Corporate Services for the Natural Resource Sector

Information Management Branch

NRS Standards for Modeling with EA:

DESIGN

Last Updated:	May 10, 2017
Version:	5.0.0
Document:	NRS_Standards_for_Modeling_with_EA-Design

Table of Contents

Version Control

Document Version	Revision	Date	Author(s)	Change Reference
1.0.0	Final	2014-04-17	Clover Mohr	Modified
2.0.0	Draft	2014-08-11	Diana Von Ratenberg	Changes: RMP Reset work
3.0.0	FINAL	2015-01-27	Diana Von Ratenberg	V1.0 posted
3.5.0	FINAL	2015-08-31	Noel Carawan-Hubin	V3.5.0 Posted
4.0.0	FINAL	2016-06-24	Design Modernization Project	V4.0.0 Published
4.1.0	FINAL	2016-12-08	Sparx Enterprise Architect Working Group (SEA)(formerly DMP)	V4.1.0 Published
5.0.0	FINAL	2017-05-10	SDLC Sub Committee – Standards Sub Team	Alignment to new SDLC

1 Introduction

This document describes the standards that must be followed for the Design phase for NRS IM/IT projects following the new SDLC.

This standard applies to projects using Sparx Systems Enterprise Architect.

If a project plans to deviate from these standards, or from the specified versions of the tools and standards, the project must first be approved to do so by the IM/IT Investment Board.

This standard assumes that project teams have completed the "Project Initiation Meet (with PMO)" in the Initiation Phase of the project and have been directed to use EA. The outputs of this meeting include determination on SDLC methodology, the standard and tool to fit the project, and a decision about how the project team will visually represent business processes – either BPMN models or UML activity diagrams.

An NRS-specific template, ***NRS EA Template***, has been provided to guide the process of entering the necessary information into EA. The template is organized to include a package for each artifact to be entered.   If a package is not used, the section should be removed from the template.

Artifacts and attributes of artifacts which are optional are noted by the term **[OPTIONAL].**

All mandatory components within this standard are noted by the term **[MANDATORY].**

Where a component is **[OPTIONAL]** but a project team decides to include the component in their model, there may be sub-components which are mandatory. Also, there are some helpful hints provided in this document. Both of these are indicated by 'Notes' with an exclamation symbol: ❗

Hyperlinks to related published standards are inserted in the standard as required.

1.1 Design Phase Deliverables

The Software Design Description (SDD) is a document that describes what the software is expected to do and how it will be expected to perform. The SDD is formalized by the IEEE 1016 standard. When using the Sparx Enterprise Architect tool, the SDD can be generated from the tool as an RTF or Word document and edited manually (if required).

As depicted in Figure 1, the SDD pulls together the Functional Requirements, System Uses Cases, Business Rules, BPMN/UML Activiity diagras, the fully populated Logical Data Model, Wireframes (optional) and System Components. The component design may contain various types of diagrams: class, sequence, state machine, and component.

The Generate Document Package has been customized to generate an NRS SDD. The primary purpose of the SDD is communication – the document is a mechanism to formally communicate the system design to clients. It is likely that the generated SDD will require some manual editing before it is ready for clients to review. Note however that the EA file contains the complete design. The SDD is a convenient tool used to socialize the proposed design, especially among project staff and reviewers that may not have access to Sparx EA.

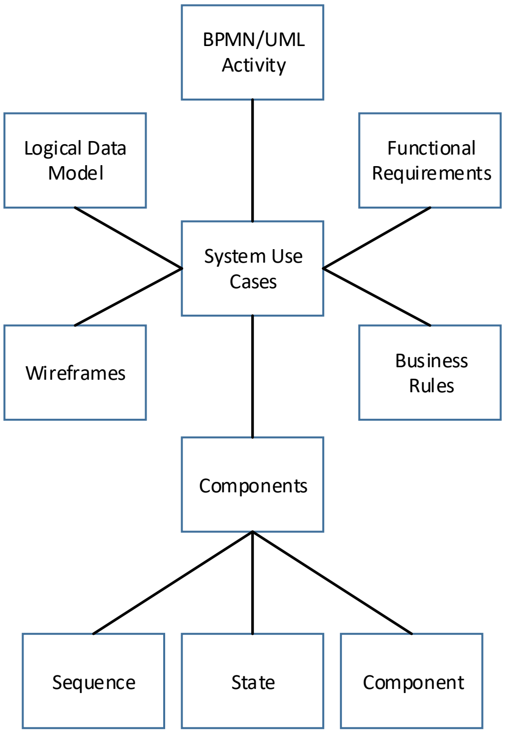


Figure 2 - System Design

2 Overview [OPTIONAL]

This section elaborates the body of work represented in the EA file from requirements.

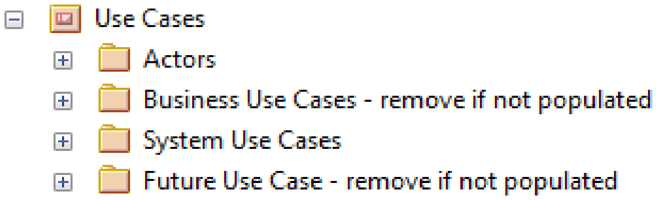
2.1 System Overview for Design Phase

For the Design Phase the following artifacts can be completed/updated by entering relevant content into the *Notes* field:

Purpose: **Scope:** (from Requirements)	Update what was entered during Requirements if any of this information has changed or needs to be extended for the Design Phase.
Overview:	Provide an overview of system scope and design. Scope description should be consistent with higher-level project document (e.g. Project Charter).

3 System Use Cases [MANDATORY]

During the design phase, System Use Cases must be fully specified. System Use Cases are the key deliverables of an NRS system design document. Use Cases specify a list of actions between the actor and the automated system.


- Use Cases
  - Actors
  - Business Use Cases - remove if not populated
  - System Use Cases
  - Future Use Case - remove if not populated

3.1 Actors Attributes [MANDATORY]

Note: Project teams must use the same Actor names for use cases as used in defining the BPMN lanes.

Actor Elements	Properties Dialog Box	Field / Button	Required Attributes (unless specified otherwise)
Actor:	Properties > General	**Name**	Name of Actor
		Notes	Short description of Actor

3.2 System Use Cases [MANDATORY]

A System Use Case describes a series of related interactions between an actor and the “system”.

❗ System Use Cases are mandatory during the Design Phase.

3.2.1 Use Case Diagrams [OPTIONAL]

Use Case Diagrams may include the subject for a set of Use Cases (sometimes referred to as a system boundary) by using the Boundary element from the Use Case Diagram toolbox, and shown as a rectangle with its name at the top and the Use Case ellipses visually located inside or outside of this rectangle.

3.3.2 Use Case Attributes

Use Case & Actor Elements	Properties Dialog Box	Field / Button	Required Attributes (unless specified otherwise)
Use Case:	Properties > General	**Name**	Name of Use Case
		Notes	Short description of Use Case
		Stereotype	<<system use case>>
		Complexity	**Easy** (default) (Used for estimating effort) ****Must be set to the appropriate value for each Use Case**
		Author	**Last name, First name**
		Status	**Proposed** (default)
	Rules > Scenarios	**Scenario**	Overtype ‘Basic Path’ with name of Scenario
		Type	**Basic Path** (default) – the direct set of steps for the scenario, individual steps within the Use Case <Optional> **Alternate** – an alternative set of steps, in parallel with part of the basic path <Optional> **Exception** – the path the scenario follows if a step of the basic path does not produce an appropriate result
		Structured Specification tab (default)	Steps (individual steps, actions, within the use case) Prefix each step with the Actor name
		Description Tab	Simple high level text description of scenario path (Used to provide more detailed description of the path, but does not replace Structured)
	Rules > Constraints	**Constraint**	Short description of the Constraint
		Notes	Detailed description of the Constraint

Use Case & Actor Elements	Properties Dialog Box	Field / Button	Required Attributes (unless specified otherwise)
		Type	**Pre-condition** (condition which must exist before the use case) **Post-condition** (condition which must exist after the use case) **Invariant** (condition which must not change during execution)
		Status	**Proposed (**NRS custom default)

3.3.3 Defining Use Case Scenarios

Context References Tab

All elements involved in the scenario must be identified in the Context Reference tab. The Context References tab is used to add traceability for downstream artifacts. EA auto fills all explicit links to the Use Case, which would include both the Actors and Functional Requirements where relationships have been created. As long as these relationships exist, they will present themselves in the reference listing. DO NOT add Functional Requirements here, as this is not the place.

For downstream traceability and the generation of test cases, you could add Logical Data Objects.

Develop Your "Structured Specification" Scenarios

For each step, you can specify in the Uses column any model elements that are referred to by that action. You can also note significant results whenever a test is desired by using the Results column.

Uses Column – Business Rules and Non-Functional Requirements

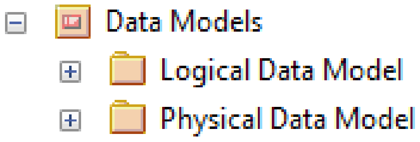
Business Rules should be linked to the Use Case by including content in the Uses field of a step.
Right-click in field | Link to existing element | Requirement | Dependency

Right-click in field | Link to existing element | Feature | Realization

In EA, you will notice that functional requirements will display as a *feature* for element type. For business rules, these will display as a *requirement* for element type.

4 Data Models [MANDATORY]

During the design phase, the logical data model must be fully elaborated. The elaborated logical data model should include all classes, class associations, attributes and keys. Once the logical model is complete, the physical data model can be generated from the logical model (captured in Sparx Enterprise Architect) using the transform scripts provided.



4.1 Logical Data Model

The IMB Data Architecture team has developed the document *NRS Data Modeling Standards with EA* which further describes data modeling requirements. Refer to *Section 4.0 – Logical Data Model* for details.

4.2 Physical Data Model

For detailed information concerning the the Physical Data Modeling standards, please refer to the *NRS Physical Data Modeling Standards.*

4.2.1 Generating the Physical Data Model

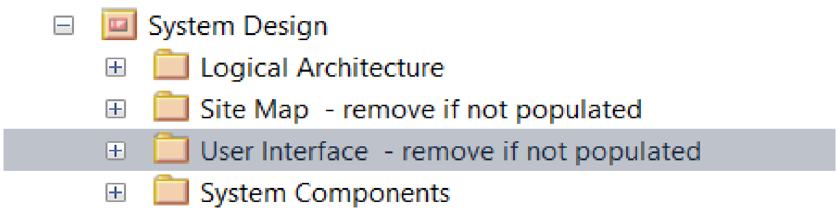
The logical data model can be generated via a script that transforms the logical data model to produce a physical data model. A number of steps are required as part of the transform operation. NRS has created a document that describes the process necessary to complete the transformation. Please refer to the *NRS Logical Data Model to Physical Data Model Transformations* document. Refer to Section 2.3 – NRS DDL Transformation Templates for supplemental XML documentation.

5 System Design

The design consists of up to four artifacts:

1. The **logical architecture model** depicts the application specific components that will be delivered.
2. The **site map** provides a high-level overview of the navigation options in the user interface.
3. The **user interface** artifacts consist of low-fidelity wireframes.
4. **System components** provide the design specifications for the component "building blocks" that make up the system.

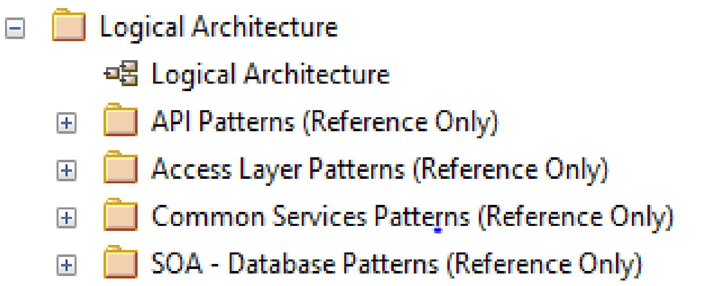
If the component specification(s) do not include any user interface components, then the "Site Map" and "User Interface" folders are to be considered **[OPTIONAL]** and should be deleted.



System Design
- Logical Architecture
- Site Map - remove if not populated
- User Interface - remove if not populated
- System Components

5.1 Logical Architecture **[MANDATORY]**

The logical architecture for the system is specified using the logical architecture diagram. The diagram provided within the NRS EA Template is constructed from components that reside in the four sub-folders of component patterns. The sub-folder of components are marked "Reference Only" but can be extended as required, and include the patterns. The system designer indicates which of the ISSS infrastructure components are being utilized for the system under design by overlaying a system specific icon on the system component.



Logical Architecture
- Logical Architecture
- API Patterns (Reference Only)
- Access Layer Patterns (Reference Only)
- Common Services Patterns (Reference Only)
- SOA - Database Patterns (Reference Only)

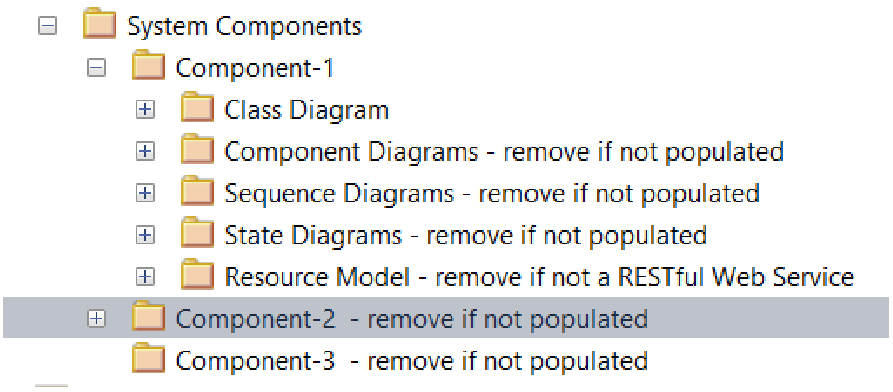
[OPTIONAL] Common components provided by the ISSS infrastructure are shaded in light green, and must be left unaltered. Dependencies on legacy systems outside of the ISSS infrastructure can be added to the diagram as required.

It is highly recommended project teams deliver their Logical Architecture Diagrams to the Release Manager as early as possible in Design Phase.

5.2 System Components **[MANDATORY]**

A component-based software design is the foundation for designing and building loosely coupled systems that are assembled from independent components. Components are modular and reusable. Architects and developers should specify the system under design using multiple components.

It is **[MANDATORY]** for project teams that at least one Component is defined in System Components.



- ⊟ System Components
  - ⊟ Component-1
    - ⊞ Class Diagram
    - ⊞ Component Diagrams - remove if not populated
    - ⊞ Sequence Diagrams - remove if not populated
    - ⊞ State Diagrams - remove if not populated
    - ⊞ Resource Model - remove if not a RESTful Web Service
  - ⊞ Component-2 - remove if not populated
  - Component-3 - remove if not populated

5.2.1 UML Sequence Diagram *[OPTIONAL]*

Complex Use Cases can be further described with a UML Sequence diagrams. The diagram can be generated from the Use Case as follows: Right-click on System Use Case --> Add --> Add Diagram --> Sequence

5.2.2 UML State Diagram *[OPTIONAL]*

Complex use cases can be further described using UML State diagrams. . The diagram can be generated from the Use Case as follows: Right-click on System Use Case --> Add --> Add Diagram --> State Machine

6 Reporting **[OPTIONAL]**

If reporting is specified, it is to consist of two deliverables within the model:

- A report layout, which can be textual or a graphical image
- A specification, based upon the logical model, that lists report data elements

Note: The build specification for reporting will reference Jasper Reports and the Jasper Reporting standards. Please reference the NRS Standards Inventory.

7 Security **[MANDATORY]**

Project teams must continue their engagement from the Requirements Phase with the appropriate IMB Security Architect for their project at the start of the Design Phase. In addition, project teams should be familiar with the OCIO's Information Security Policy. For additional guidelines on Security and Privacy Best Practices, please reference this FAQ.

7.1 Security and Role Mapping

Role Mapping identifies who may access system information. Project teams must list and describe the WebADE roles that may access the system and make note of any special cases, such as those involving sensitive or restricted data.

8 Traceability [OPTIONAL]

Traceability is used to document links between artifacts. Traceability provided at the design stage should include:

Matrix Specification Artifacts & Profiles	Purpose	Source Package (Type)	Target Package (Type)	Link Type Direction	Overlays
Artifact: <<EAmatrixspecification>> **PPPP System Use Cases to Functional Requirements**	Identify orphan requirements	Use Cases (UseCase)	Functional Requirements (Requirement)	Realization Source->Target	<None>
Artifact: <<EAmatrixspecification>> **Functional Requirements to Logical Data Model**	Identifies who is able to execute which use cases	Use Cases (UseCase)			CRUD
Artifact: <<EAmatrixspecification>> **System Use Cases to User Interface (Screens)**					

9 Generate Documents – SDD [MANDATORY]

There is one deliverable related to design, the SDD.

The generate documents package is strictly a tool for generating the noted deliverables using the custom NRS document templates. Manual editing of the generated document may be required to produce a System Design Document for review.

System Design Document: Documents the design of an automated solution that satisfies the business requirements. The Design is specific to the technical environment that the system will be required to operate in and the tools to be used in building the system. The SDD is automatically generated from the EA file.

! Note: Project teams should be aware that the design for individual component can be printed individually. For more information, reference Sparx Systems EA User Guide "Generate Documentation".

10 Quality Assurance Review for Models

The following "Quality Assurance for Sparx EA Models" is available for project teams to guide them through the quality assurance reviews on the Sparx EA Working Group SharePoint Site.

The Business Portfolio Manager (BPM)/Business Transformation Project Lead (BTPL) is accountable for the successful completion of the quality assurance reviews for each of their respective projects.

11 Submission of Sparx EA Models

Project teams are expected to submit their models following the completion of Interim and Final Quality Assurance Reviews within both the Requirements and Design Phases of the SDLC.

For project teams new to the submission process for models into the NRS Sparx EA Repository, please email NRS.SEA@gov.bc.ca (Sparx EA Working Group) for support. Additional information on competencies for use of the NRS Sparx EA Repository environment is available on the Sparx EA SharePoint Site FAQ, specifically the Sparx EA Repository Competency List.

For project teams familiar with the NRS and the use of Tortoise SVN, please check out the documentation available on the Sparx EA Working Group SharePoint Site to ensure you are setup and ready to submit your *.xmi file export from Sparx EA into subversion. Please contact NRS.SEA@gov.bc.ca with any questions.