



Roles and Importance of Use Cases in Systems Engineering

William K. Day

Consultant

Regulatory Compliance Associates



Hard-to-fix engineering issues

- Does your program have any of these issues?
 - Recurring quality problems
 - Clumsy and easily misused user interfaces
 - Poor exception handling, fault tolerance
 - Ineffective system validation
 - Difficult to manufacture, service, and support
- Why?



Why?

- Poor understanding of the problem domain
- Incomplete translation into the solution

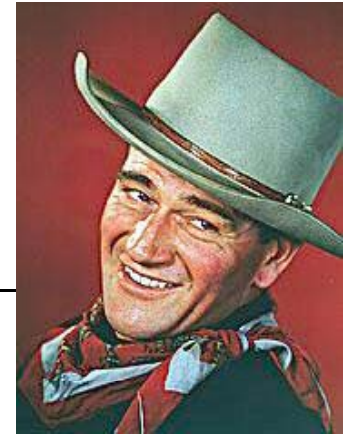


What to do?

- Document all system actors
- Document each actor's needs
- Develop a use case map
- Build the use cases
 - Start with the actor steps
 - Add the system responses
- Iterate...a lot

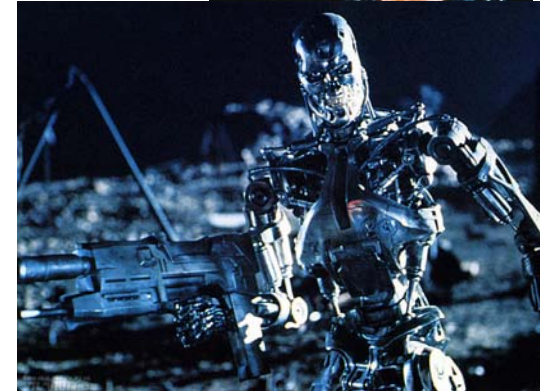
Actor

- Anyone or anything that interacts with the system
 - Who or what uses the system?
- Not part of the system, they represent roles that a user of the system can play
 - One user could be represented by different roles (different actors)
 - One role (actor) could represent different users



An actor may...

- ...*initiate* system actions
- ...*interchange* information with the system
- ...passively *receive* information from the system
- ...represent a *human, machine, or another system*





Actor

- Make sure you have a complete list of actors including environments and sub-classes
 - Patient
 - At home
 - Travelling
 - Blind
 - Deaf



Actor

- Typically missing actors
 - Service technician (FRU diagnostics)
 - Mfg technician (calibration, final test diagnostics)
 - V&V engineer (injecting failure modes)
 - Sales (demonstration modes)
 - Clinical software (set prescription, get results)



Actor needs

- Should be documented in that actor's terms
 - The patient needs the system to <solve a problem>.
 - The service technician needs the system to identify which FRU(s) are non-functional, as far as feasible.
- Should define the problem, not the solution
- Top level documents that typically come into Engineering are already too specific

Two domains

Problem Domain

Actor Needs

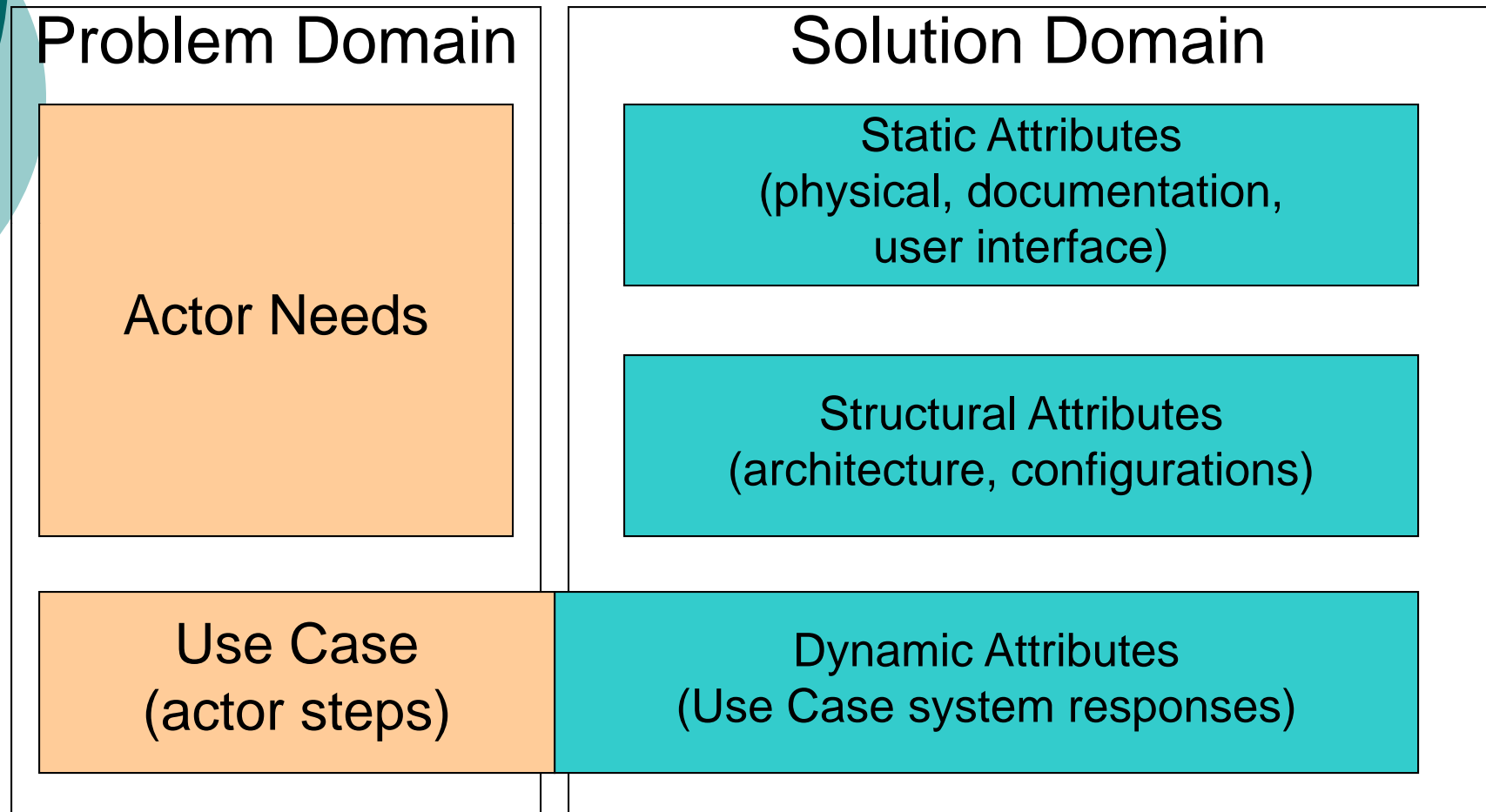
Solution Domain

Static Attributes
(physical, documentation,
user interface)

Structural Attributes
(architecture, configurations)

Dynamic Attributes
(functional requirements)

Use cases bridge the domain gap





What's a use case?

- A sequence of actions between external entities (actors) and the system
- Captures an *externally visible* function from start to finish
- Yields an *observable* and *testable* result that is of value to a particular actor
- Examples
 - Load weapon
 - Fire weapon
 - Ship weapon
 - Test weapon accuracy



Always have a map

- Build out the full list of use cases and sub-use cases as a use case map to facilitate:
 - Completeness
 - Consistency
 - Reuse (uses, extends)
 - Prioritization
- Publish and review the map often
 - Filling in parts of the map exposes new ideas for other parts of the map



Use Case map

Patient

- Receive patient specific product usage training
- Adjust localization settings
 - Adjust date
 - Adjust time
 - Adjust timezone
 - Adjust automatic daylight savings time setting
- Adjust device settings
 - Adjust audible volume profile (awake)
 - Adjust audible volume profile (asleep)
 - Adjust display profile (day)
 - Adjust display profile (night)
- Program therapy
 - Adjust therapy type
 - Adjust therapy time
 - Adjust therapy total volume
 - Adjust therapy fill volume
 - Adjust therapy last fill volume
 - Adjust therapy last fill solution type
- Perform pre-therapy steps
- Perform therapy
 - Temporary disconnect
 - Reconnect after temporary disconnect
 - Abort therapy
- Perform post-therapy steps
- Review therapy log
- Review alarm log
- Power on system
- Power off system
- Call Service
- Send system in for service
- Report complaint



Why use cases?

- Derived directly from the actor's needs
 - Easier to show complete flow down to solution
- Speaks the actor's language
 - Easier for non-technical reviewers
- Reusable and extendable
- Actors can be simulated



Why use cases?

- Provides consistent referential context at all times
- Inherently dynamic
 - Dynamic functionality very difficult to describe with standalone requirements
 - Resolves an actor's problem
 - May have alternatives or exceptions
 - May have sequences, states, modes, loops



Use cases enhance usability

- Provide documented actor steps
 - Human factors engineering
 - Step reduction (i.e. automation)
 - Performance considerations in context
- Identify opportunities for...
 - Site configurability
 - User customization
 - Localization



Use cases enhance hazard analysis

- Each actor step can be analyzed for misuse opportunities
 - Fine grained probabilities
- Each system response can be analyzed for failure modes
 - Impact of reliability estimates
 - May need redundancy
- Entire step/response sequences can be analyzed as a unit
 - Fault trees



Use cases enhance secondary users

- Manufacturing enhanced
 - Early identification of assembly tests, diagnostics, automated test equipment, calibrations, error coding, setup, software installation, final tests, factory installed options, etc.
- Service enhanced
 - Early identification of software upgrade & downgrade functions, field replaceable units, field installed options, remote diagnosis, service manuals, etc.



Use cases enhance documentation

- Actor specific tasks allow for targeted user manuals and help systems
 - Actor steps needed even more than system responses
- Actor specific tasks allow for more directed technical support



Use cases enhance testability

- Use cases written much like test cases
- Actor-oriented testing
 - Improves test design and organization
 - Reduces redundant testing
 - Improves environment simulation
- Test development can start much earlier in the process
- Exception conditions documented in context



Use cases enhance testability

- Alternative paths documented in context
- Expected results documented in context
- Exposes more externally observable system behavior to the tester
 - Can also be used by test writers to identify and request testability functions and instrumentation to internal processes



Use Case sections

- Title
- Actors
- Pre-conditions
- Normal workflow
 - Post-conditions
- Alternative workflows
 - Alternative post-conditions



First, describe Actor Steps...

Use Case Title

Temporary disconnect

Primary Actor

Patient

Other caregivers may assist the patient in performing steps of this use case.

Pre-conditions

An active therapy is in progress.

System is displaying active therapy status.

The system is in one of the active therapy's dwell phases.

Normal Workflow

Actor Step

Patient initiates Temporary Disconnect mode.

Patient confirms initiation of Temporary Disconnect mode.

Patient confirms closing the Transfer Set.

Patient confirms that the Sterile Disconnect Caps are open.

Patient washes hands.

Patient disconnects the Patient Line from the Transfer Set.

Patient places the Patient Line in the Organizer.

Patient caps the Patient Line.

Patient caps the Transfer Set.

Patient confirms that the Sterile Disconnect Caps are connected.

Post-conditions

The patient is free to move away from the system.

The system continues with the active therapy, timing the dwell phase.

Alternative Workflows

Patient takes no action for 30 seconds after the Temporary Disconnect confirmation prompt is displayed.

Patient declines to confirm the Temporary Disconnect confirmation prompt.

Patient takes no action for 30 seconds after the Close Transfer Set confirmation prompt is displayed.

Patient declines to confirm the closing of the Transfer Set.

Patient takes no action for 30 seconds after the Open Sterile Disconnect Caps confirmation prompt is displayed.

Patient declines to confirm the opening of the Transfer Set.

Patient takes no action for 30 seconds after the Connect Disconnect Caps confirmation prompt set is displayed.

Patient declines to confirm the connection of the Disconnect Caps.

Patient aborts therapy.



... then add System Responses

Normal Workflow

Actor Step

Patient initiates Temporary Disconnect mode.

Patient confirms initiation of Temporary Disconnect mode.

Patient confirms closing the Transfer Set.

Patient confirms that the Sterile Disconnect Caps are open.

System Response

System records initiating event to Event Log.

System displays Temporary Disconnect confirmation prompt.

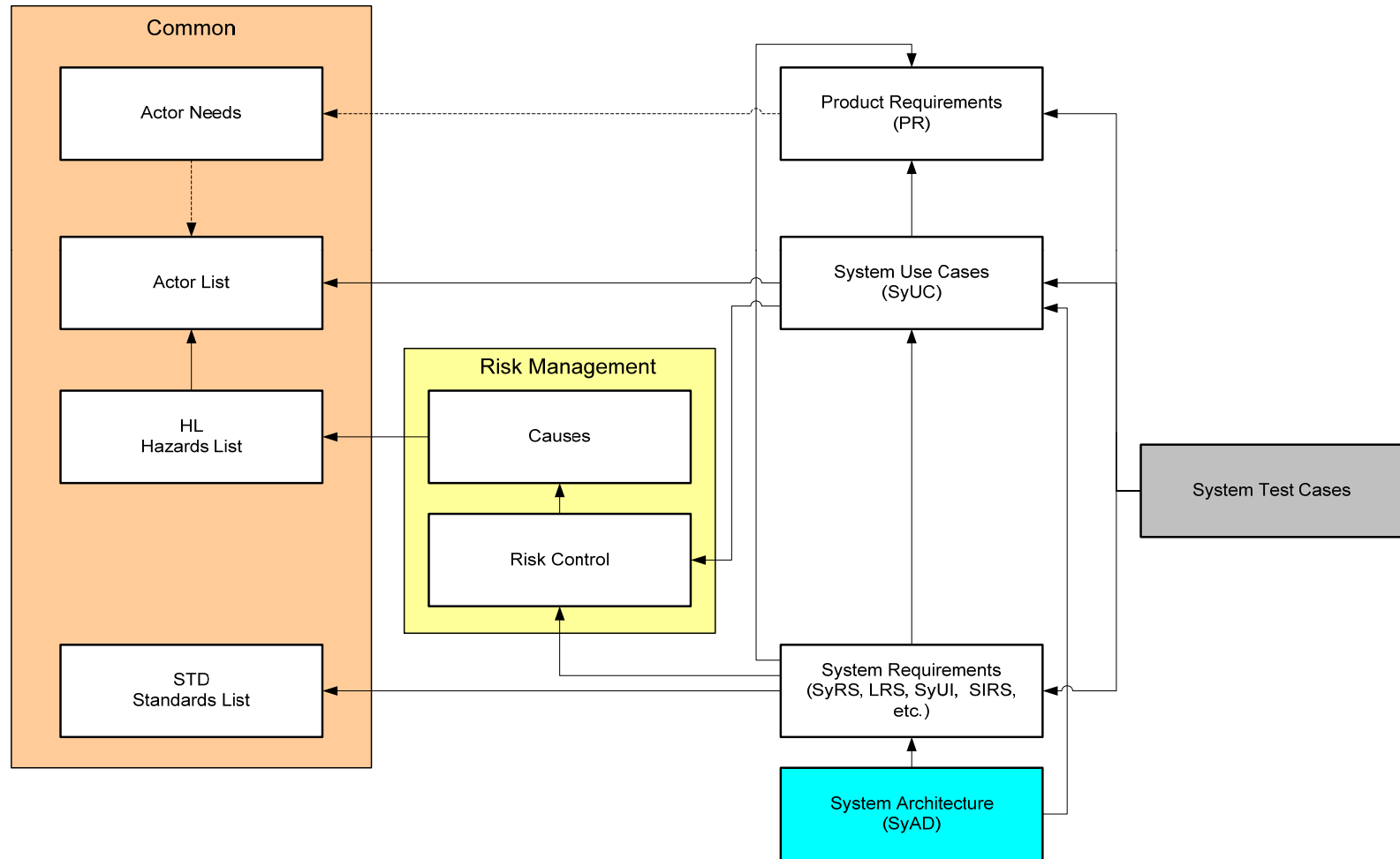
System closes valve
System displays Temporary Disconnect mode indicator.
System increments disconnected dwell cycle counter by 1.
System records date/time value to Therapy Log.
System records confirmation to Event Log.
System displays Close Transfer Set prompt.

Systems displays Open Sterile Disconnect Caps confirmation prompt.

System displays the Wash Hands prompt.
System waits 2 seconds.
System displays the Disconnect Patient Line prompt.

System waits 2 seconds.
System displays the Place Patient Line in Organizer prompt.
System waits 2 seconds.
System displays the Place Caps prompt.
System waits 2 seconds.
System displays the Press Confirm When Complete prompt.
System waits 2 seconds.
System returns to start of this loop to display the Wash Hands prompt above.

Use Case Tracing





Questions?

Good place to start:

"Writing Effective Use Cases" by Alistair Cockburn



About William K. Day

My email: wday63@gmail.com

My website: <http://williamkday.com/>

My LinkedIn profile: <http://linkedin.com/in/williamkday/>