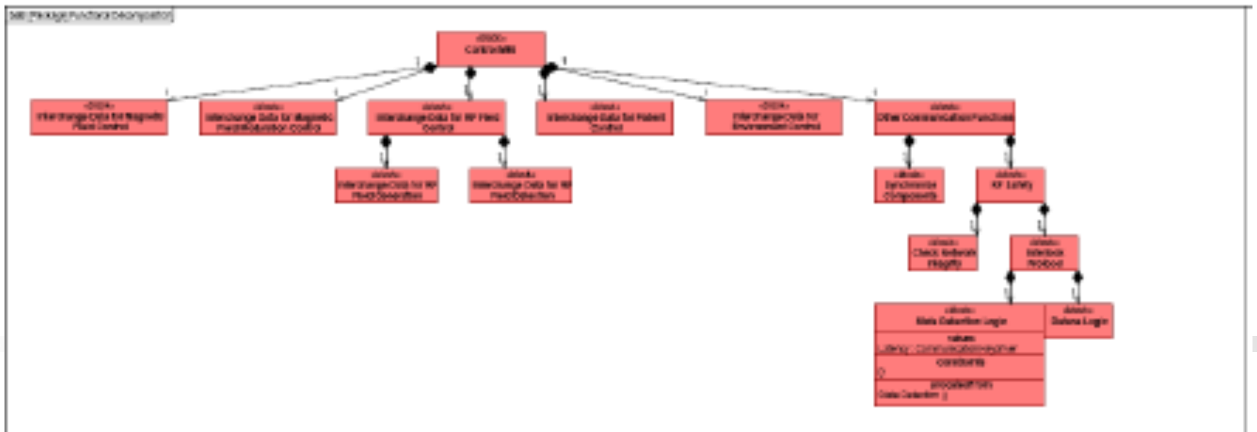
Questions?

Thanks for listening

SysML

- MRI Architecture Modeled in SysML


Sources
CDAS Execution Model (Phil van Liere)
Communication Technology Investigation (Borches, Douglas)
«part» [Other Devices] : Other Devices
These devices do not perform specific MRI functions but support them (e.g. LCD)
FlowPort2 : Control
FlowPort1 : Status
FlowPort7 : Control
FlowPort : Status
FlowPort1 : Pulse Sequence
FlowPort2 : Pulse Sequence
«part» [Host] : Host
«part» [Control & Acquisition System] : Control & Acquisition System
FlowPort4 : Control Signals
FlowPort1 : Control Signals
«part» [Scanner Devices] : Scanner Devices
FlowPort3 : Image Data
FlowPort2 : Status
FlowPort3 : Status
FlowPort5 : Response Signals
FlowPort6 : Samples
FlowPort2 : Response Signals
FlowPort3 : Physiology
Field(s)
FlowPort2 : Image Data
FlowPort1 : Samples
FlowPort1 : Physiology
«part» [Reconstructor] : Reconstructor
«part» [Patient] : Patient

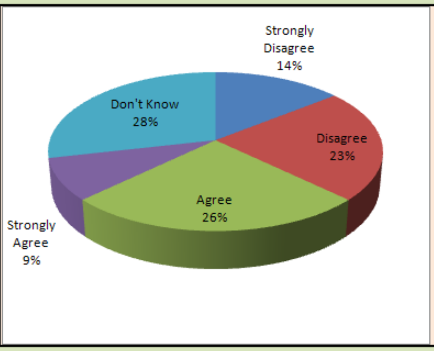


SysML

A text-based document is preferred to a model-based description for the SDS

General

Strongly Disagree	14%
Disagree	23%
Agree	26%
Strongly Agree	9%
Don't Know	29%

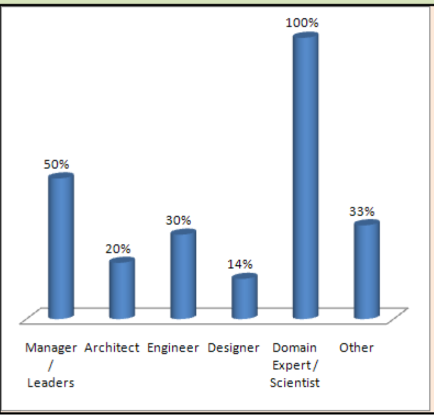


Findings

There is some tension regarding whether to stick with a text-based document or not. About half people want to stick with text-based documetns. However among the half that prefer a model-based description, there is a large group of people who really want to move to this approach.

Per working title (Strongly Agree/Agree)

Manager / Leaders	50%
Architect	20%
Engineer	30%
Designer	14%
Domain Expert / Scientist	100%
Other	33%



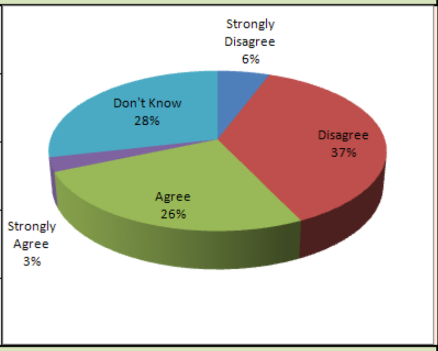
Findings

We see that half of the Managers and all Domain experts want to stick to text-base documents, while Designers and Architects want to move to a model-base document. Also most Engineers and Other employees are in favor of model-base description.

Formal semantics (e.g. UML) should be used in the SDS

General

Strongly Disagree	6%
Disagree	37%
Agree	26%
Strongly Agree	3%
Don't Know	29%

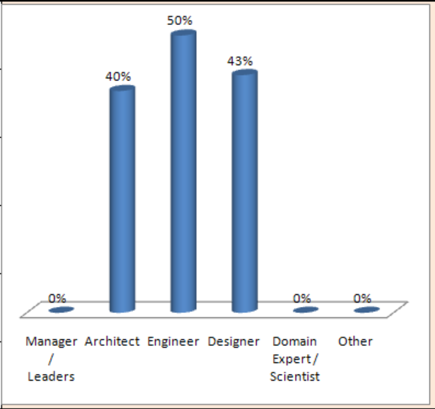


Findings

Although some people thinks formal semantics should be used, the majority thinks it should not be used.

Per working title (Strongly Agree/Agree)

Manager / Leaders	0%
Architect	40%
Engineer	50%
Designer	43%
Domain Expert / Scientist	0%
Other	0%



Findings

Managers, Domain experts and Other employees do not want at all formal semantics.

Managers want text-based descriptions!

SW experts want standard modeling languages

SysML

- **Experiences**
  - Practitioners do not want to spend time learning SysML
  - Some disciplines not comfortable with it (architecture level)
  - Time at meetings spent discussing about notation not contents
  - Tool dependency
  - Models do not resemble MRI
  - SysML requires too many views to create an A3 Architecture Overivew