The Requirement Specification Framework (RSF)
Model Driven Systems Engineering for Energy System Operators

OPE/INC
April 2014
Definition: The Requirement Specification Framework (RSF)

The Requirement Specification Framework (RSF) is an integrated and harmonized collection of support tools for use in energy related projects that aim to improve the quality and efficiency of the requirement specification management process.

**Tools**: Standards (IEC-IRM), Software, (Use-case) Templates, Structures, Selected Modelling Languages (UML/ArchiMate), Predefined Requirement Specifications, etc.

**Quality**: Consistency, Testability, Completeness, Understand-ability, Traceability, Impact Analysis, Sharing & Leveraging, Customization, Documentation, etc.

**Process**: Requirement Elicitation, Communication with Stakeholders, Conformance verification, Change Management, Estimation, Weighing & Scoring, Structuring, etc.
The V-model of the system development/procurement cycle

- Concepts of Operation
- Architecture & Requirements
- Specification & Design
- Procurement & Implementation
- Integration & Validation
- Module Test & Verification
- Operation & Maintenance

The Requirement Specification Framework (RSF)

Quality Loop

Time
Contents

1. A short introduction to modelling and its benefits (9 slides)
2. The Requirement Specification Framework (RSF) (5 slides)
3. An example: the State Estimation (SE) use-case (20 slides)
A short introduction to modelling and its applications
A short introduction to modelling and its benefits (and costs)

A model: what the architect does with cardboard...

Simplification & visualization for communication purposes
Why use a model?

- To manage complexity
- To use a formal language (UML)
- Generate Documentation / Code
- Structures & methods (e.g. SysML)
- Re-use libraries of model-elements
- Ensure consistency / reproducibility
- Simplify maintenance of products
- Helps in leveraging and education
- Reduces the development effort
- To “make smarter decisions”
How to use models? Or rather: how to use a model repository?

- We already use (2D) models (whiteboard, Visio, flow-charts, napkins, etc.)
- A modelling language (UML) relies on a repository (e.g. Enterprise Architect)
- From the (Multi-D) model-database various diagrams & views can be created
- Diagrams are used for Data Entry (to create the model) and Visualization
Features that are needed to achieve the goal (manage complexity)

A model supports all of the interdependencies and relationships between different aspects (actors, processes, functions, platforms, etc.) and provides visualizations.
A model represents the real-life complexity that needs to be managed

- A model represents all of the real-life complexities that need to be managed
- Models exclude complexities that do not have to be managed (in this context)
- A model adds some (temporarily) complexity (language, methodologies, tooling)
- Software tooling is needed to effectively use models for (specification) tasks
RSF applies Black-box modelling on a (top down) functional level

- How the system is “used” by entities in its environment (hence: use-case)
- As opposed to a “White-box” model that defines the system’s “inner workings”
Requirement Realization Options

- The model defines Requirement-owners that cause (or realize) the requirements
- The realization dependencies are mainly used for traceability and impact analysis
- Other dependencies (data flows) and containments (packages) are also supported
The Advantages of Model Based (or Driven) System Engineering

- The main aim of Model Based (or Driven) System Engineering (MBSE or MDSE) and the RS-Framework is to Manage Complexity. A model (tool) is used to:
  
  - View systems and environments from multiple perspectives or viewpoints
  - Improve system understanding through visualization & validation functions
  - Perform impact analysis (discover causes & effects using model traceability)
  - Identify the consequences of changes/alternatives early in the system lifecycle
  - Improve estimation and risk-susceptibility of modifications & implementations
  - Capture and share the expert knowledge in a persistent (baselined) repository
  - Reuse existing information and lessons-learned in new projects (efficiency)
  - Experiment with systems & process migrations without any risks

- The model becomes the backbone of the systems design/procurement process
- The model provides the reference for verification along the full system lifecycle
The Requirement Specification Framework (RSF)
Some RSF Features

- Driver for RSF: Word documents are too difficult to manage and keep consistent.
- RSF creates Word documents from a model that helps management & quality:
  - Strict implementation of the Systems Engineering (ISO/IEC 15288) methodology
  - An integrated approach on a functional level (use-cases) for the system life-cycle
  - Defines the business processes, information, application functions & infrastructure
  - Common, holistic, view on the subjects for all stakeholders (multi-disciplinary)
  - Early verification & validation of requirements & specs to reduce correction efforts
  - Based on (ISO/IEC/W3C) standards and sector best practices (ENTSO, EPRI, etc.)
  - SMART, pre-defined, requirements for Interface Reference Model (IRM) use-cases
  - Supported by formal modelling languages (UML) and tooling (Enterprise Architect)
Introduction
What, Where, When, Who, How, & Why (the Zachman framework extended with:)
Which Verification
Which Environment
Which Requirements

Containment
the IEC-61968-1 Interface Reference Model (IRM) chapters extended with:
SCADA functions and field devices
Processes & project implementation
Infrastructures and facilities

Use Case
Description (Block definition diagram)
Use case diagram (Use-case diagram)
Interaction analysis (Sequence diagram)
Functional description (Activity diagram)
Requirement specification (Alignment)

Requirement Library

Predefined templates

Environment Library
Enterprise (ArchiMate) model layers:
Business processes and products
Application and system functions,
Infrastructure & platform services,
Migration & motivation extensions

Test cases & scenarios

Akers & Roles Library
the IEC Use-case repository
Actors & Role model conforms with:
ENTSO-E, DKE, EG3, SMCG/SGCG. (Extended with project actors & roles)

Note: Predefined standard specifications for SCADA, EMS, DMS, OMS, AMI, RTU, Cabinets, etc.
The interface Reference Model (IRM: IEC 61968 part 3-10)

**Distribution Management Business Functions**
- **(NO) Network Operation** (Interface Standard: Part 3)
- **(AM) Records & Asset Management** (Interface Standard: Part 4)
- **(OP) Operational Planning & Optimization** (Interface Standard: Part 5)

**Business Functions External To Distribution Management**
- **(EMS) Energy Management & Energy Trading** (Interface Standard: Part 10)
- **(RET) Retail** (Interface Standard: Part 10)
- **(SC) Supply Chain and Logistics** (Interface Standard: Part 10)

IEC 61968 Compliant Middleware Services
- **(NE) Network Extension Planning** (Interface Standard: Part 7)
- **(CS) Customer Support** (Interface Standard: Part 8)
- **(MR) Meter Reading & Control** (Interface Standard: Part 6)

**Electric Distribution Network Planning, Constructing, Maintaining, and Operating**

**Generation and Transmission Management, Enterprise Resource Planning, Supply Chain, and General Corporate Services**

17 DNV GL © 2014
The use of use-cases to subdivide the scope in function “blocks”

- Use-case template (IEEE-EPRI IntelliGrid Template - IEC-PAS 62559 standard)
- Contained in the IEC Interface Reference Model (IRM) structure
- With dependency links and associations to context elements
Requirement Specification Framework
- Example: The State Estimation Use-case
Example: The State Estimation Use-case

### Zachman Framework

<table>
<thead>
<tr>
<th><strong>Perspective</strong></th>
<th><strong>Scope</strong></th>
<th><strong>Introduction WHAT</strong></th>
<th><strong>Introduction HOW</strong></th>
<th><strong>Introduction WHO</strong></th>
<th><strong>Introduction WHY</strong></th>
<th><strong>Introduction WHEN</strong></th>
<th><strong>Introduction WHERE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Contextual</strong></td>
<td>What (goal aims)</td>
<td>How (project)</td>
<td>Who (people)</td>
<td>Why (motivation)</td>
<td>When (time)</td>
<td>Where (location)</td>
<td></td>
</tr>
</tbody>
</table>

**Perspective Business Model**

<table>
<thead>
<tr>
<th><strong>Scope</strong></th>
<th><strong>Business Layer</strong></th>
<th><strong>Archimate Business Layer</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logical</strong></td>
<td>Passive Structure Data Objects</td>
<td>Behavior Functions</td>
</tr>
<tr>
<td><strong>Physical</strong></td>
<td>Passive Structure Artifacts</td>
<td>Behavior Platform Functions</td>
</tr>
</tbody>
</table>

**Perspective System Model**

<table>
<thead>
<tr>
<th><strong>Scope</strong></th>
<th><strong>Application Layer</strong></th>
<th><strong>Archimate Application Layer</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logical</strong></td>
<td>Passive Structure Data Objects</td>
<td>Behavior Functions</td>
</tr>
<tr>
<td><strong>Physical</strong></td>
<td>Passive Structure Artifacts</td>
<td>Behavior Platform Functions</td>
</tr>
</tbody>
</table>

**Perspective Technology**

<table>
<thead>
<tr>
<th><strong>Scope</strong></th>
<th><strong>Technology Layer</strong></th>
<th><strong>Archimate Technology Layer</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logical</strong></td>
<td>Passive Structure Data Objects</td>
<td>Behavior Functions</td>
</tr>
<tr>
<td><strong>Physical</strong></td>
<td>Passive Structure Artifacts</td>
<td>Behavior Platform Functions</td>
</tr>
</tbody>
</table>

**Perspective Representation Sub Contractor**

<table>
<thead>
<tr>
<th><strong>Scope</strong></th>
<th><strong>Archimate Technology Layer</strong></th>
<th><strong>Archimate Technology Layer</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Logical</strong></td>
<td>Passive Structure Data Objects</td>
<td>Behavior Functions</td>
</tr>
<tr>
<td><strong>Physical</strong></td>
<td>Passive Structure Artifacts</td>
<td>Behavior Platform Functions</td>
</tr>
</tbody>
</table>
Example: The State Estimation Use-case

- The RSF model for the State Estimation use-case:
  - Introduction (Zachman structure)
  - Context (ArchiMate 2) model
  - Requirements & Use-cases
  - Various other packages

  - The Context model serves many use-cases
  - The Use case repository (according to IRM)

Shown on the left are the top-level packages
Of RSF in the Enterprise Architect software tool (the Project Browser)
Example: The State Estimation Use-case

The Requirements & Use Cases package

- Holds the use-cases & requirements (for UCs)
- Inside the IRM Containment structure
- The State Estimation use-case
  - Is inside the CLC Package
  - Which is inside the NO Package

The structure and containment is compliant with the IEC 61968 part 3/10 (CIM) standard.

Use-case repositories (EPRI, IEC, Europe SGCG) all use the same structure and abbreviations.
Example: State Estimation Use-case

The State Estimation use-case

- Largely in line with the EPRI repository
- A generic starting point for specific models

https://www.smartgrid.gov/document/state_estimation

- The internal structure is according to the
- IntelliGrid (now PAS/IEC-62559) template

- The package can be used to document
- use-case specific features using the
- documentation (notes) features of EA.

- Next page shows the coverage of the RSF
- Use-case template with the PAS standard.
The RSF Use-case template compared to IntelliGrid (IEC-62559)

### The RSF Context Model
- Actors & Roles
- Business Processes
- Application Functions
- Infrastructure Platforms
- Data & Business Objects

### The IntelliGrid (IEC-62559) Template
- 1 Descriptions of Function
  - 1.1 Function Name
  - 1.2 Function ID
  - 1.3 Brief Description
  - 1.4 Narrative
  - 1.5 Actor (Stakeholder) Roles
  - 1.6 Information exchanged
  - 1.7 Activities/Services
  - 1.8 Contracts/Regulations
- 2 Step by Step Analysis of Function
  - 2.1 Steps to implement function
  - 2.2 Architectural Issues in Interactions
  - 2.3 Diagram
- 3 Auxiliary Issues
  - 3.1 References and contacts
  - 3.2 Action Item List
  - 3.3 Revision History

### The RSF Use-case Template
- 1.1 Use-case description
- 1.2 Overview and scope
- 1.4 Functional description
- 1.5 Data Model & Data Flow
- 1.3 Interaction analysis
- 1.6 Requirement Specification
- 1.7 Verification and Testing
- Revisions and Remarks

Note: The sequence of Use-case sections is changed for clarity.
The RSF Use-case template uses codes as part of the use-case name to support the function ID (in section 1.2).
The RSF Template provides support for various diagrams in every section (not just a sequence diagram in 2.3).
Example: The State Estimation Use-case

- The use case is documented using the UML language and diagrams (with notes)
- The diagrams may use many elements from the context model (blue & yellow)
Example: The State Estimation Use-case

- Predefined templates model elements, structure, requirements, and other items
- The scope diagram is used to map responsibilities (RACI) and use-case scope
Example: The State Estimation Use-case

- Sequence diagrams to document Sequence of Events (normal, exception, alt...)
Example: The State Estimation Use-case

- Diagrams are used to provide unambiguous data on interfaces and messaging
- The language used matches the common vendor software engineering approach
Example: The State Estimation Use-case

- Data flow and structure analysis references CIM profiles and packages directly.
- The mapping of Business Objects & CIM classes is part of the context model.

The State Estimation use-case relies on data that is defined by the CIM explicit connectivity model (see below). This package is an extension to the CIM Core Package that in association with the Terminal class models Connectivity, that is the physical definition of how equipment is connected together. In addition it models Topology, that is the logical definition of how equipment is connected via closed switches. The Topology definition is independent of the other electrical characteristics.
Example: The State Estimation Use-case

- The requirement alignment (custom) diagram is “invisible” for documentation.
- Its purpose is for model maintenance only. It allows to define dependencies.
- The diagram below shows the requirements for a sub-use-case & aggregation.
- Many dependencies in the model exist for traceability & impact assessment.

The State Estimator (SE) shall produce a complete solution for the power system model, including bus voltages and angles, for the observable and unobservable areas of the power system model.
Example: The State Estimation Use Case

Use cases feature:

- Attributes
- Constraints
- Scenarios
- Other Links
- Other Files
- & Requirements

Requirements are

- Internal (none)
- External (8)
- Visible or Invisible
Example: The State Estimation Use-case

The associations in the model allow for traceability and impact assessment

- What-if analysis
- Optimization options
- Risk susceptibility
- Transformations
- Drill-down views
- Life Cycle support
  - Design, As-is, Deployment, etc.
  - Gap analysis (from baselines)
- Glossary (references, authorities)
- Relationship Matrix / Coverage Map
Example: The State Estimation Use-case

Requirements:
- Approved
- Implemented
- Mandatory
- Proposed
- Validated

Example: The State Estimation Use-case

- Document Generation using (Predefined) Templates for any repository viewpoint
- Document management using fragments, master documents & custom layouts.
Example: The State Estimation Use-case

- State Estimation Use-case document in MS-Word: Consistent & Up-to-date!
- 6 visible diagrams (+5 invisible diagrams in the model, not included in the output)
Requirement Specification Framework
- Remainders
Origins of the RSF Framework and touch points with other services

- Energy Sector Knowledge. Systems, roles, responsibilities
- Information Modelling & Data (Quality) Management
- Management & Optimization of (core) Business Processes
- Quality Control and Assurance. Methodologies & Auditing.
Context sensitivity of (generic & specific) RSF Components

Tools & Methods
- Zachman & Archimate
- Unified Modelling Language (UML)
- Sparx Enterprise Architect (Tool)

…
- IntelliGrid Use Case Template
- IEC Interface Reference Model
- Requirement Verification Testing

Context Specific
- SCADA/EMS/DMS
- Standard Specification
- Actors & Roles Model

Increase of complexity, level of detail, and (content matter) expertise required
Balancing between business needs and standard solutions

Standard
- Functions
- Data Model
- Infrastructure

Adaptation
- of Processes
- of Requirements
- of the system (through configuration)

Custom
- Software Functions
- Information Model
- Hardware Platforms

Increase of (initial) complexity, efforts, costs & risks