ACHIEVING RELIABILITY OBJECTIVES WITH THE AID OF A SERVICE ORIENTED ARCHITECTURE

Brian Leonard, California ISO
Greg Robinson, Xtensible Solutions

Abstract

This paper describes why and how a Service Oriented Architecture (SOA) is being implemented by California ISO to realize its business processes. After a brief overview of the program scope, a general description is given of the process being used in the program to specify how business requirements - such as those for reliability - are modeled and then orchestrated in the SOA. This includes articulating the sequence and contents of information exchanged among disparate applications across organizations. To support each exchange, services are defined in Web Services Description Language (WSDL) whereby the abstract service definition is automatically built from California ISO's semantic data model, which is an extension of the utility industry's Common Information Model (CIM).

Introduction

California ISO has initiated a program called Market Redesign and Technology Upgrade (MRTU). The program is mitigating several critical issues that exist in the current market structure in California. As this program implements the new market functionality, the opportunity is being taken to improve corporate-wide project management process, realize efficiencies in information technology, data architecture and management, and to provide a prudent amount of design flexibility to minimize vendor "lock-in" for any future changes to systems. The overall scope of MRTU is depicted in the following diagram.

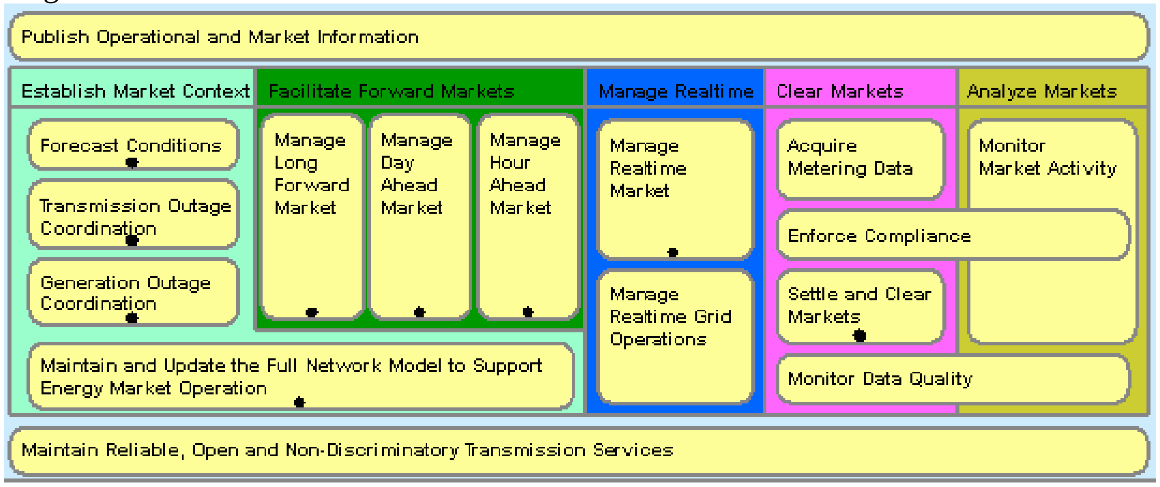

Publish Operational and Market Information
Establish Market Context
Facilitate Forward Markets
Manage Realtime
Clear Markets
Analyze Markets
Forecast Conditions
Transmission Outage Coordination
Generation Outage Coordination
Manage Long Forward Market
Manage Day Ahead Market
Manage Hour Ahead Market
Manage Realtime Market
Manage Realtime Grid Operations
Acquire Metering Data
Enforce Compliance
Settle and Clear Markets
Monitor Data Quality
Monitor Market Activity
Maintain and Update the Full Network Model to Support Energy Market Operation
Maintain Reliable, Open and Non-Discriminatory Transmission Services

Figure 1: Overview of Market Redesign and Technology Upgrade

As can be gathered from the diagram, the scope of this program is large, necessitating the effective management of a large number of parties and their products. The Service

Oriented Architecture (SOA) has been deemed to be the overarching principal for integration design to provide a flexible and scalable integration infrastructure for the program. It is widely known in the Information Technology industry that a Service-Oriented Architecture (SOA) is extremely flexible and scalable because the options for deploying services are fairly limitless. As such, it serves as a good basis for utilities to automate processes that transcend the unending array of business and technology changes. However, the flexibility of SOA also makes integration more complex because many more decisions are required, for example ensuring services are business process oriented, determining the appropriate granularity of services, determining how services are aggregated into higher level services, managing data ownership and replication, and determining how business processes are orchestrated and monitored. The remainder of this paper provides highlights of California ISO's general approach for addressing these matters in its SOA.

Defining the Services of a SOA

The California ISO SOA services are classified as shown in figure 2 below. The detail for service classifications, include naming convention, execution mode, service payload and attributes of a service, are discussed in subsequent sections.

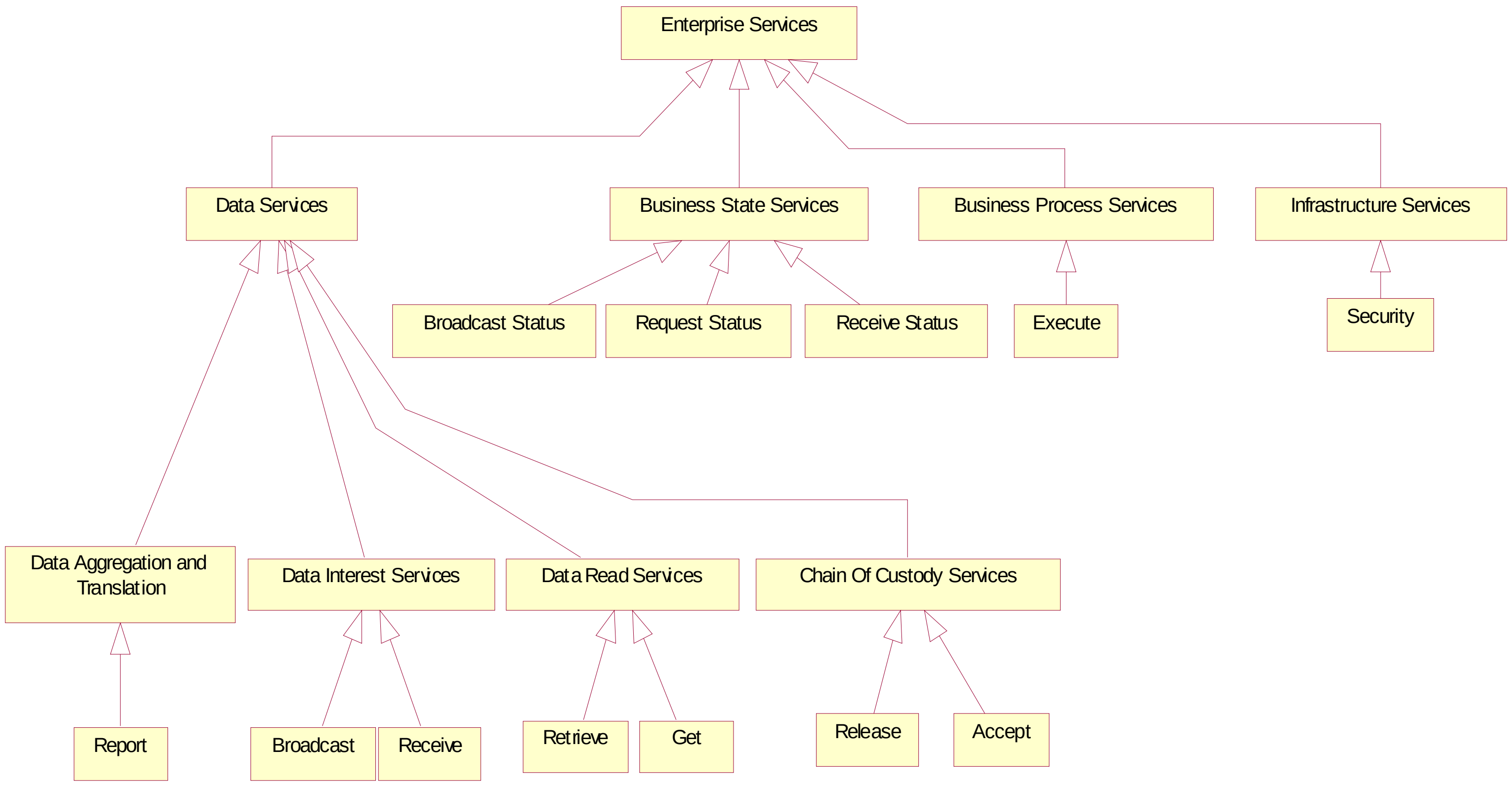


Figure 2: Service Classifications

These service classifications are briefly described in table 1 below.

Classification	**Description**
Broadcast	The *broadcast*service classification is a service that notifies the enterprise that the data in a business object has changed or a new entity has been created. This typically contains the changes only, but can contain the entire business object. This is equivalent to a “publish” in the publish/subscribe model.
Receive	The *receive* service classification is akin to a subscriber in the publish/subscribe paradigm. If a system has interest in changes to a business object when those changes are made, that system will implement a *receive* service for that piece of information.
Publish	The *publish* service classification is a service that is releasing custody of a business object. By releasing custody of an entity, the system is no longer considered the System of Record for that entity.
Accept	The *accept* service classification is the complement to the *publish* service classification. An *accept* service is accepting custody of a data entity.
Get	The *get* service classification describes services that retrieve data from the Operation Data Store (ODS). Data retrieved utilizing a *get* service is *near real time* but not guaranteed to the absolute latest. The ODS receives data changes as they are made by numerous systems by implementing *receive* services. By doing this, the ODS will be within seconds of having up to date information. This alleviates significant load on the transactional system.
Retrieve	The *retrieve* service classification mirrors exactly that of the *get* service classification. It is a request/reply service to return a collection of data entities. The difference is that *retrieve* services resolve to the transactional system. By resolving to the transactional system, systems utilizing the *retrieve* services are guaranteed to receive the most up to date information.
Execute	The *execute* service classification provides a process control entry points to systems participating in the enterprise. An *execute* service can be either synchronous or asynchronous depending on the nature of the control point.
Broadcast Status	The *broadcast status* service classification describes services that broadcast the status or state of interest to the enterprise. The status being communicated can be the state of a business significant item (day ahead market for 2/19/2004 is closed) or an error condition (day ahead market for 2/19/2004 failed to close).
Request Status	The *request status* service classification is the complements to the *broadcast status* classification. This service type allows the introspection of the state or status of business significant items (is the day ahead market closed for 2/19/2004?)
Receive Status	The *receive status* service classification is the complement to the *broadcast status* classification. This service type is implemented by a system interested in knowing when the status changes.
Report	It will become necessary to aggregate or translate data. The reasons to do this may be for consistency in translation or for performance. This may take the form of data marts maintained by the ODS, or simply an aggregation that occurs at the ODS level. The *report* service

Classification	Description
	classification is used to retrieve this aggregated or translated data in an efficient manner.

Table 1: Brief Description of Service Classifications

Execution mode is described as either synchronous or asynchronous. All services are described in Web Services Description Language (WSDL). WSDL defines a format for an XML document for describing Web services as a set of endpoints operating on messages. Service payloads, which are based on the semantic mode described later in this document, are automatically generated from UML diagrams by a product referred to as the MDI™ (Model-Driven Integration) Workbench. The service payload describes the expected types of data associated with the service classification. Each service can be implemented by one or more systems and utilized by one or more systems. The implementer of a service indicates that the body or content of that service is actually built by that system. A system that utilizes a service is making a request to that service.

The California ISO SOA is always implemented as Request-Response operation. This does not mean that services are always synchronous. For services that are defined as asynchronous, the service acknowledges the receipt of the request in its response, but performs the actual work for that service request independent of the service request.

The following table depicts how these services are related to one another. Note that data services have two key categories: read only business objects and business objects that have create, update and delete (CUD) operations performed on them.

Service Category	Description	Supporting Service Classifications
Business Process	Performing an actual business activity. Calculating Residual Unit Commitment or Proxy Bids, for example.	*Execute*
Business State	A business significant state. For example, is the Day Ahead Market for 1/25/2005 Open or Closed?	*Request Status* *Broadcast Status*
Business Objects that are Created, Updated or Deleted (CUD)	A system of record is an information storage system which is the data source for a given data element or piece information. By definition, a system cannot modify a business object unless it is the system of record. Being the system of record for a business object carries certain requirements.	*Broadcast* *Publish* *Accept* *Retrieve*
Business Objects that are Read Only	This speaks directly to system integration. If my system is not the System of Record for a business object, I must obtain the information from an authoritative	*Receive* *Retrieve* *Get*

	source.	
Reporting/Data Aggregation	Data aggregation or translation.	*Report*
Infrastructure Services	These are horizontal components expected to provide large grained services such as security, auditing, logging, correlation etc.	

Table 2: Related Service Classifications

Services Rules

To have a properly functioning whole system, each of these services must be used in accordance with rules of engagement. Such rules include:

- When a Entity can be modified with a system, then that system that is modifying the Entity is the system of record (SOR).
- Any time the SOR revises or otherwise changes an SOA Entity, the SOR must notify the enterprise of the change via a broadcast event.
- For every publish service, there will be a corresponding accept service. Publish and accept services work as a single transaction and therefore must be coupled.
- For every publish/accept transaction, there will be a corresponding broadcast service invoked by the integration platform.
- A state transformation on an entity has an associated significant business event (there must be a business reason for the state transition).
- When utilizing the broadcast/receive (publish/subscribe) paradigm for receiving data, it is the responsibility of the implementer to know how to recover from catastrophic failure or cold start by utilizing the corresponding get or retrieve service.
- Publish and broadcast events must handle fault of middle tier. The publish and broadcast implementations must be able to handle fault so that if they do not succeed, they continually retry until successful. This is to guarantee no loss of events in the event the integration platform is unavailable for a given length of time.

System of Record

A system of record is an information storage system which is the data source for a given data element or piece information. A service cannot modify a business object unless it is the system of record. Being the system of record for a business object carries the requirements:

- For every business object that a system is the system of record for, the corresponding broadcast service must be invoked any time that business object is changed. The enterprise must be notified of changes to business objects. There may be any number of systems interested in changes to said business object.
- For every business object that a system is system of record for, a corresponding retrieve service must be made available to the enterprise. Not all systems will choose to listen to changes for a business object only. The systems may need to

utilize a retrieve service to acquire the data. Retrieve services are also required for cold-start situations and contingency planning.

- In order to become the system of record, a system must implement a corresponding accept service.
- In order to no longer be the system of record for a business object, a corresponding publish service must be invoked.

The Method for Defining Services

To define those services, certain artifacts are needed, which are summarized in the following table.

Artifact	Description
BPM	Business Process Model. A diagram depicting the business processes and activity flows.
BPM Drill Down	Whereas a BPM describes a high-level business process, a BPM Drill Down goes in to more detail supporting a high-level business process.
Business Use Cases	Use cases are a technique for capturing the functional requirements of a system. Use cases work by describing the typical interactions between the users of a system and the system itself, providing a narrative of how a system is used. There are different levels of use cases including Business Use Cases and System Use Cases. A business use case discusses how a business responds to a customer or an event (Actor). As will be shown below for the NERC Functional Model, California ISO is able to leverage previously defined use cases maintained as part of the MDI Business Models. California ISO also contributes artifacts that it believes are useful to others in industry to this model clearinghouse, from where artifacts are made generic into industry best practices (informative) and some eventually become industry standards (normative).
Input and Output Description	A diagram and narrative that describes data that moves in and out of a system.
Fact Models	A diagram and narrative that define business terms.
Semantic Data Model	The semantic data model identifies business objects at the enterprise level. Internally to a system, the representation of data can be what ever is required. However, when the business object is presented to the enterprise via the California ISO SOA, that object must conform to an object found in the semantic data model. The Semantic Data Model is based on the CIM. It is managed and extended as necessary using Xtensible's MDI Framework.
Business Requirements	Describing the business needs to be solved by a system. Business Requirements consists of clear and succinct statements, narrative descriptions and diagrams. Use cases are key parts of business requirements.

Table 3: Key Inputs for Defining Services

Each category of services has a primary input artifact and additional contributing artifacts. How these artifacts are used is described below:

1. **Defining Business Process Services.** Business process models are diagrams that depict business processes and activity flows, which provides a good means for recognizing business significant events. However, the business process model may not be at the right granularity. For example, a business process model may have too much information and should therefore result in multiple services. Some steps in a business process model should not result in any services. Therefore, business use cases, activity diagrams and sequence diagrams are used to flesh this out. Note that every business process is a service.

2. **Defining Business States.** Business states are primarily discovered from the artifacts of the previous step. However, while there may be many states of a system, a business state of interest to the enterprise will typically be depicted on the highest-level business process model. Once the business states have been identified, SOA Services Rules and Implementation Patterns are used to determine how that business state is to be managed. For example:
   - Any time a business state changes, the enterprise must be notified by invoking the corresponding broadcast status service.
   - The system must surface a request status service for the state to the enterprise. This is so that any system can determine the status at any time, in the event it was not available when a broadcast status service was invoked.
   - Any system that is interested in being notified when the business state changes, must implement a receive status service.

3. **Discovering Create, Update and Delete Business Objects.** Identifying business objects that are changed in any way within a system indicates that the system must be the system of record for that object. Key use cases are those that perform create, update and delete on business objects. Once the CUD business objects are identified, they must be aligned with the semantic data model.

4. **Discovering Read Only Business Objects.** Identifying read only business objects is a fairly straightforward and is primarily achieved with Data Input and Output Descriptions. Once done, they are aligned with the semantic data model.

5. **Discovering Reporting and Data Aggregation.** There are many reasons why it is necessary to aggregate or translate data. This may take the form of data marts maintained by the ODS, or simply an aggregation that occurs at the ODS level. The report service classification is used to retrieve this aggregated or translated data in an efficient manner. Business use cases and system requirements are used to determine what the reports are needed, which in turn become services.

6. **Modeling the Services.** Once a comprehensive set of services are discovered, we model the services. The purpose of modeling the services is twofold. First, it will clearly communicate the service categories. Secondly, we need to provide

context for the business services defined. It is important to produce a UML sequence diagram showing how it is expected that the services will be executed. It is not necessary to include the read only data integration services in a sequence diagram. More than one sequence diagram may be required convey the full context of the business services

As a result of this process, necessary services are identified, the inputs and outputs of the services are defined, the business process they support are described, and whether they are utilized or implemented is articulated. If the service is implemented, then it is the requirement of the system being analyzed to expose those service so the enterprise. If the service is to be utilized, than that becomes a requirement for another system to expose that interface. The list of utilized services is correlated at the integration level. Sequence diagrams provide the business services context and convey the general order of their execution.

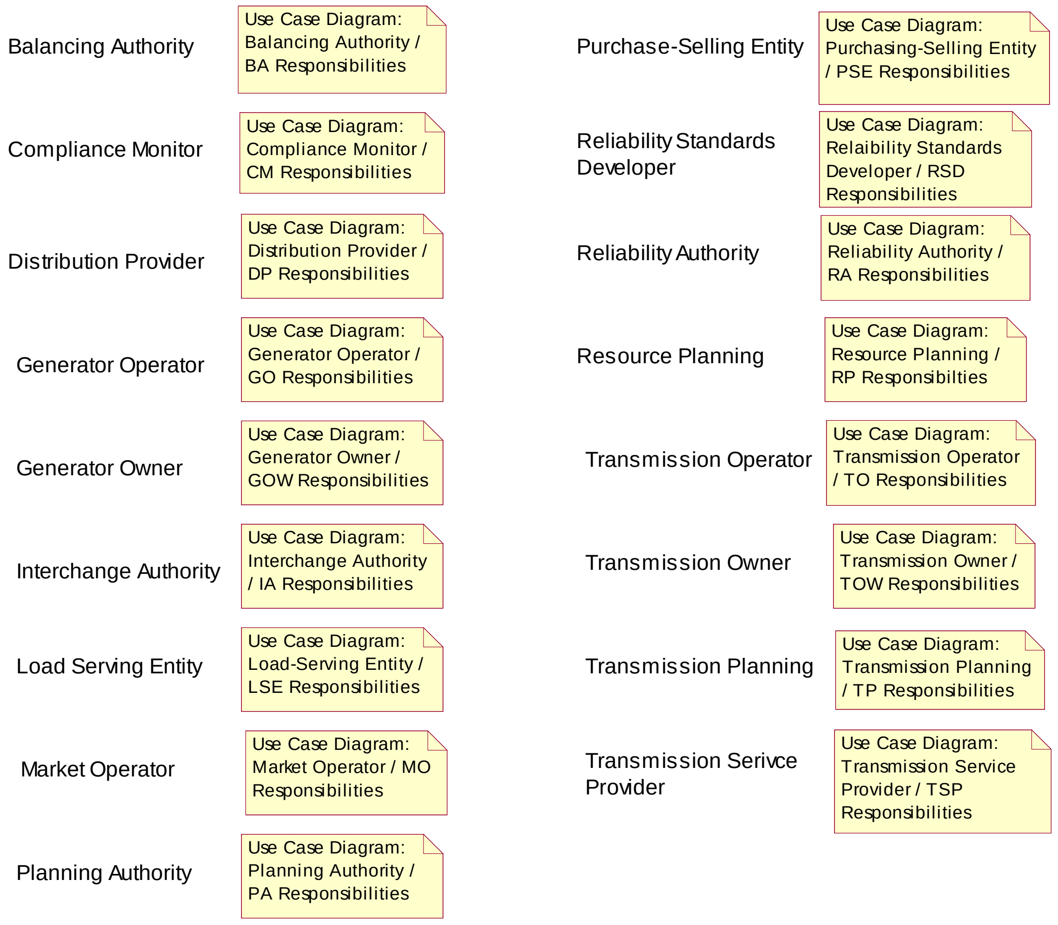
Linking Reliability Requirements into the SOA

California ISO of course wants to take reasonable measures that enhance the reliability of California's electric networks. Even as improved automation options become available to the many companies operating in the California market that are dealing with various aspects of planning and operating parts of the interconnected electric network, achieving overall reliability objectives has become much more difficult in recent years for a multitude of reasons, including:

- Evolving markets and market participants
- Changes in asset ownership, service, and operation
- Changing utility organizational structures
- Loss of corporate memory as people are shuffled as a consequence of inter- and intra-organizational changes
- Tight capital coupled with imbalanced risk/reward impacting investment planning

To achieve reliability among companies operating with such disparate circumstances, NERC has recently developed the NERC Functional Model (FM). It enables an individual company to better understand reliability obligations directly relevant to its business. It facilitates this by organizing reliability standards that do not assume particular organizational structures. Utility industry companies are generally able to describe which of their organizations and supporting systems are used to meet individual requirements. However, many find it difficult to systematically demonstrate how reliability requirements are met when affiliated business processes span multiple organizations and systems. So the remainder of this paper touches on how these reliability requirements applicable to California ISO can be supported in its SOA.

The FM defines functions involving reliability that are performed by various entities (e.g., Control Areas, Regional Transmission Organizations, Independent System Operators, Generator Owners, Distribution Providers, Purchasing-Selling Entities). Unfortunately, the FM is only a paper document, making it a challenge to tie its requirements into business process managed solutions. The MDI Business Models include a UML rendering of the NERC FM, that provides a means to trace requirements from the NERC FM to how they are implemented in the SOA. Referring to figure 3, observe that there are seventeen functions in the model. For each of these functions, tasks are defined that must be performed by an entity that claims responsibility for performing that function. An organization that registers with NERC for performing a given function has responsibility for ensuring that all tasks of that function are performed in accordance with associated Reliability Standards. [1]



Balancing Authority
Use Case Diagram: Balancing Authority / BA Responsibilities
Compliance Monitor
Use Case Diagram: Compliance Monitor / CM Responsibilities
Distribution Provider
Use Case Diagram: Distribution Provider / DP Responsibilities
Generator Operator
Use Case Diagram: Generator Operator / GO Responsibilities
Generator Owner
Use Case Diagram: Generator Owner / GOW Responsibilities
Interchange Authority
Use Case Diagram: Interchange Authority / IA Responsibilities
Load Serving Entity
Use Case Diagram: Load-Serving Entity / LSE Responsibilities
Market Operator
Use Case Diagram: Market Operator / MO Responsibilities
Planning Authority
Use Case Diagram: Planning Authority / PA Responsibilities
Purchase-Selling Entity
Use Case Diagram: Purchasing-Selling Entity / PSE Responsibilities
Reliability Standards Developer
Use Case Diagram: Relaibility Standards Developer / RSD Responsibilities
Reliability Authority
Use Case Diagram: Reliability Authority / RA Responsibilities
Resource Planning
Use Case Diagram: Resource Planning / RP Responsibilties
Transmission Operator
Use Case Diagram: Transmission Operator / TO Responsibilities
Transmission Owner
Use Case Diagram: Transmission Owner / TOW Responsibilities
Transmission Planning
Use Case Diagram: Transmission Planning / TP Responsibilities
Transmission Serivce Provider
Use Case Diagram: Transmission Service Provider / TSP Responsibilities

Figure 3: Major Functions of the NERC Functional Model

An example is shown in Figure 4, where use cases (represented as ovals) relevant to the business actor ‘Reliability Authority’ (represented as a stick figure) are summarized in a UML diagram. Color coding of use cases indicates the applicable section of NERC requirements for the modeled function: making deals (blue, n/a here), ahead of time (green), real time (red), after the hour (rose), or compliance (gray). According to NERC, the Reliability Authority “ensures the real-time operating reliability of the interconnected bulk electric transmission systems within a Reliability Authority Area.” [2]

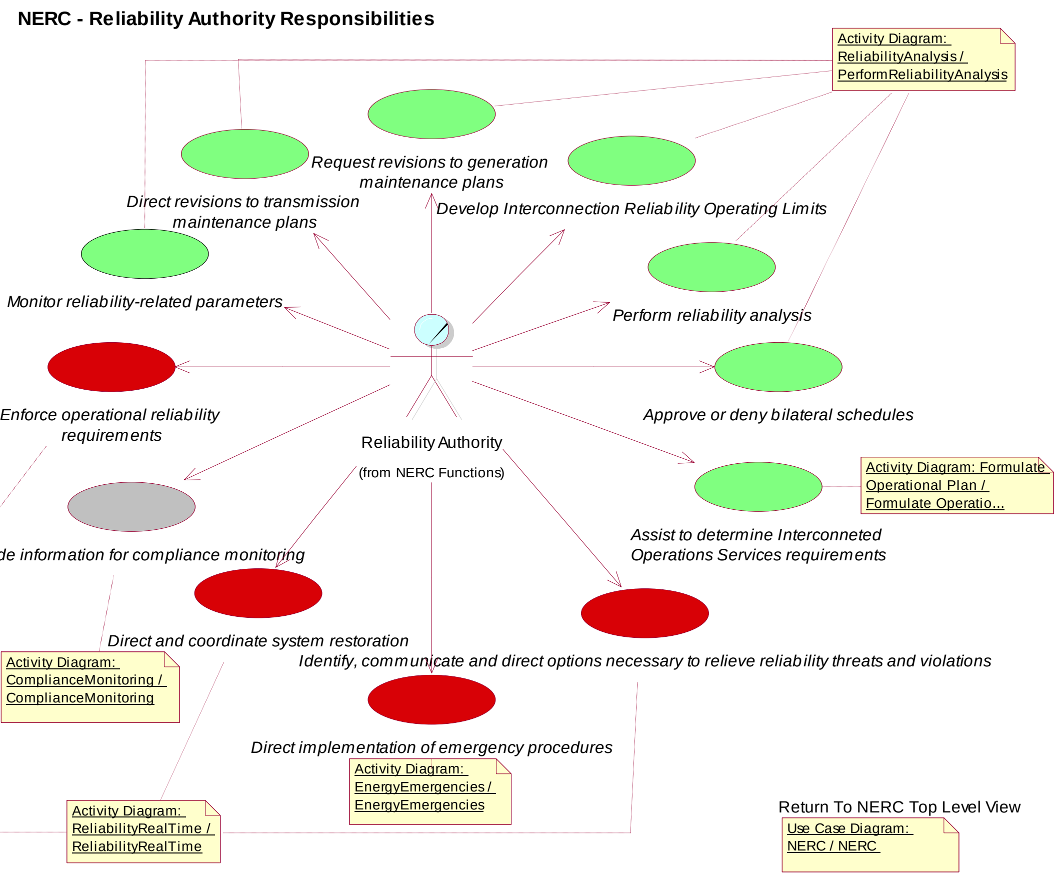

NERC - Reliability Authority Responsibilities
Activity Diagram: ReliabilityAnalysis / PerformReliabilityAnalysis
Request revisions to generation maintenance plans
Direct revisions to transmission maintenance plans
Develop Interconnection Reliability Operating Limits
Monitor reliability-related parameters
Perform reliability analysis
Enforce operational reliability requirements
Approve or deny bilateral schedules
Reliability Authority
(from NERC Functions)
Activity Diagram: Formulate Operational Plan / Formulate Operatio...
Provide information for compliance monitoring
Assist to determine Interconneted Operations Services requirements
Direct and coordinate system restoration
Identify, communicate and direct options necessary to relieve reliability threats and violations
Activity Diagram: ComplianceMonitoring / ComplianceMonitoring
Direct implementation of emergency procedures
Activity Diagram: EnergyEmergencies / EnergyEmergencies
Return To NERC Top Level View
Use Case Diagram: NERC / NERC
Activity Diagram: ReliabilityRealTime / ReliabilityRealTime

Figure 4: Reliability Authority Use Case Overview

Use cases such as these are used to organize application specific information flow and exchange requirements. They are supported by activity diagrams, which show the flow of activities for a given topic among relevant organizations. For example, a snippet of the "Perform Operational Analysis" activity diagram is shown in figure 5. The activities performed by an organization are shown in sequence within a swim lane. In this example, we see a few activities of the Reliability Authority that are exchanging information with activities being performed by the Balancing Authority and the Interchange Authority. Each rectangle in this diagram containing text that begins with "MT" is a message types reference. The reference is to a *message type* UML class diagram, part of the MDI Business Models, that consists of classes from the semantic model that are relevant to the topic covered by the message type. The MDI Framework provides the means to automatically convert these diagrams to XML schemas (service payloads) that are used to govern information exchange.

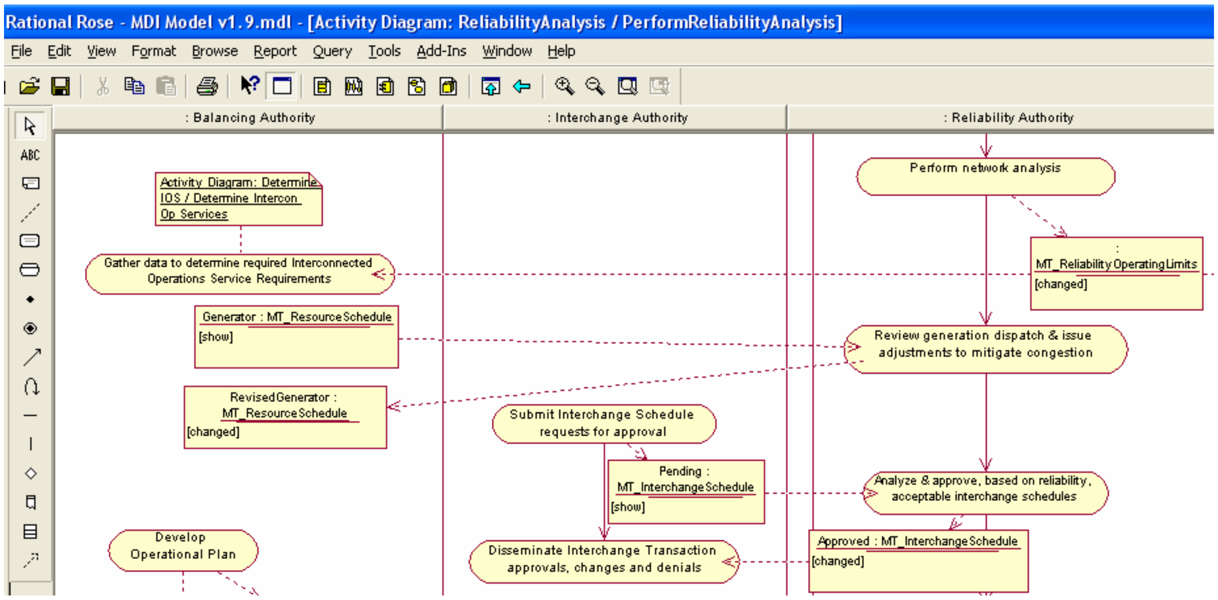

Rational Rose - MDI Model v1.9.mdl - [Activity Diagram: ReliabilityAnalysis / PerformReliabilityAnalysis]
File Edit View Format Browse Report Query Tools Add-Ins Window Help
: Balancing Authority
: Interchange Authority
: Reliability Authority
Activity Diagram: Determine IOS / Determine Intercon Op Services
Gather data to determine required Interconnected Operations Service Requirements
Generator : MT_ResourceSchedule [show]
RevisedGenerator : MT_ResourceSchedule [changed]
Develop Operational Plan
Submit Interchange Schedule requests for approval
Pending : MT_InterchangeSchedule [show]
Disseminate Interchange Transaction approvals, changes and denials
Perform network analysis
: MT_Reliability OperatingLimits [changed]
Review generation dispatch & issue adjustments to mitigate congestion
Analyze & approve, based on reliability, acceptable interchange schedules
Approved : MT_InterchangeSchedule [changed]

Figure 5: A Snippet From The Perform Operational Analysis Activity Diagram

California ISO, as well as market participants, can use these models as a way to understand how reliability requirements are being handled within their enterprise, especially in business processes that span multiple applications and departments. From an overall market perspective, there is an opportunity to leverage this same approach on business-to-business (B2B) transactions.

Conclusion

The migration from the current monolithic, tightly coupled systems to a Service Oriented Architecture (SOA) is resulting in faster integration of applications and information. It is improving California ISO's ability to react quickly to business changes while still providing the right information to people when they need it. The reusable methodology, business process models, common information models, and information exchange models – all based on appropriate standards - articulate how business objectives are implemented and provide end-to-end requirements traceability. The result is that market rules can be based on real system operating constraints that create economic incentives for maintaining reliability.

Acknowledgements

The authors thank the California ISO team, especially Ken Whitaker, whom have developed many of the ideas discussed in this paper.

References

[1] "NERC Reliability Functional Model, Function Definitions and Responsible Entities," NERC, Version 2 (Draft 12: October 23, 2003).

[2] Newberry, Ken and Greg Robinson, "The Semantic Integration Framework at TVA," GITA 2004, Conference Proceedings.

[3] Committee drafts of IEC 61968, IEC TC57 Working Group 14 (refer to http://www.wg14.com), 2004.