Use Case Guidelines

Use Case Name / Goal

Main Scenario[†]

('happy day' or main success scenario)

Primary Actor/Role

Secondary Actors/Roles

Special* Cases (exception scenarios)

Variant* Paths (other possible scenarios)

Links:

to Definitions of Terms
to Exception Use Cases
to Included/other Use Cases
from System Requirements
from Test Cases
from analysis models (e.g. activity & sequence diagrams)
from design Objects (later)
to/*from* Dependencies

Associated Requirements:
Preconditions (initial state)
Triggers
Invariants
Frequency, repetition
Concurrency
Constraints
Stakeholders
Success Guarantee/Result
Failure Guarantee/Result

Other Attributes:
Scope (level of abstraction)
Priority
Change History
Decision Rationale
Open Issues

© Ian Alexander / JBA Ltd 2003

† *or possibly more than one*
* *both of these are often called 'alternative' courses/scenarios/paths*
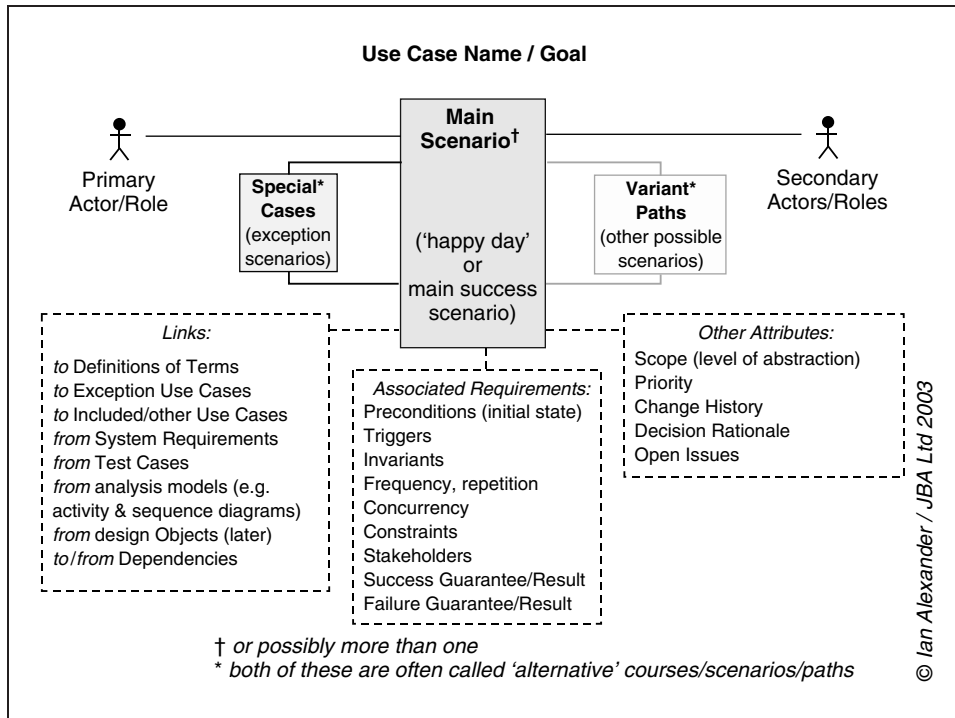
abbreviated from CREWS-L'ECRITOIRE *(see Chapter 8 for full details)*

General Guidelines

GG_1: Name each Use Case as an active goal (e.g. 'Publish a Report', 'Service the Car', 'Buy a Gift').
GG_2: Fill out a Use Case template (e.g. Templates in Appendix 1-3)—scenarios alone are not enough.
GG_3: In each Use Case, write a set of scenarios describing alternative ways to achieve the goal.
GG_4: Write all the scenarios of a Use Case in consistent terminology, style, and content.
GG_5: Validate your Use Cases with a checklist (e.g. these guidelines).

Style Guidelines

Style of scenario interactions:

SG_1: Describe scenario interactions in action clauses, e.g. 'The cook puts the prepared pizza into the oven.'
SG_2: Do not include circumstances, e.g. 'in the morning', 'at the office', 'quickly'.
SG_3: Describe what should occur; avoid negation (e.g. 'not', 'never') and modal verbs (e.g. 'could', 'may').

Style of scenario flows of actions:

SG_4: Describe sequences of interactions to tell a story; avoid flashbacks and forward references.
SG_5: Make explicit when repetitions and concurrency are needed.
SG_6: Avoid describing alternatives (e.g. 'if', 'else') within a scenario—describe alternatives separately.

Style of scenario initial states and final states:

SG_7: Do not describe initial and final states within scenarios—put them outside the scenario description.

Style of scenario terminology:

SG_8: Always use the same name for a given object or action; take care to avoid name clashes.
SG_9: Avoid ambiguous terms like pronouns (e.g. 'he', 'him').
SG_{10}: Do not refer directly to Use Case and scenario meta-models.

Content Guidelines

Scenario general content: single flow of interactions:

CG_1: Make each scenario describe a single flow of interactions to achieve an agent's goal. Describe alternative scenarios separately.

Scenario interaction contents:

CG_2: Describe Interactions in the general format: <Agent> <does> <something>. Examples: The Teller hands the Cash to the Customer. The Guard silences the Alarm. The Pilot acknowledges the Controller's command.

Content of flows of actions:

CG_3: Make each scenario tell a story, using a list of Interactions. Write one interaction per line.
CG_4: For repetitions, use the template: Repeat <Interactions> Until <Condition>
CG_5: State constraints explicitly, using the template: If <Condition> Then <Interactions>
CG_6: Interactions can be concurrent. Use the template: <Interaction> Meanwhile <Interaction>

Scenario initial and final states:

CG_7: Make each scenario satisfy initial states (preconditions) and end in final states (desired results).

Scenario content, with respect to goals:

CG_8: Make each scenario illustrate the achievement of a goal (such as handling an exception).

Scenario content, with respect to level of context:

CG_9: Describe each scenario at one level of context only:

- At the contextual level, scenarios describe flows of services among agents.
- At the functional level, scenarios describe interactions with the system.
- At the physical level, scenarios describe interactions between system components.