

Using Requirements Specification Patterns

David Gelperin
ClearSpecs Enterprises

In this paper, we recommend minimum sets of specification patterns for different types of systems. First we describe a comprehensive set of spec patterns and a classification scheme of system types. Then we identify the spec patterns to use for each type of system.

Specification Patterns

The following table describes the 18 ClearSpecs specification patterns and their purposes. More information on individual patterns is available among the technical reports available at www.clearspecs.com.

<i>Category</i>	<i>Spec Pattern</i>	<i>Purpose</i>
Overview		
	1. Background	Information that connects development work to organizational needs, decisions, and history
	2. Features	Natural language descriptions of the system features to be developed or modified
Usage Models		
	1. User Stories	Brief descriptions of the goal-directed activity of someone in an organizational role
	2. Use Cases	Detailed descriptions of interactive system usage to accomplish an organizational goal
	3. Scenarios	Detailed descriptions of a specific instance of system usage – may correspond to a path through a use case
	4. Test Specs	Detailed description of a system usage scenario with setup, wrapup, and checking steps

Category	Spec Pattern	Purpose
Behavior Models		
	1. Decision Tables	Detailed descriptions of the system actions resulting from a complete set of logical conditions
	2. State Transition Tables	Detailed description of the system actions and entity states resulting from various logical conditions and trigger events including initial entity states
Facts		
	1. Constant Conditions	Detailed descriptions of conditions that must be preserved during all system activity
	2. Condition Dependencies	Detailed descriptions of the logical dependencies between conditions
Derivations		
	1. Derived Value	Detailed definition of a calculated value by specifying the calculation
	2. Derived Condition	Detailed definition of an abstract condition (for example potential in potential customer) defined by Boolean functions referencing the attributes of domain and system entities
	3. Derived Action	Work breakdown structure of an action

<i>Category</i>	<i>Spec Pattern</i>	<i>Purpose</i>
Definitions		
	1. Entity Specs Internal Input Output	Detailed definitions of domain and system entities that describe their structure, required and optional attributes, value ranges, units of value and relationships to other entities
	2. Action Contracts	Detailed definitions of a system action using pre, post, and constant conditions
	3. Quality Specs	Detailed definitions of measures for assessing non-functional requirements
	4. Common Descriptions	Brief natural language descriptions of terms or phrases
	5. Acronyms	An acronym and the phrase its stands for

System Classifications

The following table shows the five problem frames proposed by Michael Jackson [Problem Frames: Analyzing and Structuring Software Development Problems, 2000] and described by Benjamin Kovitz [Practical Software Requirements, 1999]. More information is available in either reference.

Problem Frame	Description	Requirements Focus
<i>Information</i>	Requests for information about some part of the problem domain	Queries and Displays (R)
<i>Workpiece</i>	Operations on domain entities existing inside the software	Operations on entities (CUD)
<i>Transformation</i> [<i>Computation</i> <i>Classification</i> <i>Data Collection</i>]	Mapping between input and output data	Mappings
<i>Control</i>	Stimulus/Response rules for the problem domain	Behavior rules
<i>Communication</i> [<i>Connection</i> <i>Coordination</i>]	Routing information between systems	Linkage protocols

Spec Pattern Recommendations

Below, we recommend the specification patterns needed to capture the essential requirements information for each problem frame.

In addition to their essential frame-specific patterns, every requirements spec should contain the following basic patterns:

- Features
- Acceptance Test Specs (optional if precise use cases are included)
- Constant Conditions
- Condition Dependencies
- Input Entity Specs
- Derived Conditions
- Quality Specs
- Acronyms

Our recommendations identify a minimum set of specification patterns for each problem frame. Any pattern not included in a recommendation may be useful for a particular system. Therefore our failure to recommend does not infer a recommendation not to use.

The scope of many systems encompass multiple problem frames, for example workpiece and information (CRUD) or communication and control. The requirements for these multi-frame systems should contain the spec patterns of each component frame.

Information Retrieval Systems (R)

Examples:

- Part of an inventory control system: displays amount in stock of any item, prints reports of items low on stock
- Web-search engine finding pages on the worldwide web that match user-specified terms

Essential spec patterns:

- structure, content, and relationships of data structures
<internal entity specs>
- decision rules
<decision tables>
- structure and content of display options
<output entity specs>
<derived values>

Workpiece Manipulation Systems (CUD)

Examples:

- Word processor
- Reservation system
- Organic molecule modeler

Essential spec patterns:

- structure, content, and relationships of workpieces
<internal entity specs>
- semantics of operations
<action contracts>
<decision tables>
<state transition tables>
- usage patterns
<use cases>

Transformation Systems

Examples:

- Compiler
- Tax preparation system
- Decision support system
- Natural language recognition system
- Bar code scanner

Essential spec patterns:

- structure and content of output (and input)
<output entity specs>
<derived values>
- decision rules
<decision tables>

Control Systems

Examples:

- Inventory control
- Telephone switching system
- Traffic light control

Essential spec patterns:

- structure and content of controlled entities
<internal entity specs>
- decision rules
<state transition tables>

Communication Systems

Examples:

- Order routing system
- Message queuing system
- WWW

Essential spec patterns:

- structure and content of network entities
<internal entity specs>
- decision rules
<state transition tables>

The following table summarizes our minimal recommendations.

	Info	Work piece	Trans	Control	Comm
Background					
Features	✓	✓	✓	✓	✓
User Stories					
Use Cases		♦			
Scenarios					
Test Specs	✓	✓	✓	✓	✓
Decision Tables	♦	♦	♦		
State Transition Tables		♦		♦	♦
Constant Conditions	✓	✓	✓	✓	✓
Condition Dependencies	✓	✓	✓	✓	✓
Derived Values	♦		♦		
Derived Conditions	✓	✓	✓	✓	✓
Derived Actions					
Input Entity Specs	✓	✓	✓	✓	✓
Internal Entity Specs	♦	♦		♦	♦
Output Entity Specs	♦		♦		
Action Contracts		♦			
Quality Specs	✓	✓	✓	✓	✓
Common Descriptions					
Acronyms	✓	✓	✓	✓	✓

The checkmarks (✓) mark the basic patterns needed by any system. The diamonds (♦) mark the essential patterns that characterize type-specific information.