Safety-Critical Example

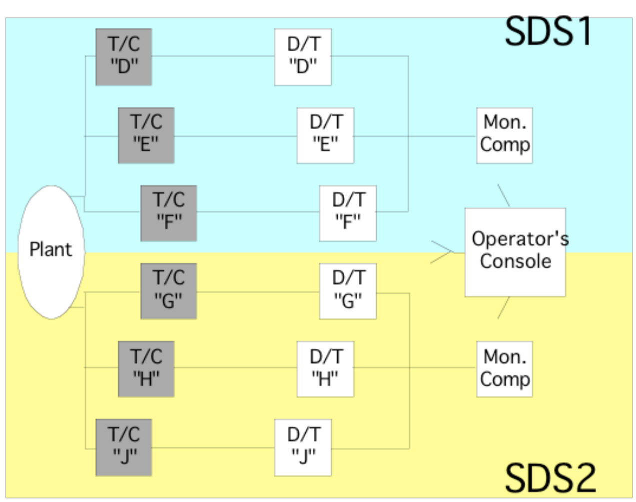
- SDS1 & SDS2: Real-Time Monitoring & Shutdown at a Nuclear Power Plant
  - These computer systems have hard deadlines in which they have to detect potential accident scenarios.
  - They also have hard deadlines in which they have to initiate alarms, and, if necessary, initiate shutdown of the reactor.

207

207

Shutdown System Context

Darlington Shutdown Systems



T/C = Trip Computer

D/T = Display/Test Computer

Mon. Comp = Monitor Computer

208

208

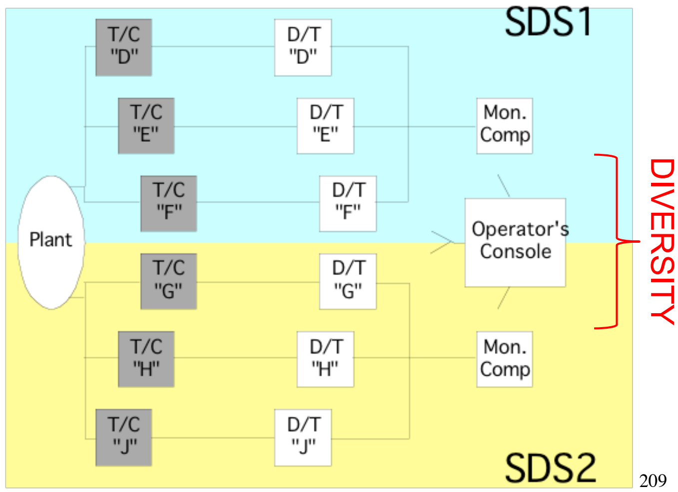
Shutdown System Context

Darlington Shutdown Systems

T/C = Trip Computer

D/T = Display/Test Computer

Mon. Comp = Monitor Computer



209

209

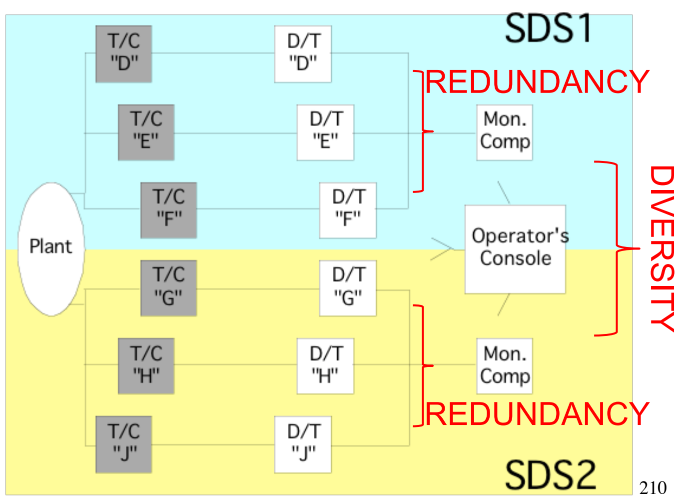
Shutdown System Context

Darlington Shutdown Systems

T/C = Trip Computer

D/T = Display/Test Computer

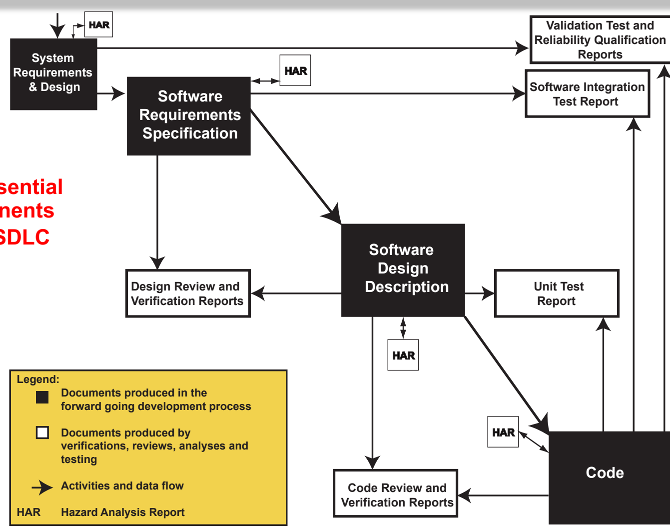
Mon. Comp = Monitor Computer



210

210

Safety-Critical Software



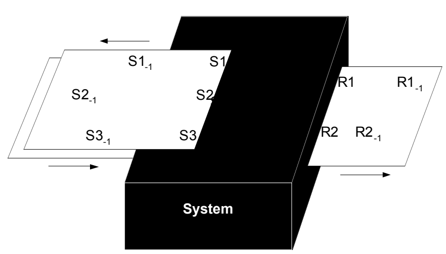
The essential components of the SDLC

211

Requirements - 1

- **T**rip **C**omputer **D**esign **R**equirements (Mills' Black-Box)

Mills, H.D.: Stepwise refinement and verification in box-structured systems. Computer 21 (1988) 23-36



$R = f (S, S_h)$

S is the set of stimuli, S_h the set of stimulus history, R the set of responses

212

System Requirements

- TCDR
  - Context diagrams
  - Stimuli & Responses
  - Constants
  - Main function tables - with rationale
  - Natural language expressions
  - Tolerances, PTRs and FTRs
  - Anticipated changes
  - Changes from previous freezes - rationale

213

213

Black-Box Requirements

- We expect that all responses are described in terms of stimuli and stimuli history only.
- It is sometimes advantageous to allow response history to appear in functional descriptions.
- In deterministic systems, response history is always a representation of stimuli history.

214

214

Notation - 1

- We refer to stimuli as monitored variables, and responses as controlled variables.
- We prefix identifiers by a suitable character followed by _ to help identify the role of the identifier.
- m_ name is a monitored variable, c_name is a controlled variable, k_name is a constant, etc.

Notation - 2

- m_name represents value of the current instance of m_name.
- m_name_{-1} represents value of the previous instance of m_name.
- If m_name is time-continuous, there's an arbitrarily small time, δt, between m_name and m_name_{-1}.
- t_{now} represents current time.

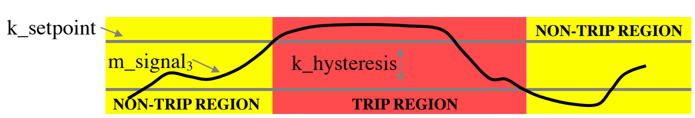
Notation - 3

- If m_name is time-discrete, time between m_name and m_name_{-1} is t(m_name) - t(m_name_{-1}). In general, t(var) returns time stamp of the instance of var.
- In a real system it will not be possible to represent $R = f(S,S_h)$ by a single function or function table
  - So we decompose the function f into a network of sub-functions

217

217

Example Function



$f\_sensortrip_i$, i=1,..,4

{For each i = 1,..,4}

Condition	*Result* **$f\_sensortrip_i$**
$k\_setpoint \le m\_signal_i$ *{i^{th} signal is now in the trip region}*	e_Trip
$k\_setpoint - k\_hysteresis < m\_signal_i < k\_setpoint$ *{i^{th} signal is now in the deadband region}*	No Change
$m\_signal_i \le k\_setpoint - k\_hysteresis$ *{i^{th} signal is now in the non-trip region}*	e_NotTrip

218

218

Anticipated Changes - 1

- Information Hiding is a software design paradigm that was introduced by Parnas in a famous paper in the early 1970s.
- The original version of Information Hiding used anticipated design changes to drive the software decomposition.
- It turns out that requirement changes are an even greater source of "secrets".

Parnas, D.: On the criteria to be used in decomposing systems into modules.
Communications of the ACM December (1972) 1053-1058

Anticipated Changes - 2

Table 9.1-1 - Anticipated Changes

Id	Anticipated Change
AC-1	Provisions shall be made in the software coding to give all power dependent setpoints the ability to handle arbitrary setpoint functions (instead of the current step functions). As a minimum requirement, facilities to accommodate a setpoint value for each 1% power change shall be provided up to an upper limit of 110% Full Power (FP).
AC-2	Ranges as specified in the table that defines values of constants, see [27], shall be pre-verified so that the application can use any trip setpoint in the relevant range.
AC-3	New parameter trips may be added.
AC-4	The algorithm and number of detectors used for the estimated power calculation may change from the current specification.
AC-5	Individual deadbands may be revised, HTLF in particular.
AC-6	The processing time for the HTLF analog inputs may be reduced by 100 ms to make provisions to reallocate delay external to the trip computer, see [47], 2.2.2.

System Design
(includes software requirements)

- **T**rip **C**omputer **D**esign **D**escription
  - Uses TCDR as a basis
  - Adds design information, e.g. pushbutton debouncing
  - Model changes to a Finite State Machine with an arbitrarily small clock-tick
  - SRS contained within TCDD

221

221

Finite State Machine Model

- $C(t)$ - set of controlled vars at time t
- $M(t)$ - set of monitored vars at time t
- $S(t)$ - set of state vars at time t
- t_0 - time of initialisation
- $S(t_0)$ must be known

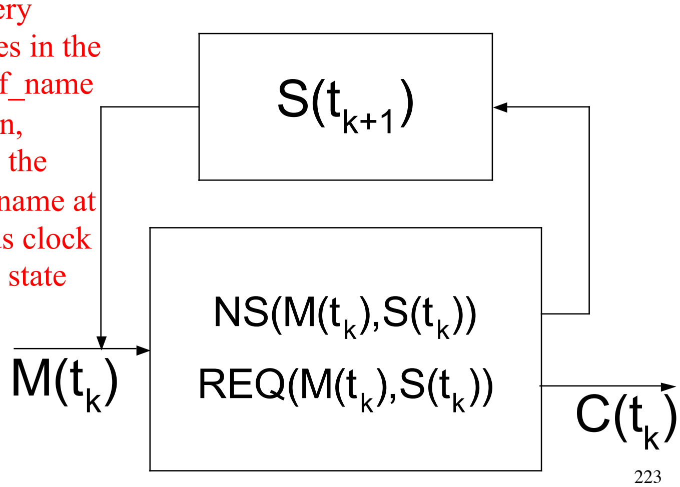
$C(t_k) = REQ(M(t_k), S(t_k))$
$S(t_{k+1}) = NS(M(t_k), S(t_k)),\ k=0,1,2,3,...$

222

222

FSM for Requirements

We have very simple states in the TCDD. If f_name is a function, f_name$_{-1}$ is the value of f_name at the previous clock tick. It is a state variable.



223

223

TCDD Documentation

- Context diagrams
- Monitored & Controlled Variables
- Constants
- Main function tables
- Natural language expressions
- M-I mappings, transfer events
- C-O mappings, transfer events
- Tolerances, PTRs and FTRs
- Anticipated changes
- Changes from previous freezes

224

224

TCDD Documentation

- Context diagrams
- Monitored & Controlled Variables
- Constants
- Main function tables
- Natural language expressions
- M-I mappings, transfer events
- C-O mappings, transfer events
- Tolerances, PTRs and FTRs
- Anticipated changes
- Changes from previous freezes

We'll find out what these are later

225

225

Design Details in TCDD

- We can see how debouncing pushbuttons affects the behaviour specified in the TCDR.
- In particular, "NOP Abnormal 1 setpoint is requested or cancelled" is specified in the TCDR without debouncing and then re-specified in the TCDD with debouncing.

226

226

Pushbuttons in TCDR

A "natural-language expression"

Condition	Result: NOP Abnormal 1 setpoint is requested or cancelled
(m_NOPspAbn1ON = e_NotPressed) & (m_NOPspAbn1OFF = e_NotPressed)	No Change
(m_NOPspAbn1ON = e_NotPressed) & (m_NOPspAbn1OFF = e_Pressed)	cancelled
(m_NOPspAbn1ON = e_Pressed) & (m_NOPspAbn1OFF = e_NotPressed)	requested
(m_NOPspAbn1ON = e_Pressed) & (m_NOPspAbn1OFF = e_Pressed)	requested

227

227

Debounce Pushbuttons - 1

- Pushbuttons in the TCDD

Condition	Result: f_NOPspAbn1OFF	Result: f_StuckNOPspAbn1OFF
m_NOPspAbn1OFF = e_NotPressed	e_pbNotDebounced	False
[m_NOPspAbn1OFF = e_Pressed] & NOT [(m_NOPspAbn1OFF = e_Pressed) Held for k_Debounce]	e_pbNotDebounced	False
[(m_NOPspAbn1OFF = e_Pressed) Held for k_Debounce] & NOT [(m_NOPspAbn1OFF = e_Pressed) Held for k_pbStuck]	e_pbDebounced	False
(m_NOPspAbn1OFF = e_Pressed) Held for k_pbStuck	e_pbStuck	True

228

228

Debounce Pushbuttons - 2

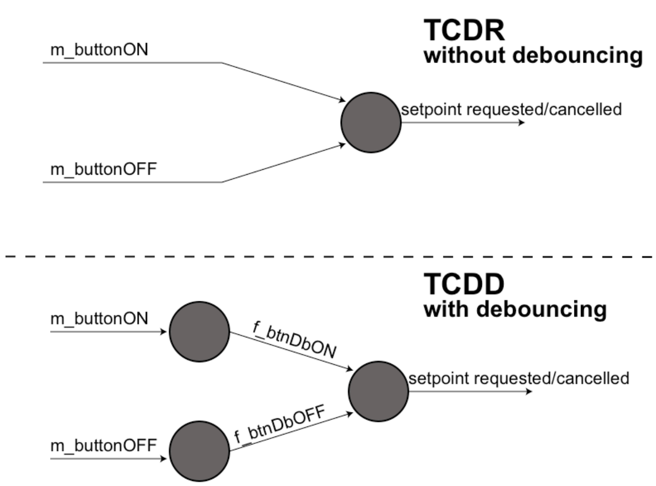
- So, NOP Abnormal 1 setpoint is requested or cancelled is now defined in terms of f_NOPspAbn1ON/OFF

Condition	*Result* **NOP Abnormal 1 setpoint is requested or cancelled**
f_NOPspAbn1ON = e_pbStuck OR f_NOPspAbn1OFF = e_pbStuck	requested
f_NOPspAbn1ON = e_pbNotDebounced & f_NOPspAbn1OFF = e_pbNotDebounced	No Change
f_NOPspAbn1ON = e_pbNotDebounced & f_NOPspAbn1OFF = e_pbDebounced	cancelled
f_NOPspAbn1ON = e_pbDebounced & f_NOPspAbn1OFF = e_pbNotDebounced	requested
f_NOPspAbn1ON = e_pbDebounced & f_NOPspAbn1OFF = e_pbDebounced	requested

229

229

Debounce Pushbuttons - 3



230

230

Input and Output Variables

- The TCDD specifies “transfer events” (what the software must do to trigger getting a software input, I, or emitting a software output, O).
- The TCDD also specifies the “M-I mappings”, and “C-O mappings” (what the hardware does).

231

231

Performance Timing Reqs & Anticipated Changes

- The TCDD specifies a modified list of performance timing requirements that takes into account the design aspects added in the TCDD.
- The TCDD also lists a (potentially modified) list of anticipated changes.

232

232

Performance Timing Reqs

Table 2.2.2 - Timing Requirements

Controlled Variable	Governing Variables	PTR	TR	Reference
c_NOPparmtrip	m_NOPai$_i$, i=1,..,18	160 ms	Default (Already more restrictive than required to meet seal-in)	TCDR
	m_NOPspAbn1OFF m_NOPspAbn1ON m_NOPspAbn2OFF m_NOPspAbn2ON m_NOPspLPOFF m_NOPspLPON	850 ms (Held for k_Debounce) / 500 ms	350 ms	TCDR and [13], #1 and [23]
	m_CalibrateEnable	N/A	1000 ms	TCDR
	M_RxFnType	2000 ms	Default	TCDR
	M_RxNOPGain$_i$, i=1,..,18	N/A	2000 ms	TCDR

Example for c_NOPparmtrip

233

233

Example Function in TCDD

2.3.1 Steam Generator Low Level Sensor Trip

Determines whether there is a Steam Generator low level sensor trip, which is used to determine whether there is an associated parameter trip.

2.3.1.1 Inputs/Natural Language Expressions

Input	NL Expression	Reference
f_SGLLsp	-	2.3.3.4
m_SGLevel$_i$, i=1,..,4	-	Table 2.6.1-1

2.3.1.2 Initial Value

Name	Initial Value	Reference
(f_SGLLsentrip$_i$)$_{-1}$, i=1,..,4	e_Trip	TCDR

2.3.1.3 Output Units/Type

Output	Type
f_SGLLsentrip$_i$, i=1,..,4	y_trip

2.3.1.4 f_SGLLsentrip$_i$, i=1,..,4

{For each i = 1,..,4}

Condition	*Result* **f_SGLLsentrip$_i$**
f_SGLLsp ≥ m_SGLevel$_i$ *{SG level$_i$ signal is now in the trip region}*	e_Trip
f_SGLLsp + k_SGLLhys > m_SGLevel$_i$ > f_SGLLsp *{SG level$_i$ signal is now in the deadband region}*	(f_SGLLsentrip$_i$)$_{-1}$
m_SGLevel$_i$ ≥ f_SGLLsp + k_SGLLhys *{SG level$_i$ signal is now in the non-trip region}*	e_NotTrip

234

234