Telescope Time Allocation Tools
Conceptual Architecture

PREPARED BY	ORGANIZATION	DATE
Mark Whitehead	NRAO	3/19/2020

Change Record

VERSION	DATE	REASON
0.1	3/13/2020	Released initial draft.
0.2	3/19/2020	Incorporated internal pre-CoDR feedback.
0.3	4/20/2020	Incorporated CoDR feedback.

TABLE OF CONTENTS

1 Architecture Background 3
1.1 Problem Background 3
1.1.1 System Overview 3
1.1.2 Context 3
1.1.3 Driving Requirements 5
1.2 Solution Background 7
1.2.1 General Architecture Principles 7
1.2.2 Architecture Refinements 10
1.2.3 Anti-Corruption Layer 12
2 Views 14
2.1.1 System Context 15
2.1.2 Domain Layer 16
2.1.3 Application Layer 43
3 Referenced Materials 47

"The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both."

- "Software Architecture in Practice", Bass et al.

1 ARCHITECTURE BACKGROUND

1.1 Problem Background

The sub-parts of this section explain the constraints that exert significant influence over the architecture.

1.1.1 System Overview

[*This section describes the general function and purpose for the system or subsystem whose architecture is described in this document.*]

The conceptual architecture described in this document will be used to develop a new set of telescope time allocation (TTA) tools for a variety of NRAO proposing facilities and review processes.

The conceptual architecture describes the minimum number of concepts and their relationships needed to execute TTA processes, including: proposal solicitation specification, proposal preparation and submission, proposal review, time allocation, allocation approval, and time award. The architecture creates a contextual boundary around the core TTA concepts while providing an isolating layer to support the flow of proposal and award information into existing scheduling and observing systems. In addition, the solicitation, proposal, allocation, and award concept structures are intended to support the capture of information required to support science-ready data products.

The overall architectural style for this system relies on three patterns: Domain Model, Layers, and Domain Object. Domain Model creates and enforces a contextual boundary around core TTA concepts to support system sustainability over a decade or longer. Layers and Domain Object enforce separation of concerns between and within levels of abstraction, respectively, to support maintainability. The overall architectural style is refined via a contemporary version of Layers, called Hexagonal Architecture, which strictly isolates the Domain Model from technology used to implement other system features related to user interfaces, messaging, and persistence.

The system requirements stress flexibility and consequently the architecture includes structures that permit updates without code changes and accommodate future facilities, new types of proposals, and different kinds of proposal review processes.

1.1.2 Context

[*This section describes the goals and major contextual factors for the software architecture. The section includes a description of the role software architecture plays in the life cycle, the relationship to system engineering results and artifacts, and any other relevant factors.*]

The TTA system conceptual architecture adheres to DMS Architecture Standards[1]. These standards are compatible with the "Vee Model" for systems engineering and utilize conceptual,

[1] DMS Architecture Standards

logical, and physical architecture phases to maintain tight coupling between what the stakeholders want and what the developers build throughout the development process. This document only describes the conceptual architecture.

Figure 1 illustrates planned iterative phases as the design evolves from conceptual models to deployed code and emphasizes how architecture development is paired with other activities as development progresses.

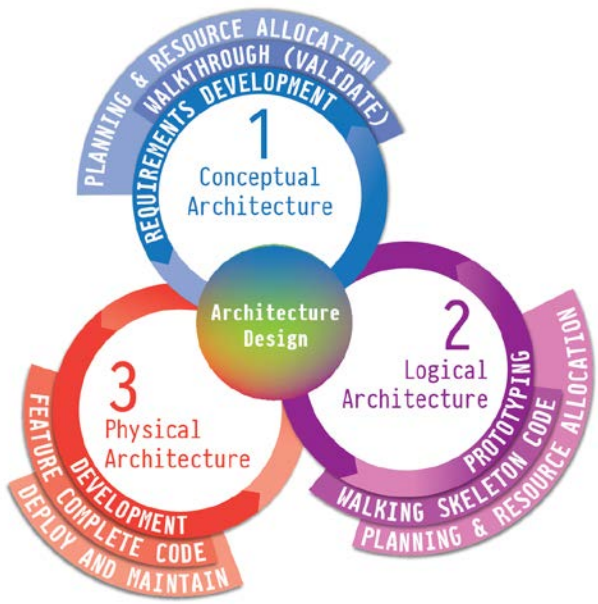


Figure 1 Planned iterative phases emphasizing how architecture development is paired with prototyping and coding as development progresses. Graphic by Reid Givens.

1.1.2.1 Conceptual Phase

This phase pairs requirements analysis and development with conceptual design; the goal is to analyze the requirements to produce an abstract model which highlights relationships and multiplicities between key concepts with no implementation details. The resulting conceptual architecture is a language that enables precise communication between stakeholders and developers and forms the basis of subsequent development and maintenance.

After all the requirements have been analyzed, the architect and stakeholder(s) "walk through" use cases to validate the conceptual architecture relative to the requirements. This paper exercise verifies there are no extraneous concepts and that the conceptual architecture contains structures that can be associated with all L0 and L1 requirements.

After the architect and stakeholder(s) agree the conceptual architecture is complete, there is an opportunity for a Conceptual Design Review and an initial round of planning and resource allocation.

1.1.2.2 Logical Phase

In this phase prototyping iterations are used to validate the conceptual architecture, capture dynamic behavior, and produce a simple end-to-end system (i.e. walking skeleton). The prototyping process exposes the parts of the conceptual design that need greater detail and the conceptual design is refined into a logical design. This phase does not identify particular technology choices unless it is advantageous to do so.

Also, this phase includes the development of unit tests for the prototype code and requires validation led by the stakeholder(s). Static code analysis is introduced in this phase to establish and maintain baseline code quality standards.

Once the architect, developer(s), and stakeholder(s) are satisfied with the walking skeleton, there is an opportunity for a Logical Design Review and another round of planning and resource allocation.

Phases 1-3 in the Telescope Time Allocation Tools Execution Plan define specific objectives for the Logical Phase[2].

1.1.2.3 Physical Phase

In this phase, development iterations elaborate the walking skeleton to incrementally include additional features. For each iteration, the logical architecture is refined into the physical architecture by including entities that point to real life software, servers, systems, etc. Software verification will be accomplished through automated system testing as part of continuous integration and deployment.

1.1.3 Driving Requirements

[*This section lists the functional requirements, quality attributes, and design constraints. It may point to a separate requirements document.*]

The conceptual architecture is largely derived from the TTA system concept[3] and TTA system description[4]. Presentations describing the overall concept from a user's perspective and the project kickoff also influenced the architecture. Analysis of this information resulted in the following quality attributes and constraints.

[2] "Telescope Time Allocation Tools Execution Plan", Treacy, Kern, 688-TTAT-010-MGMT, Version 0.01
[3] "Telescope Time Allocation (TTA): Concept", Balser et al., 688-TTAT-002-MGMT, Jul. 02, 2019
[4] "Telescope Time Allocation (TTA): System Description", Balser et al., 688-TTAT-004-MGMT, Mar. 13, 2020

1.1.3.1 Quality Attributes

1.1.3.1.1 Sustainability

NRAO wants to utilize this system for a decade or more. Therefore, the system must be based on architectural features that permit cost effective change to requirements, environments, and configurations.

1.1.3.1.2 Maintainability

The architecture must support variations in feature sets, organizational processes, and algorithmic behavior. The design must especially include architectural features that “Isolate the user interface from other parts of the system as requirements for this area are likely to have the most ‘churn’”[5].

1.1.3.1.3 Performance and Reliability

The majority of the processing for this system involves responding to client requests. For this type of processing, the architecture must support specific system performance requirements[6]:

The System have the following performance metrics which occur at peak times during the day of the proposal deadline. Here we quote values for the PST during the 20A semesters.
(a) Server load shall be less than 3 - 4. The stress zone is a load near 7 - 8.
(b) Server shall be able to handle 140 simultaneous users.
(c) Server shall be able to handle 60 proposals submitted within a two-hour period.
(d) Server shall be able to handle 10,000 pages served over a two-hour period.

According to ISO/IEC FCD 25010, the Performance attribute is related to “performance relative to the amount of resources used under stated conditions” and the Reliability attribute is related to “the degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.” Quality Attribute Scenarios (or equivalent) should be used to refine the system performance requirements so that the appropriate quality attribute tactics can be applied. Related metrics should be established in the logical phase and monitored through the physical phase and deployment.

Additional processing requirements involve exchanging data with other systems. It should take between 1-600s to transfer TTA information to any facility-specific system.

1.1.3.1.4 Configurability

There are numerous references throughout the System Description related to configuring the system without editing code. Appropriate design features must be chosen to support these requirements.

---

[5] “2019-06-Project KickOff”, Kern

[6] See 2.3.4 in 688-TTAT-04-MGMT, System Description

1.1.3.1.5 Usability

The system will primarily interact with different types of human users. System requirements for the User Interface will be developed in a subsequent architectural phase. Consequently, the conceptual architecture does not directly address the details of usability.

1.1.3.2 Constraints

Table 1 indicates key constraints for the design. Note that it is not required that all constraints be addressed in the conceptual architecture phase and that constraints 2-4 express desires that are not hard requirements.

ID	Constraint
CON-1	The TTA system will be a web-based tool. (TTA-L0-1.2)
CON-2	To the extent that it is efficient to do so, the implementation is expected to draw from the ALMA tools as well. (TTA-L0-1.1)
CON-3	The user interface will follow the design and functionality of the ALMA OT. (TTA-L0-1.3)
CON-4	If possible, proposal submission via the TTA system should be similar for any NRAO instruments. (TTA-L0-1.4)

Table 1 TTA Constraints

1.2 Solution Background

[The sub-parts of this section provide a description of why the architecture is the way that it is, and a convincing argument that the architecture is the right one to satisfy the behavioral and quality attribute goals levied upon it.]

CON-1 suggests a layered architecture. The desire for maintainability suggests use of the Layers and Domain Object patterns to enforce separation of concerns between and within, respectively, layers of abstraction. Due to NRAO's need to sustain TTA Tools for a decade or more, we further refine Layers by selecting a Hexagonal Architecture style to establish a core Domain Model that is strictly isolated from the rest of the application and from technology choices needed to meet overall system requirements.

1.2.1 General Architecture Principles

[*This section provides a rationale for the major design decisions embodied by the software architecture. It describes any design approaches applied to the software architecture, including the use of architectural styles or design patterns, when the scope of those approaches transcends any single architectural view. The section also provides a rationale for the selection of those approaches. It also describes any significant alternatives that were seriously considered and why they were ultimately rejected. The section describes any relevant COTS issues, including any associated trade studies.*]

1.2.1.1 Domain Model

According to Buschmann et al.[7], the Domain Model pattern:

"...defines a precise model for the structure and workflow of an application domain - including their variations. Model elements are abstractions meaningful in their domain; their roles and interactions reflect domain workflow and map to system requirements."

In consonance with sustainability and given the natural turnover of staff, it is vital to leverage architectural features that permit all stakeholders to use a precise language throughout the life of the system. A precise language facilitates reasoning about the system and cost effectively accommodating new requirements. The TTA Domain Model creates a contextual boundary around a highly unified software core representing key TTA concepts.

Domain-Driven Design[8] (DDD) was used to create the TTA Domain Model. DDD defines a minimum set of design primitives that can be readily modeled in standard UML and SysML. These primitives will be refined in the logical and physical architecture phases.

The design primitives are defined as follows[9]:

- Entity - Something with identity and continuity, tracked through different states, time, life cycle, etc.
- Value Object - An attribute that describes the state of something else; can be an assemblage of other objects or reference entities.
- Aggregate - A cluster of associated objects treated as a unit for the purpose of data changes. Aggregates have a root and a boundary. The boundary defines what is inside the aggregate. The root is a single, specific entity contained in the aggregate. The root is the only member of the aggregate that outside objects are allocated to hold references to, although objects within the boundary may hold references to each other.
- Repository - Represents all objects of a certain type as a conceptual set; a collection with more elaborate querying capability.
- Factory - Creates and reconstitutes complex objects and aggregates, keeping their internal structure encapsulated.
- Service - An aspect of the domain expressed as action, activity, or operation rather than object; something done for a client on request. A Service has no state of its own nor any meaning in the domain beyond the operation it hosts. A Service should have a defined responsibility and that responsibility and the interface fulfilling it should be defined as part of the Domain Model (i.e. parameters and results should be Domain Model domain objects and operation names should come from the language defined in the Domain Model).

---

[7] "Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing", Vol 4, Buschmann et al., 2010

[8] "Domain-Driven Design: Tackling Complexity in the Heart of Software", Evans, Eric, 2003

[9] Ibid.

1.2.1.2 Layers

According to Buschmann et al.[10], the Layers pattern:

"...helps to structure applications that can be decomposed into groups of subtasks in which each group of subtasks is at a particular level of abstraction, granularity, hardware-distance, or other partitioning criteria."

By enforcing separation of concerns between levels of abstraction, the Layers pattern supports maintainability.

1.2.1.3 Domain Object

Buschmann et al.[11] define Domain Object as a pattern that:

"...separates different functional responsibilities within an application such that each functionality is well encapsulated and can evolve independently".

Relative to the Layers pattern, the conceptual architecture uses the Domain Object pattern to enforce separation of concerns *within* levels of abstraction and therefore also supports maintainability.

1.2.2 Architecture Refinements

1.2.2.1 Hexagonal Architecture

Cockburn, Fowler, Freeman et al. document a contemporary interpretation of the Layers pattern called Hexagonal Architecture, originally known as "Ports and Adapters". This interpretation results in an architecture in which…

"...the code for the business domain is isolated from its dependencies on technical infrastructure, such as databases and user interfaces. We don't want technical concepts to leak into the application model, so we write interfaces to describe its relationships with the outside world in its terminology (Cockburn's ports). Then we write bridges between the application core and each technical domain (Cockburn's adapters)."[12]

---

[10] "Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing", Vol 4, Buschmann et al., 2010

[11] "Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing", Vol 4, Buschmann et al., 2010

[12] "Growing Object-Oriented Software, Guided by Tests", Freeman et al, 2009

Figure 2 provides a high-level graphic representation of hexagonal architecture.

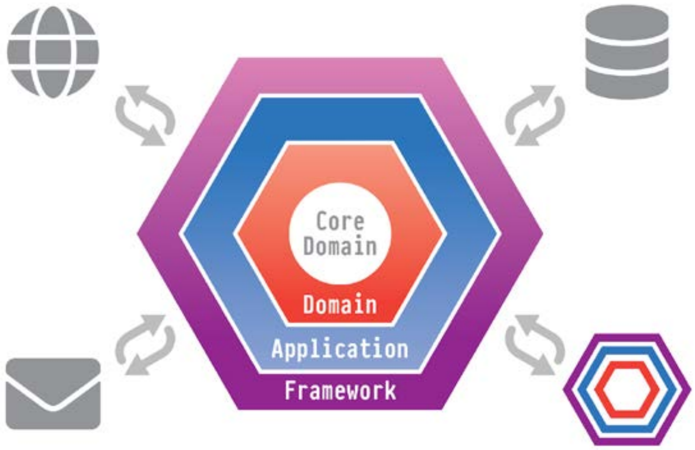


Figure 2 TTA Hexagonal Architecture emphasizing layers isolating the core domain from technology choices associated with user interfaces, messaging, persistence, and other systems. Graphic by Reid Givens.

The ports and adapters feature of this architecture will be further refined in subsequent phases using the Dependency Inversion Principle in the usual way[13]. Figure 3 shows how interaction between various technologies and the Application Layer can be implemented via abstract interfaces.

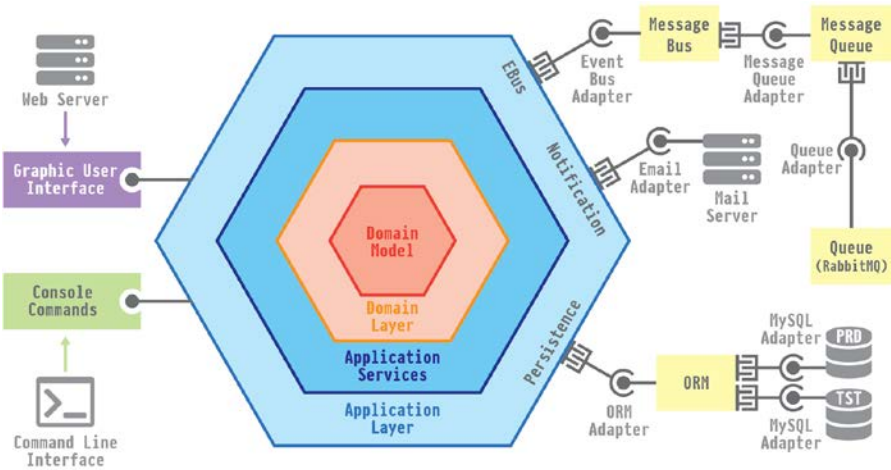


Figure 3 A detailed view of the interaction between the Framework Layer and Application Layer via interfaces. Graphic by Reid Givens.

The TTA conceptual architecture utilizes hexagonal architecture to address usability, maintainability, and configurability quality attributes. This design decision refines the general Layers pattern into the following specific TTA layer definitions.

[13] See SOLID Design Principles for details.

1.2.2.2 Domain Layer

The Domain Layer contains entities (Domain Objects) comprising a Domain Model derived from the TTA domain and expressed as Domain-Driven Design primitives. The Domain Objects represent 'business logic' - the rules the application must follow - and define how the Application Layer can interact with them.

Additionally, the Domain Layer can contain supporting domain logic such as Domain Events (events fired at important points in the business logic) and use-cases (definitions of what actions can be taken on the application).

1.2.2.3 Application Layer

Entities in the Application Layer orchestrate the use of entities found in the Domain Layer. The Application Layer also adapts requests from the Framework Layer to the Domain Layer.

1.2.2.4 Framework Layer

The Framework Layer includes entities that are not part of the Domain Model but are needed to satisfy system requirements. Specific Framework Layer entity examples include UX, persistence, messaging, job processing, or other systems.

1.2.3 Anti-Corruption Layer

Figure 2 includes the concept for systems interacting with one another via framework layers. Inter-system data transfer must be addressed because the TTA system must "…support the creation of observing projects for each allocation request with positive disposition in a format appropriate for each facility."[14]

Over the course of many years, different radio astronomy facilities have developed their own unique conceptual models for creating and executing projects. There are six patterns covering a range of strategies for relating different conceptual models[15]. The TTA system will use the Anti-corruption Layer (ACL) strategy to create an isolating layer to provide other systems with information or functionality in terms of their own domain model (see Figure 4). This strategy allows TTA to maintain a highly unified core conceptual model while supporting any facility which may have different development teams, budgets, requirements, etc. ACL will be instantiated in the TTA Framework layer and will consist of some combination of services, translators, adaptors, or facades. Additional design choices will be made in the Logical and Physical phases.

---

[14] "2019-03-TTA Tools Concept", Kern et al.
[15] See this analysis for details.

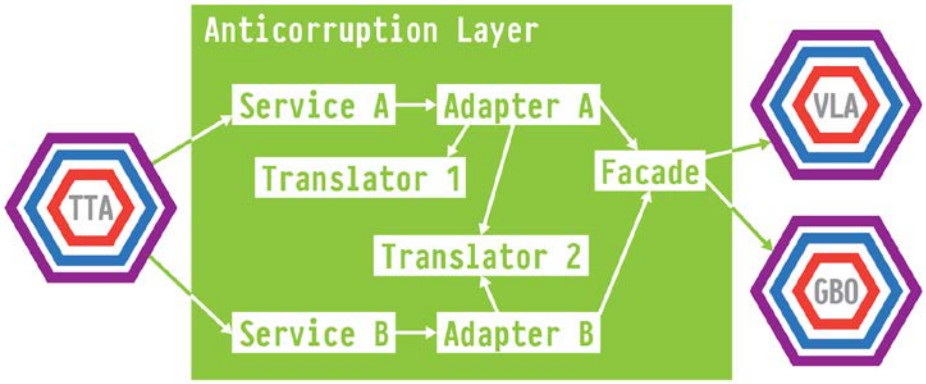


Figure 4 Depiction of the ACL strategy used to provide VLA and GBO with project information in terms of their down domain models. Graphic by Reid Givens.

2 VIEWS

The TTA system conceptual architecture was modeled in Cameo System Modeler. Figure 5 shows the model package structure. This section provides views of the System Context, Domain Layer, and Application Layer packages. It is expected that the Application and Framework layers will be refined substantially in the logical and physical architecture phases.

Domain Layer package views consist of a primary presentation, an element catalog, use cases, and requirements mapping. The primary presentation is a SysML Block Definition Diagram (BDD) and the element catalog defines each of the blocks in the BDD. The uses cases are standard UML/SysML. The requirements mappings are dependency structure matrices showing how the blocks in each view map to requirements.

Application Layer package views consist of only a primary presentation.

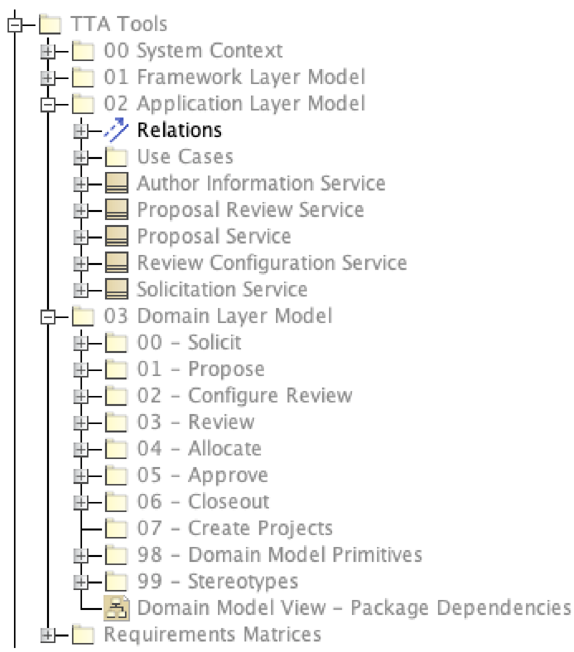


Figure 5 TTA system Cameo package structure

2.1.1 System Context

The System Context defines the users and other external entities that interact with the system. This view is used to define the environment that needs to be considered, define the system boundary, and identify required interfaces.

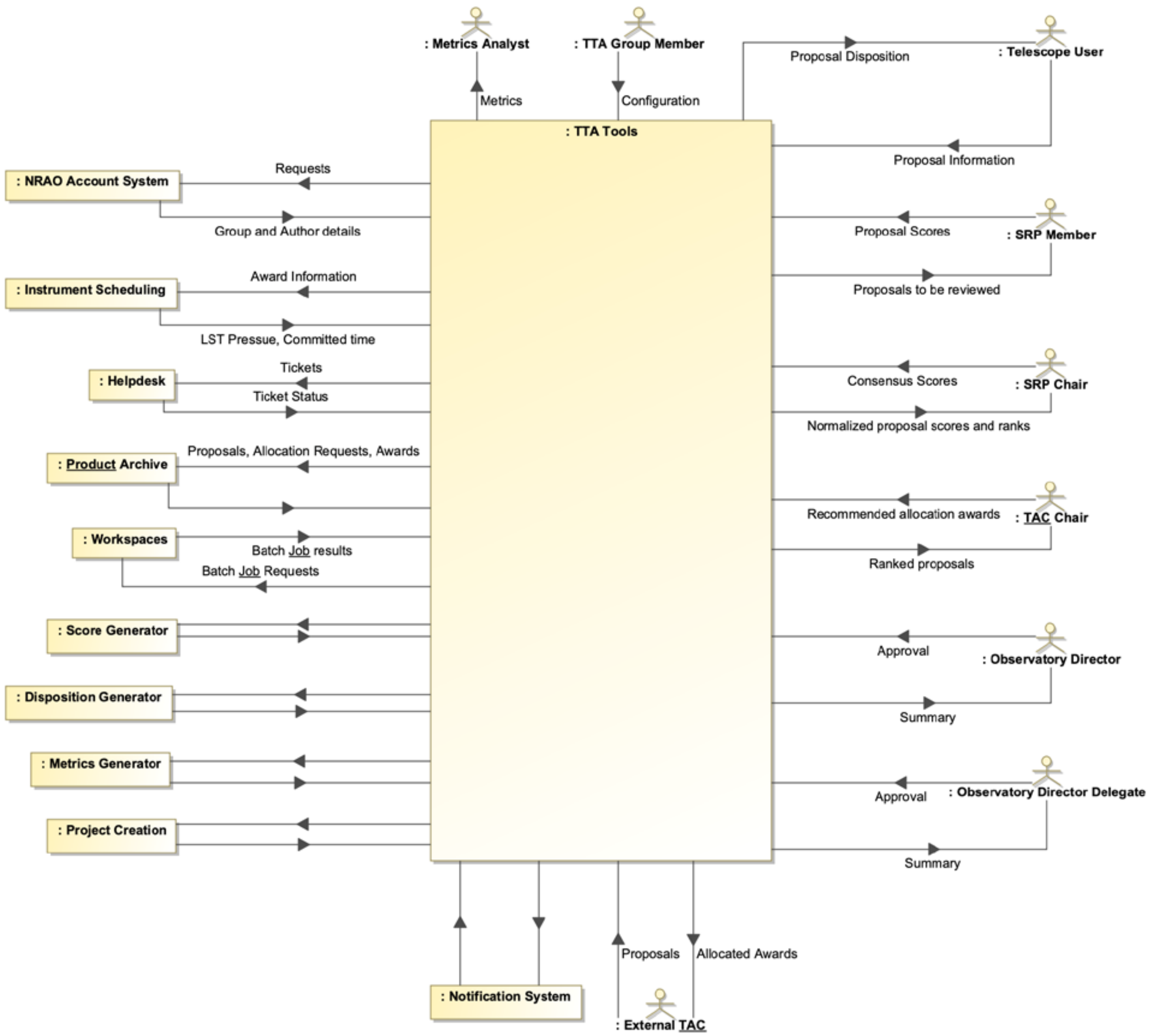


Figure 6 TTA System Context. Note that the Score, Disposition, and Metric Generators are place holders for entities that are likely part of TTA Tools but will be refined in a subsequent phase.

Table 2 provides short definitions for each of the actors shown in Figure 6. Based on requirements analysis, it is known that TTA Tools requires entities that generate scores, dispositions, and metrics. However, it is not yet clear where and how they fit into the architecture. Therefore, the entities are modeled as external systems and will be refined in subsequent phases. Design decisions about the Notification System have also been delayed in order to take advantage of a similar system that is currently being developed for a different SSA project.

Name	Description
Metrics Analyst	Compiles usage statistics for reporting
TTA Group Member	Supports and executes proposal and time allocation process
Telescope User	PI or Co-I, creates proposals
SRP Member	Provides scientific reviews of proposals
SRP Chair	Lead generation of consensus scores
TAC Chair	Recommends Time Allocation
Observatory Director	Approves allocation awards
Observatory Director Delegate	Approves allocation awards
External TAC	Provide external projects that have been allocated time
NRAO Account System	Provides authorization and authentication
Instrument Scheduling	Responsible for scheduling observations
Helpdesk	Provides telescope user issue management
Product Archive	Persistence layer products and supports data delivery
Workspaces	Provides job processing
Notification System	Manages sending notifications to different group members
Score Generator	Placeholder, will be refined in subsequent phase
Disposition Generator	Placeholder, will be refined in subsequent phase
Metrics Generator	Placeholder, will be refined in subsequent phase
Project Creation	Supports creation of observing projects for each allocation request with positive disposition in a format appropriate for each facility

Table 2 System Context Actor Information

2.1.2 Domain Layer

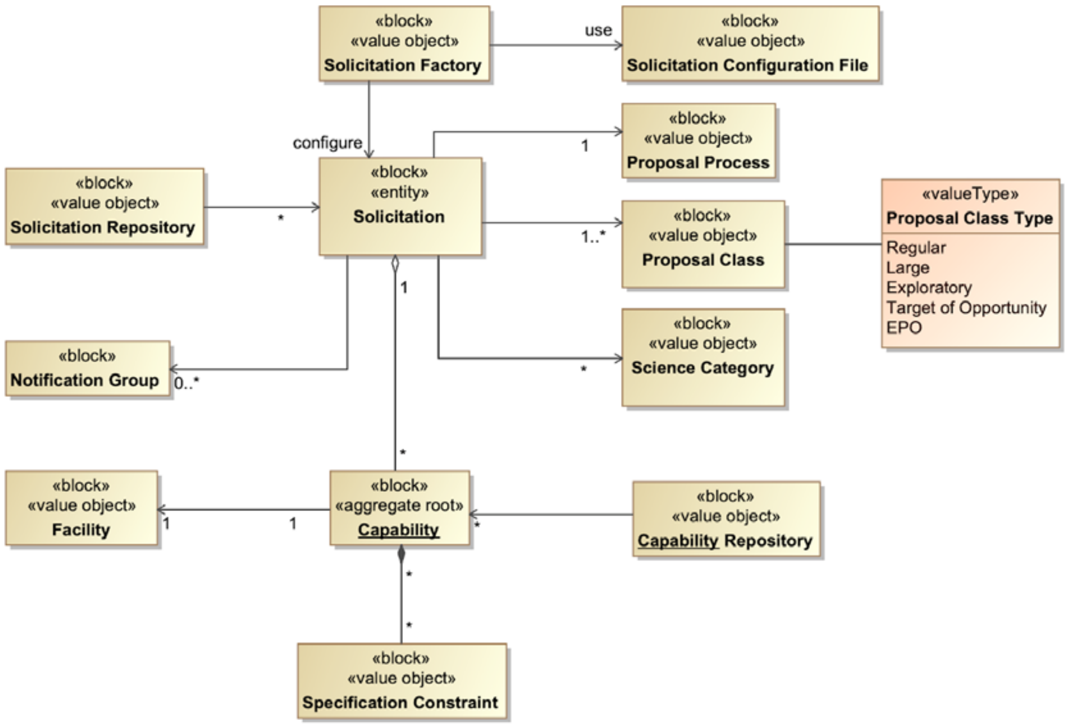
The Domain Layer Model consists of a set of packages which map to sections 3.1 to 3.9 in “TTA Tools System Description”. Each package contains models of the core concepts associated with each section. The following package views show the concepts along with their associations and multiplicities. A summary view showing the dependencies between packages is also provided.

In the following views, entity, value object, and aggregate design primitives are expressed as SysML stereotypes while repository, factory, and service primitives are expressed as block names.

2.1.2.1 Solicit

Telescope users submit proposals to access AUI NA telescopes in the context of solicitations. Solicitations define the resources available to proposers and the time period over which approved proposals execute. The Solicit package contains all of the concepts associated with solicitations. Support for multiple concurrent solicitations is a key feature of this conceptual architecture.

2.1.2.1.1 Primary Presentation

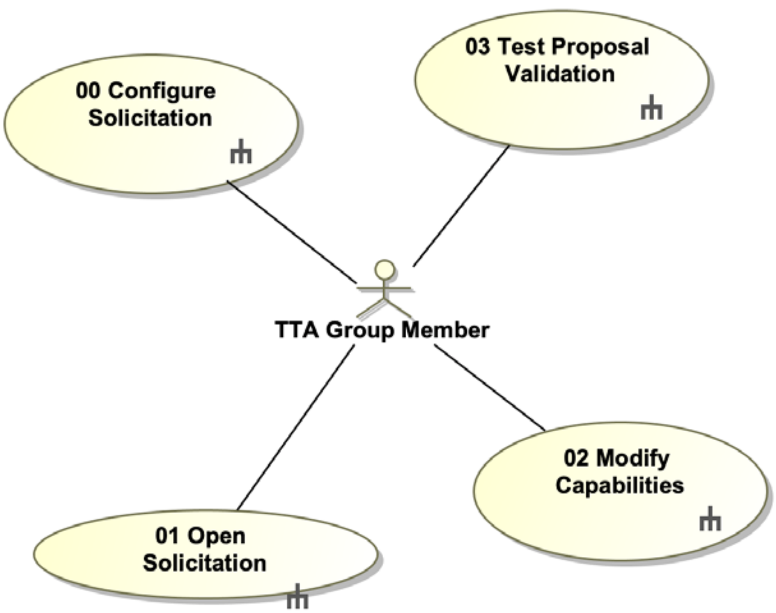


2.1.2.1.2 Element Catalog

Domain Object	**Definition**
Facility	One or more antennas that coordinate to perform observations. For example, the VLA consists of 27 antennas but is typically one Facility. The HSA may consist of all 10 VLBA antennas and all 27 VLA antennas but is considered as one Facility since the signals from all telescopes are correlated together. A Facility may also be a computing cluster to reprocess data.
Proposal Process	How a proposal is processed through the system.
Proposal Class	A designation providing a set of different validation rules within a Solicitation. For example, Regular versus Large proposals.
Specification Constraint	Restrictions on available resources within a Capability for a Solicitation.
Capability	The different ways a Facility may be operated and the resources available.

Solicitation	An announcement from the observatory to the community to submit a request to use observatory resources. Each solicitation is composed of Capabilities and a Proposal Process.
Science Category	The astronomical sub-field of science related to a Proposal.

2.1.2.1.3 Use Cases



2.1.2.1.4 Requirements Mapping

Legend
- ↗ Satisfy
- ⇢ Satisfy (Implied)

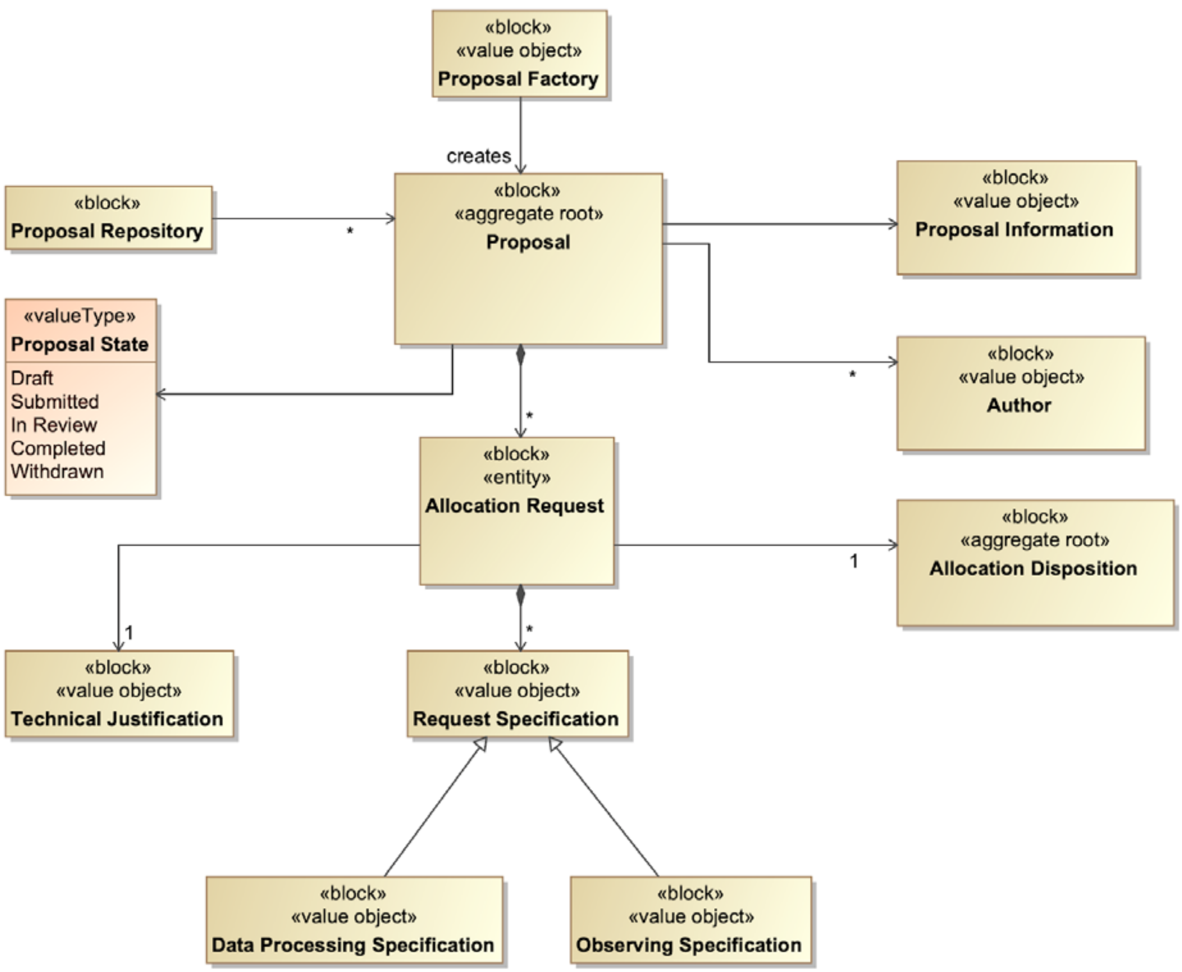
TTA-L1-1 Proposal Solicitation

	TTA-L1-1.2 Support Multiple Solicitations	TTA-L1-1.3 Solicitation Configuration	TTA-L1-1.4 Configure Solicitation	TTA-L1-1.4.6 Facility	TTA-L1-1.4.7 Capabilities	TTA-L1-1.4.7.1 Capability Configuration	TTA-L1-1.4.8 Validate Capabilities	TTA-L1-1.5 Notification Groups	TTA-L1-1.6 External Solicitation	TTA-L1-1.7 Solicitation Capabilities	TTA-L1-1.8 Solicitation Proposal Process	TTA-L1-1.9 Solicitation Facilities	TTA-L1-1.11 Semester Proposal Classes	TTA-L1-1.12 Facility-Capability Mapping	TTA-L1-1.14 Specification Constraints	TTA-L1-1.15 Open Solicitation	TTA-L1-1.16 Unique Identifier	TTA-L1-1.17 Execution Period	TTA-L1-1.18 Call Period	TTA-L1-1.19 Science Categories	TTA-L1-1.20 Semester Proposal Solicitation	TTA-L1-1.21 DDT Proposal Solicitation	TTA-L1-1.22 Sponsored Proposal Solicitation	TTA-L1-1.23 Resident Shared Risk Observing	TTA-L1-1.24 Demo Proposal Solicitation	TTA-L1-1.25 DDT Proposal Classes	TTA-L1-1.26 Demo Capabilities	TTA-L1-1.27 Demo Configuration	TTA-L1-1.28 Demo Notifications	TTA-L1-1.29 Demo Closeout
Capability		↗ ⇢			↗	↗	↗			↗ ⇢				↗	⇢						↗	↗	↗	↗			↗			
Capability Repository		↗				↗	↗			↗											↗	↗	↗	↗			↗			
Facility		↗		↗								↗		↗															↗	
Notification Group								↗																						
Proposal Class		↗											↗													↗				
Proposal Process		↗									↗																			
Proposal Service							↗																							↗
Proposal Service Interface																														
Science Category		↗																		↗										
Solicitation	↗	↗		↗		↗			↗		↗	↗	↗			↗	↗	↗	↗		↗	↗	↗	↗	↗					
Solicitation Configuration File		↗	↗	↗		↗			↗												↗	↗	↗	↗	↗	↗		↗		
Solicitation Factory	↗	↗							↗																					
Solicitation Repository	↗	↗							↗																					
Solicitation Service	⇢	⇢					⇢		↗		⇢	⇢				⇢					⇢	⇢	⇢	⇢	⇢					
Solicitation Service Interface	↗	↗					↗				↗	↗				↗					↗	↗	↗	↗	↗					
Specification Constraint		↗								↗					↗															

2.1.2.2 Propose

Telescope users create proposals describing how and why they want to use facility resources. The Propose package contains all the concepts associated with proposals. The Request Specification concept provides a flex point in the design to support requests for resources other than observing time. For example, as data processing becomes a more important factor in the evaluation process, the Request Specification concept can be extended to accommodate requests for computing resources, bandwidth, storage, etc.

2.1.2.2.1 Primary Presentation

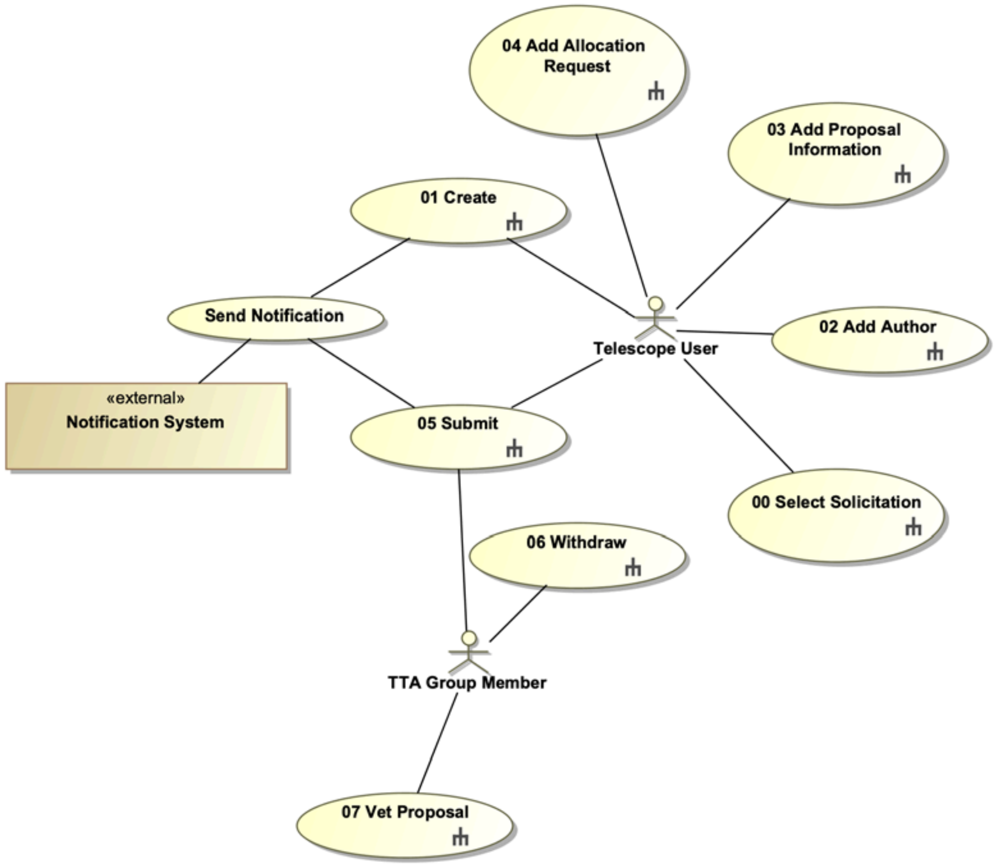


Note: Since all proposals with positive allocation dispositions will result in observing projects, there must be an association between Proposals, Allocation Dispositions, and Projects. This association will be modeled when the Create Project package is refined.

2.1.2.2.2 Element Catalog

Domain Object	Definition
Author	Any person on a proposal.
Proposal	A request to use observatory resources that includes a scientific and technical justification.
Proposal Information	The part of a Proposal that includes identifying information and the scientific justification. This information is independent of the resources being requested.
Allocation Request	The part of a Proposal that specifies the details of the requested observatory resources.
Request Specification	Specifies the resources that are being requested in the Allocation Request.
Technical Justification	A description of an observing process and considerations used to create an Allocation Request.
Allocation Disposition	The disposition of a given Allocation Request. Includes results of any evaluation process, scheduling constraints, and proprietary information.

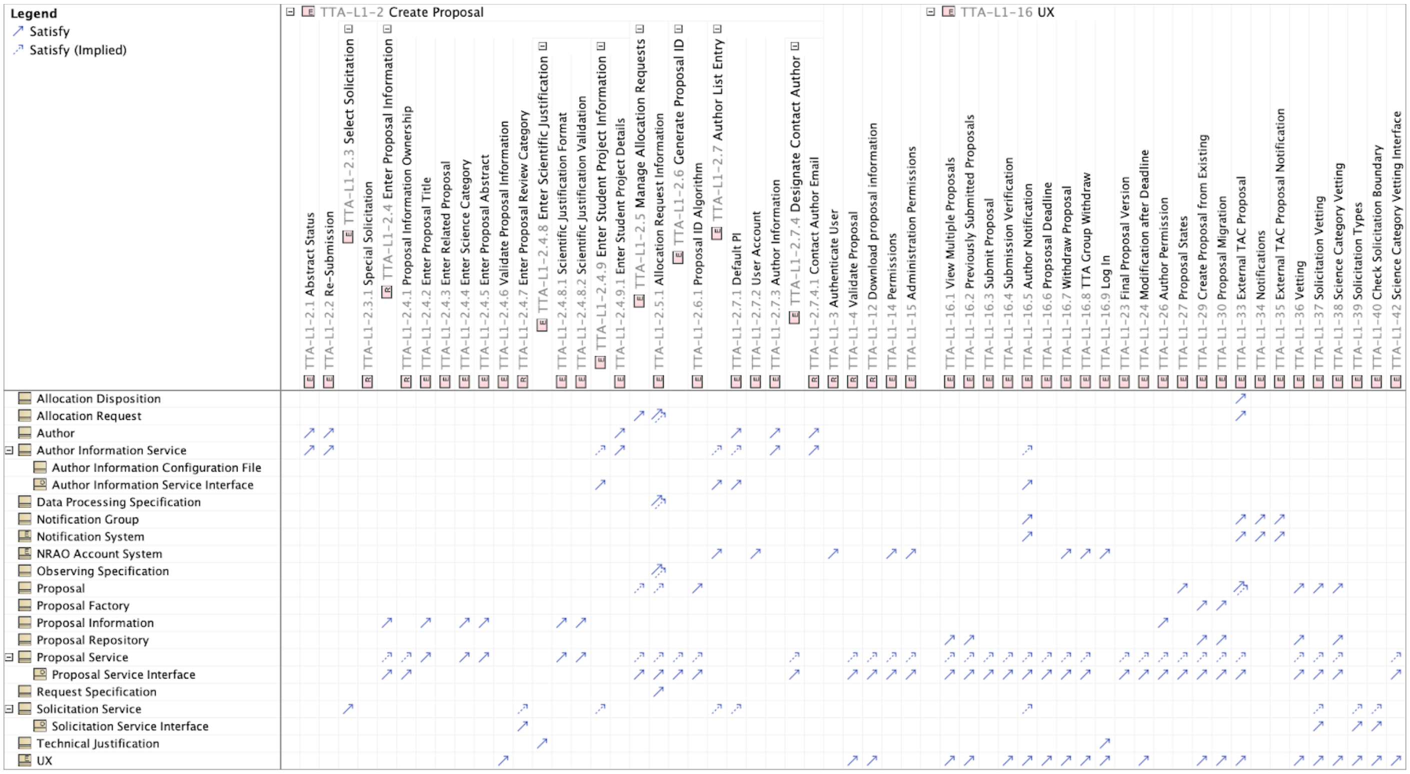
2.1.2.2.3 Use Cases



2.1.2.2.4 Requirements Mapping

Since there is currently no definition for Related Proposals, TTA-L1-2.4.3 will be addressed in the Logical Phase.

Note that TTA-L1-2.3.1 relates to proposals which are submitted for a “special” solicitation; reviews for these solicitations are handled outside of the TTA Tools are therefore out of scope.



Legend
Satisfy
Satisfy (Implied)
TTA-L1-2 Create Proposal
TTA-L1-2.1 Abstract Status
TTA-L1-2.2 Re-Submission
TTA-L1-2.3 Select Solicitation
TTA-L1-2.3.1 Special Solicitation
TTA-L1-2.4 Enter Proposal Information
TTA-L1-2.4.1 Proposal Information Ownership
TTA-L1-2.4.2 Enter Proposal Title
TTA-L1-2.4.3 Enter Related Proposal
TTA-L1-2.4.4 Enter Science Category
TTA-L1-2.4.5 Enter Proposal Abstract
TTA-L1-2.4.6 Validate Proposal Information
TTA-L1-2.4.7 Enter Proposal Review Category
TTA-L1-2.4.8 Enter Scientific Justification
TTA-L1-2.4.8.1 Scientific Justification Format
TTA-L1-2.4.8.2 Scientific Justification Validation
TTA-L1-2.4.9 Enter Student Project Information
TTA-L1-2.4.9.1 Enter Student Project Details
TTA-L1-2.5 Manage Allocation Requests
TTA-L1-2.5.1 Allocation Request Information
TTA-L1-2.6 Generate Proposal ID
TTA-L1-2.6.1 Proposal ID Algorithm
TTA-L1-2.7 Author List Entry
TTA-L1-2.7.1 Default PI
TTA-L1-2.7.2 User Account
TTA-L1-2.7.3 Author Information
TTA-L1-2.7.4 Designate Contact Author
TTA-L1-2.7.4.1 Contact Author Email
TTA-L1-3 Authenticate User
TTA-L1-4 Validate Proposal
TTA-L1-12 Download proposal information
TTA-L1-14 Permissions
TTA-L1-15 Administration Permissions
TTA-L1-16 UX
TTA-L1-16.1 View Multiple Proposals
TTA-L1-16.2 Previously Submitted Proposals
TTA-L1-16.3 Submit Proposal
TTA-L1-16.4 Submission Verification
TTA-L1-16.5 Author Notification
TTA-L1-16.6 Propsosal Deadline
TTA-L1-16.7 Withdraw Proposal
TTA-L1-16.8 TTA Group Withdraw
TTA-L1-16.9 Log In
TTA-L1-23 Final Proposal Version
TTA-L1-24 Modification after Deadline
TTA-L1-26 Author Permission
TTA-L1-27 Proposal States
TTA-L1-29 Create Proposal from Existing
TTA-L1-30 Proposal Migration
TTA-L1-33 External TAC Proposal
TTA-L1-34 Notifications
TTA-L1-35 External TAC Proposal Notification
TTA-L1-36 Vetting
TTA-L1-37 Solicitation Vetting
TTA-L1-38 Science Category Vetting
TTA-L1-39 Solicitation Types
TTA-L1-40 Check Solicitation Boundary
TTA-L1-42 Science Category Vetting Interface
Allocation Disposition
Allocation Request
Author
Author Information Service
Author Information Configuration File
Author Information Service Interface
Data Processing Specification
Notification Group
Notification System
NRAO Account System
Observing Specification
Proposal
Proposal Factory
Proposal Information
Proposal Repository
Proposal Service
Proposal Service Interface
Request Specification
Solicitation Service
Solicitation Service Interface
Technical Justification
UX

2.1.2.3 Configure Review

NRAO primarily conducts two types of review processes, Panel Proposal Reviews and Observatory Site Reviews. The Panel Proposal Review consists of Feasibility Reviews, Individual Science Reviews and Consensus Reviews[16]. Feasibility and Individual Science Reviews require panels to be created and maintained throughout the review process while adhering to rules governing the relationships between reviewers, panels, and review materials. The Configure Review package contains all the concepts needed to create and manage Science and Feasibility reviews.

2.1.2.3.1 Science Review Configuration

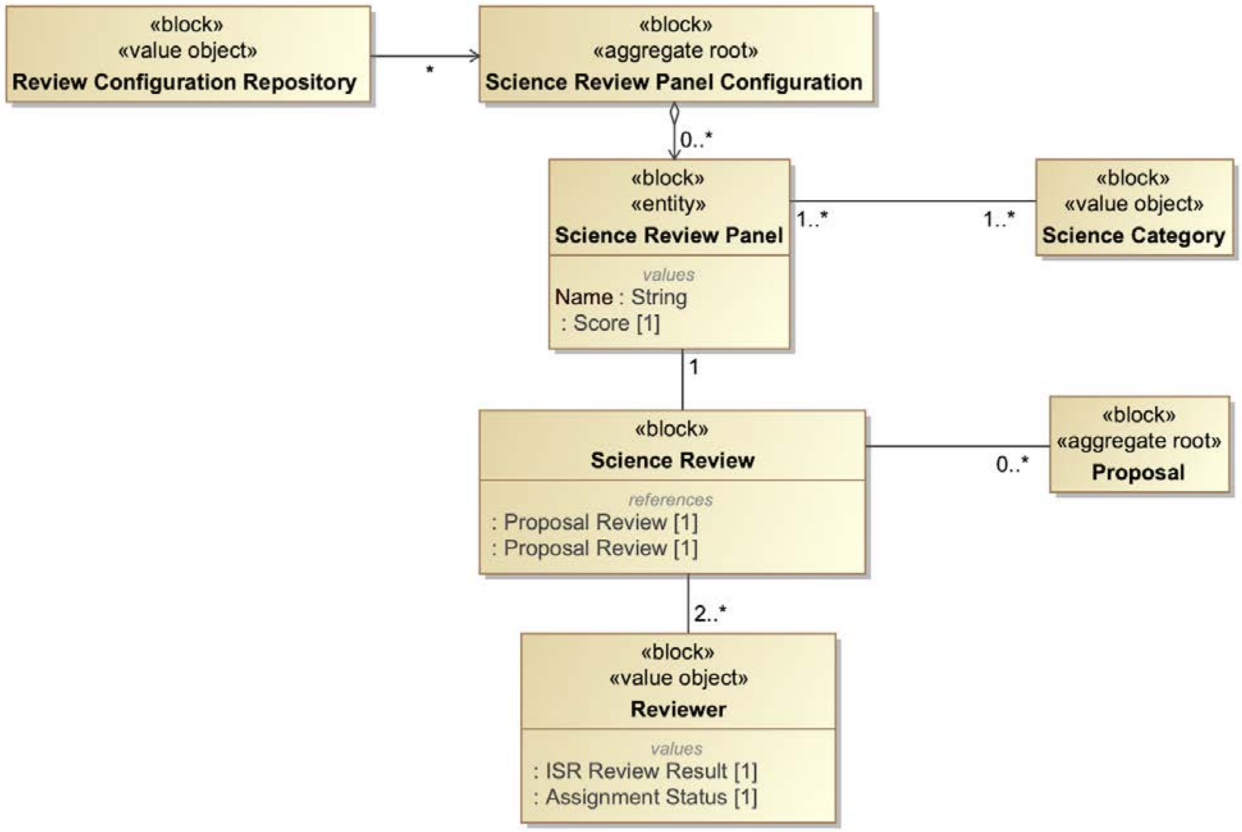
In the view provided below, the Science Review represents a ternary relationship with the following multiplicities:

(SRP, Reviewer) : 0..* Proposal
(SRP, Proposal) : 2..* Reviewer
(Reviewer, Proposal) : 1 SRP

Each part of the ternary relationship is determined by 'fixing' the association on the left to determine the multiplicity on the right. This arrangement satisfies the requirements that a Reviewer can only be on one Science Review Panel and each Proposal must be assigned two or more Reviewers.

[16] For details, see section 3.5 in "Telescope Time Allocation (TTA): System Description", Balser et al., Jan. 31, 2020

2.1.2.3.1.1 Primary Presentation



2.1.2.3.1.2 Element Catalog

Domain Object	Definition
Science Review	A panel-based, dual anonymous process designed to evaluate the scientific merit of proposals.
Science Review Panel (SRP)	A group of people who are tasked to review the scientific merit of a Proposal. Each SRP has a chair and, potentially, a chair pro tem. There is a many-to-many relationship between Science Categories and SRPs.
Reviewer	A person who evaluates the scientific merit of a proposal.

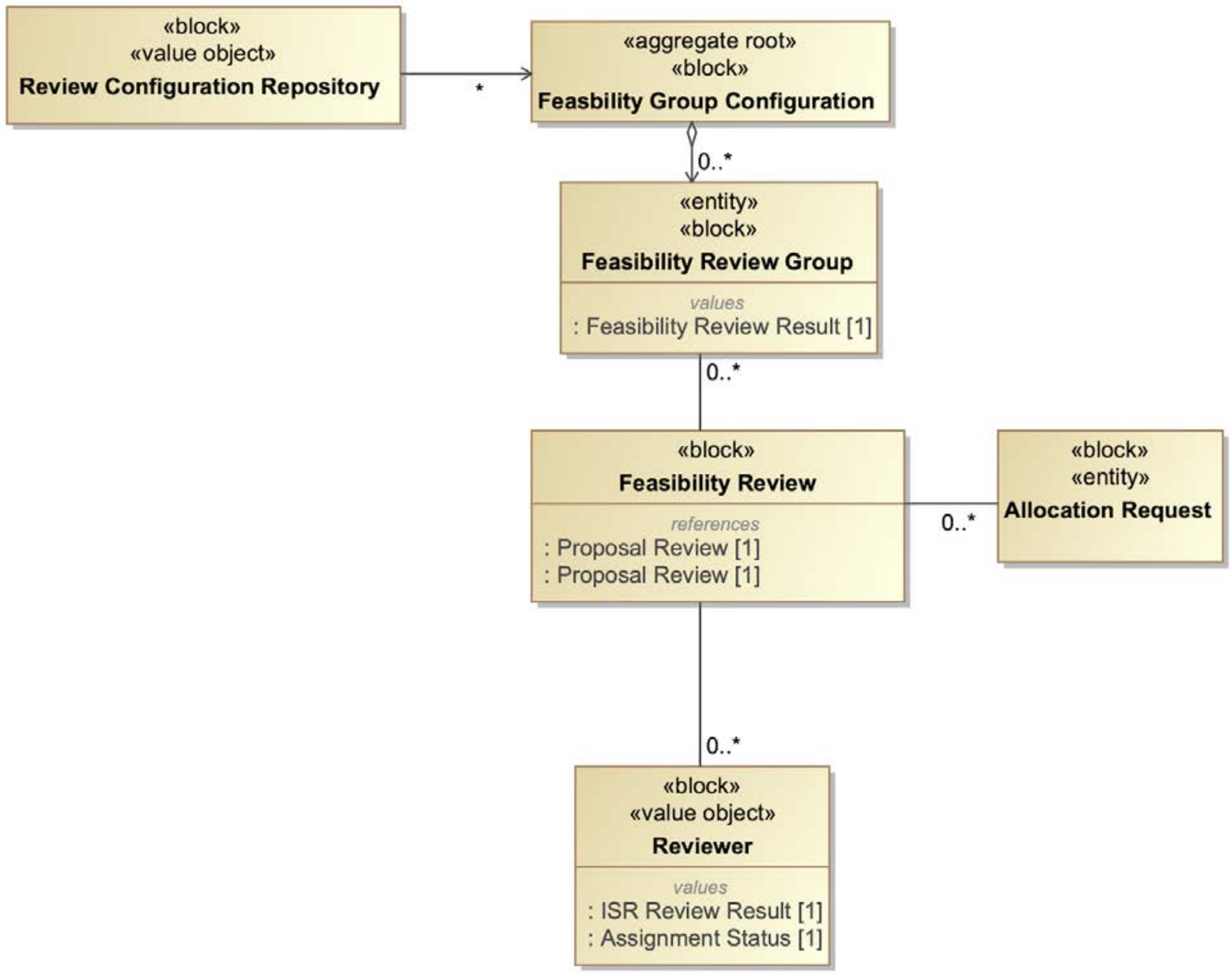
2.1.2.3.2 Feasibility Group Configuration

In the view provided below, the Feasibility Review represents a ternary relationship with the following multiplicities:

(FRG, Reviewer) : 0..* Allocation Request
(FRG, Allocation Request) : 0..* Reviewer
(Reviewer, Allocation Request) : 0..* FRG

Each part of the ternary relationship is determined by 'fixing' the association on the left to determine the multiplicity on the right. This arrangement expresses the many-to-many relationship between Feasibility Review Groups, Reviewers, and Allocation Requests.

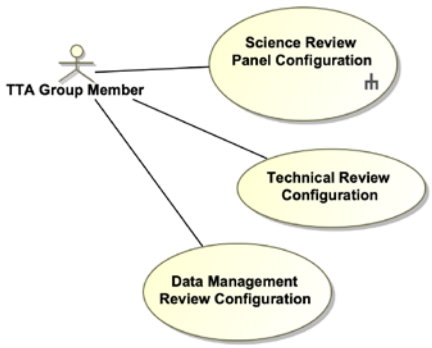
2.1.2.3.2.1 Primary Presentation



2.1.2.3.2.2 Element Catalog

Domain Object	**Definition**
Feasibility Review	A review of the feasibility (technical or data management) of a given Allocation Request.
Feasibility Review Group	Consists of one or more feasibility reviewers that are tasked to review the same set of Allocation Requests.

2.1.2.3.3 Use Cases



2.1.2.3.4 Requirements Mapping

Legend
↗ Satisfy
⇢ Satisfy (Implied)

	TTA-L1-46 Science Review Panel Configuration	TTA-L1-47 Feasibility Review Configuration	TTA-L1-48 Feasibility Review Assignments	TTA-L1-49 Starting SRP Configuration	TTA-L1-50 SRP Definition	TTA-L1-51 Data Management Review Configuration	TTA-L1-52 Review Panel Setup Access	TTA-L1-53 Review Configuration File	TTA-L1-54 Applying Configuration File Changes
Feasbility Group Configuration		↗	↗			↗	↗		
Feasibility Review			↗			↗			
Feasibility Review Group		↗	↗			↗			
NRAO Account System							↗		
Proposal Service			⇢		⇢				
Proposal Service Interface			↗		↗				
Review Configuration Repository	↗	↗	↗	↗	↗	↗	↗		
Review Configuration Service	⇢	⇢	⇢	⇢	⇢	⇢	⇢	⇢	⇢
Review Configuration Configuration File								↗	↗
Review Configuration Service Interface	↗	↗	↗	↗	↗	↗	↗	↗	↗
Reviewer		↗	↗		↗				
Science Review					↗				
Science Review Panel	↗				↗				
Science Review Panel Configuration	↗			↗	↗		↗		
Solicitation Service				⇢					
Solicitation Service Interface				↗					
UX	↗	↗	↗	↗	↗	↗	↗	↗	↗

2.1.2.4 Review

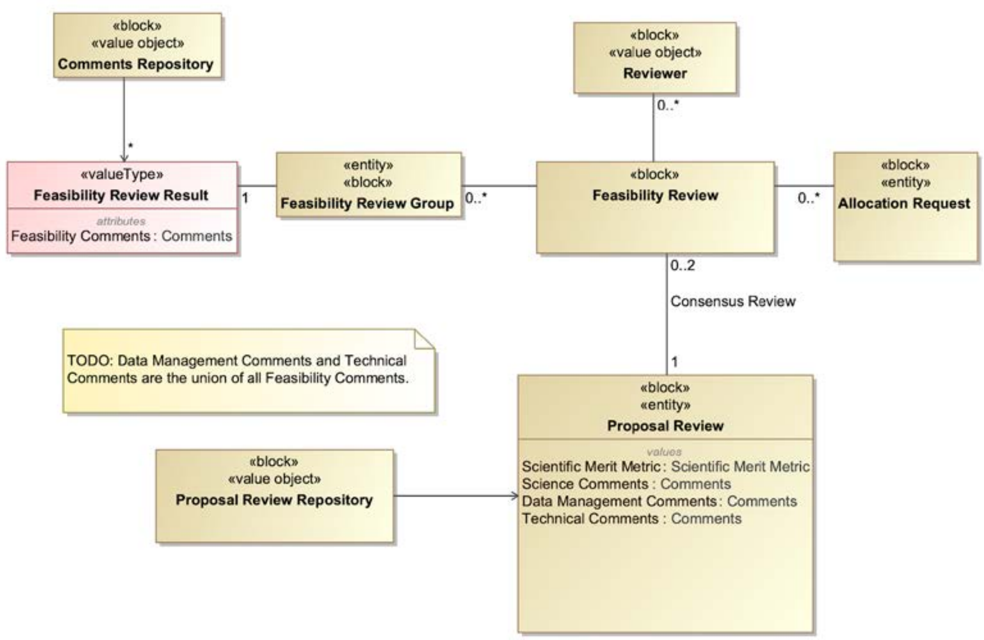
NRAO primarily conducts two types of review processes, Panel Proposal Reviews and Observatory Site Reviews. The Panel Proposal Review consists of Feasibility Reviews, Individual Science Reviews and Consensus Reviews; information from the Feasibility and Individual Science Reviews is used in the Consensus Review to quantitatively rank proposals. The ranking is expressed in the Proposal Review entity. For Observatory Site Reviews, TTA Group Members generate Proposal Reviews with qualitative scores.

The Review package contains all the concepts needed to conduct Panel Proposal Reviews and Observatory Site Reviews.

2.1.2.4.1 Panel Proposal Review – Feasibility

The key idea in this arrangement is that the Feasibility groups produce comments which are discussed and refined during the Consensus Review. The final resulting comments become part of the Proposal Review entity.

2.1.2.4.1.1 Primary Presentation

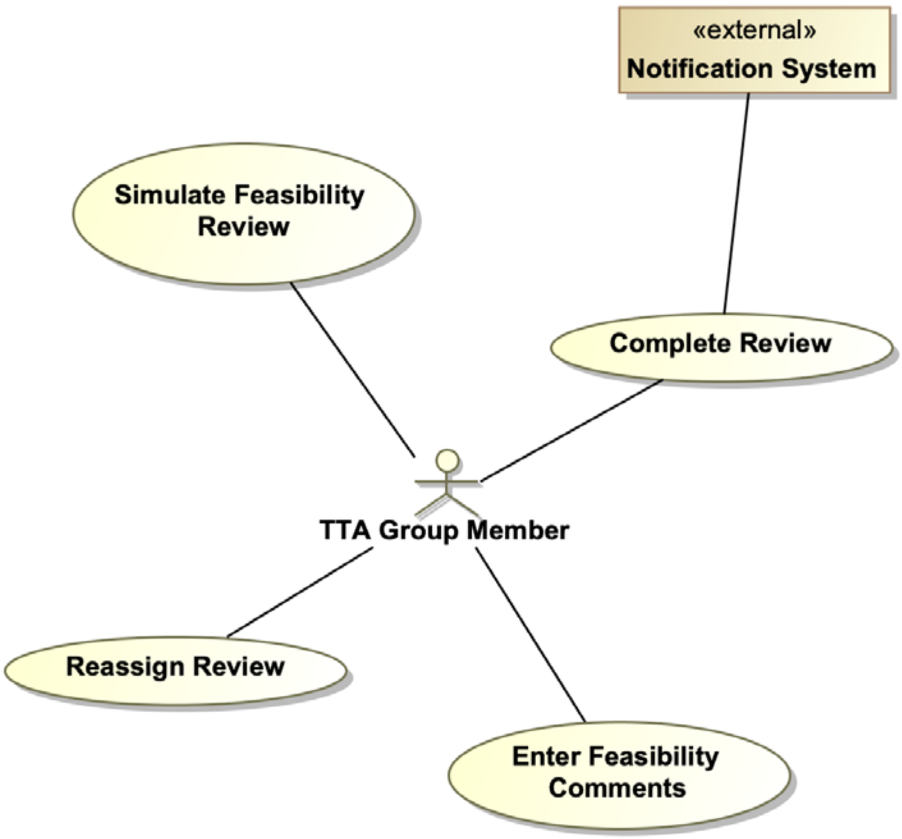


2.1.2.4.1.2 Element Catalog

Domain Object	Definition
Feasibility Review Result	Structure containing feasibility comments associated with an Allocation Request.

Proposal Review	An evaluation of the scientific merit and feasibility of the proposal. A proposal review consists of comments for the PI, internal comments, and a scientific merit metric.

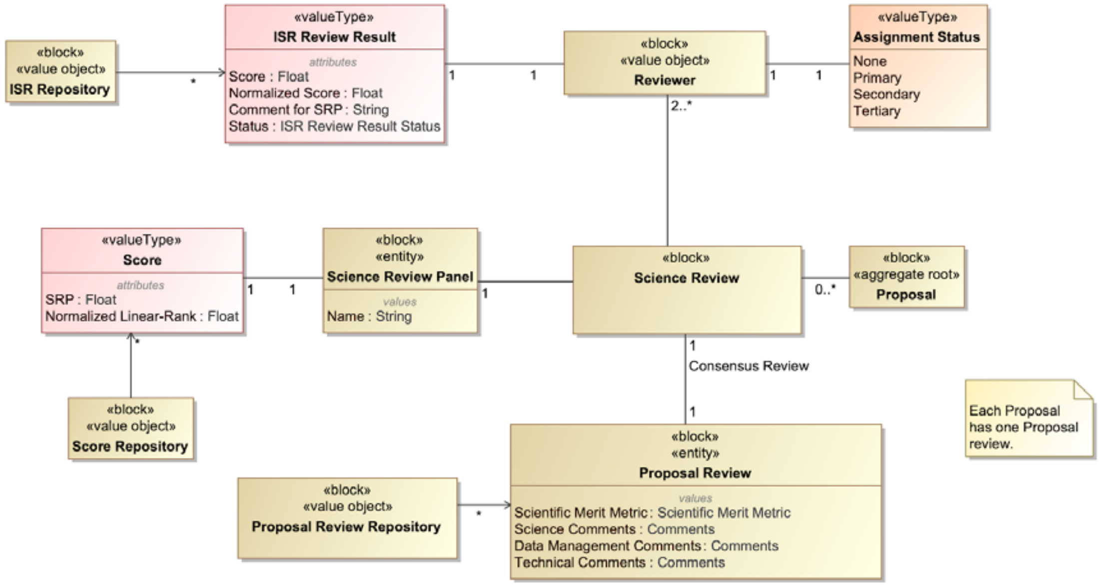
2.1.2.4.1.3 Use Cases



2.1.2.4.2 Panel Proposal Review – Consensus Science

The key difference between the Consensus Science review and the Consensus Feasibility review is that the science review involves scores (i.e. Individual Science Review scores - raw and normalized-, Science Review Panels scores, and Normalized Linear-Rank scores) that algorithmically yield a quantitative Scientific Merit Metric.

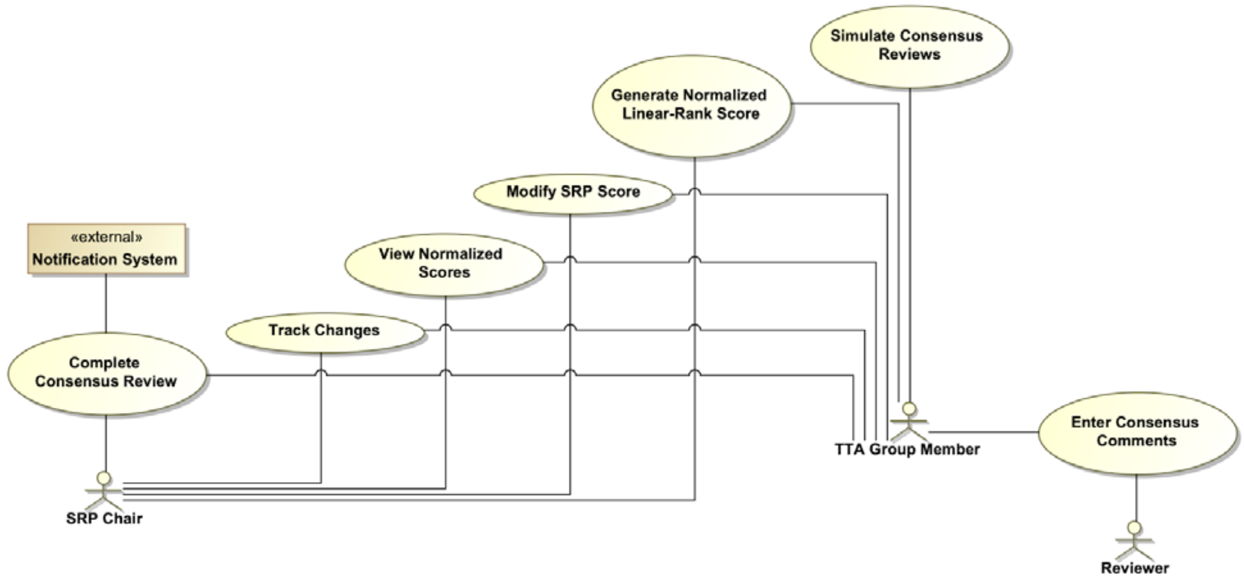
2.1.2.4.2.1 Primary Presentation



2.1.2.4.2.2 Element Catalog

Domain Object	Definition
ISR Review Result	Structure containing information pertaining to an Individual Science Review.
Score	Structure containing SRP and Normalized Linear-Rank scores for a proposal.
Proposal Review	An evaluation of the scientific merit and feasibility of the proposal. A proposal review consists of comments for the PI, internal comments, and a scientific merit metric.

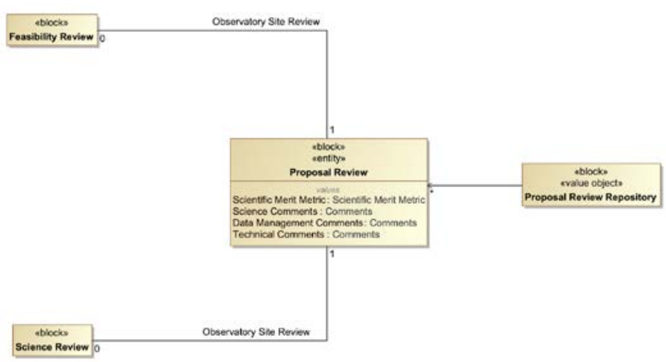
2.1.2.4.2.3 Use Cases



2.1.2.4.3 Observatory Site Review

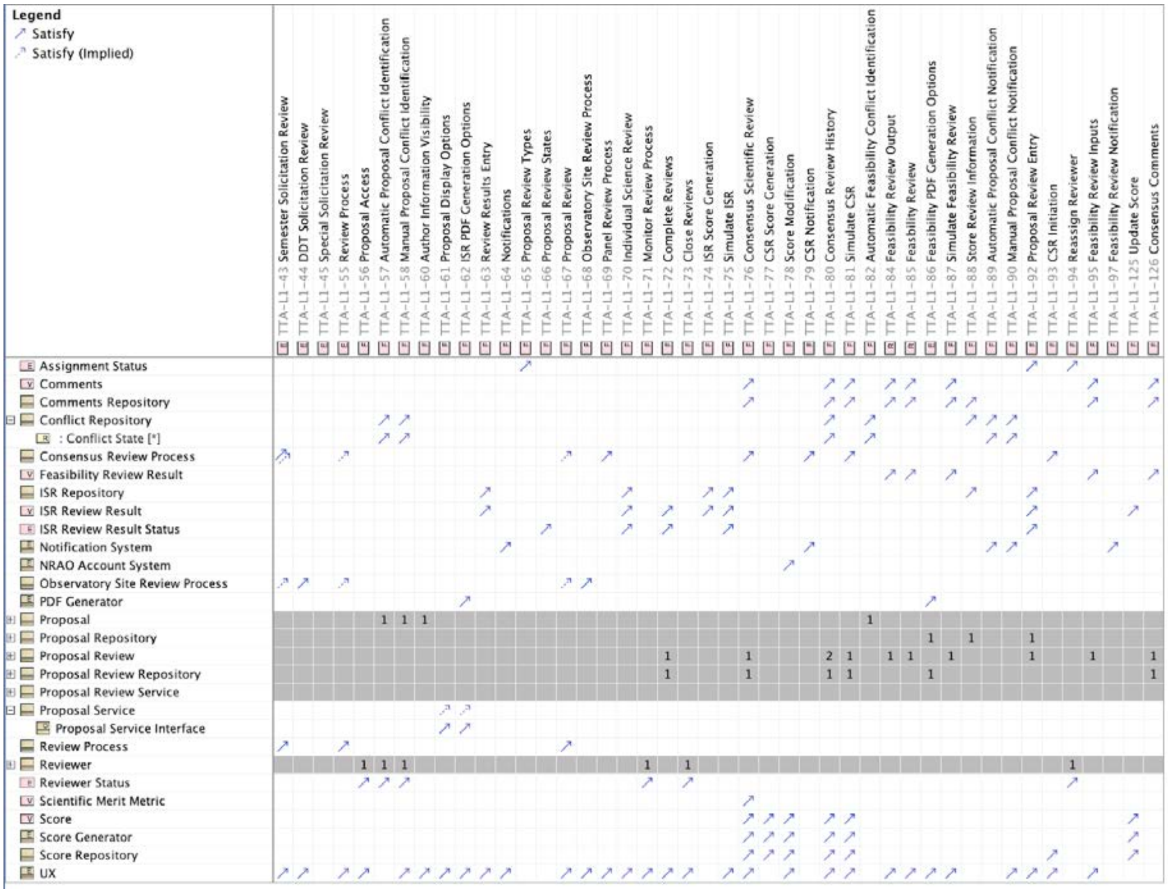
Observatory Site Reviews do not involve Feasibility groups or Science Review Panels. TTA Group Members create Proposal Reviews entities and manually enter Boolean Scientific Merit Metrics.

2.1.2.4.3.1 Primary Presentation

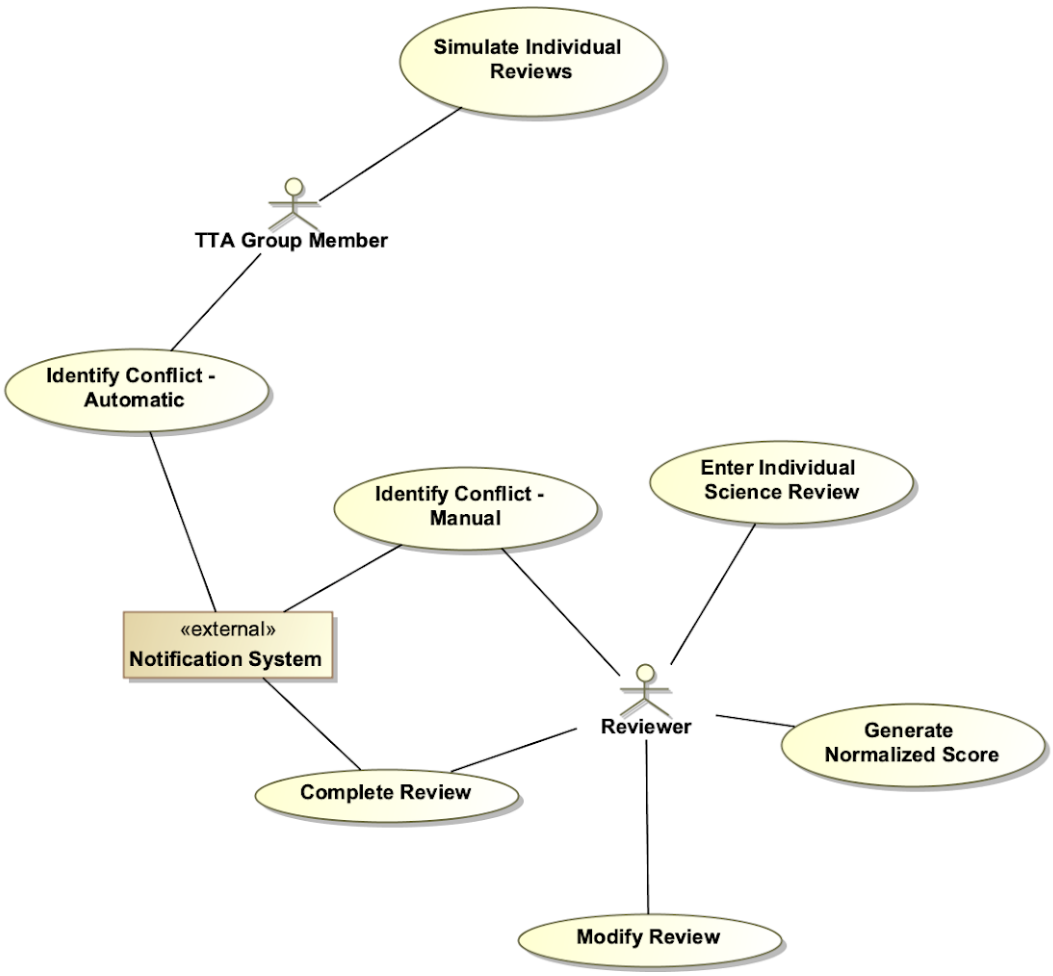


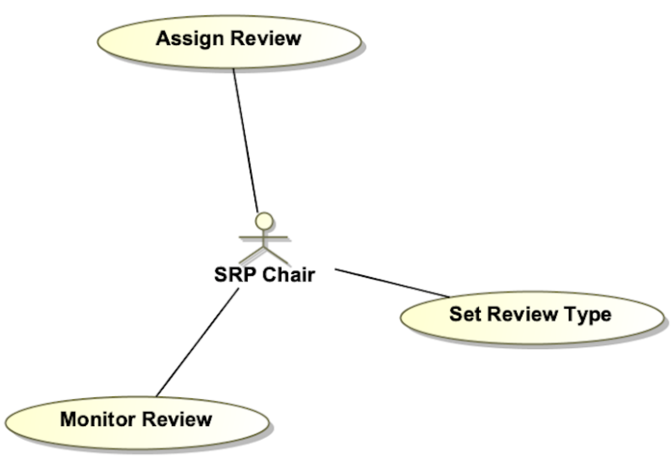
2.1.2.4.3.2 Use Cases

2.1.2.4.4 Requirements Mapping



2.1.2.4.5 Use Cases



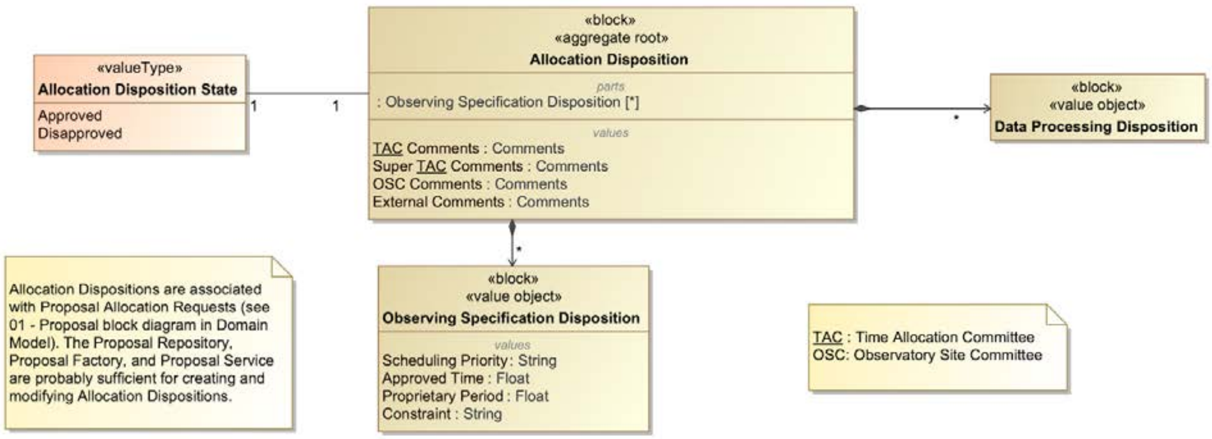


2.1.2.5 Allocate

The information produced by various review processes is used to allocate telescope time in Time Allocation Committee meetings or Observatory Site Committee meetings or External Committee meetings. Allocation Disposition entities model awards and include technical information related to facility resources as well as comments from review groups. Allocation Disposition entities are associated with Allocation Request entities. The Allocate package contains all the concepts related to reports needed in the committee meetings that generate Allocation Dispositions.

2.1.2.5.1 Allocation Disposition

2.1.2.5.1.1 Primary Presentation



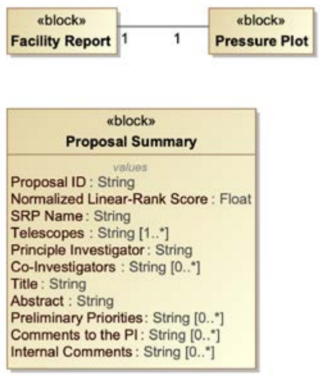
2.1.2.5.1.2 Element Catalog

Domain Object	Definition
Allocation Disposition	The disposition of a given Allocation Request to use observatory resources. This includes scheduling priorities, approved time, disposition comments, disposition constraints, and proprietary periods.

2.1.2.5.2 Allocation Reports

TTA Group members draft reports providing narratives of the scheduling issues for each Facility. These reports, along with pressure plots, are used in committee meetings to make allocation decisions.

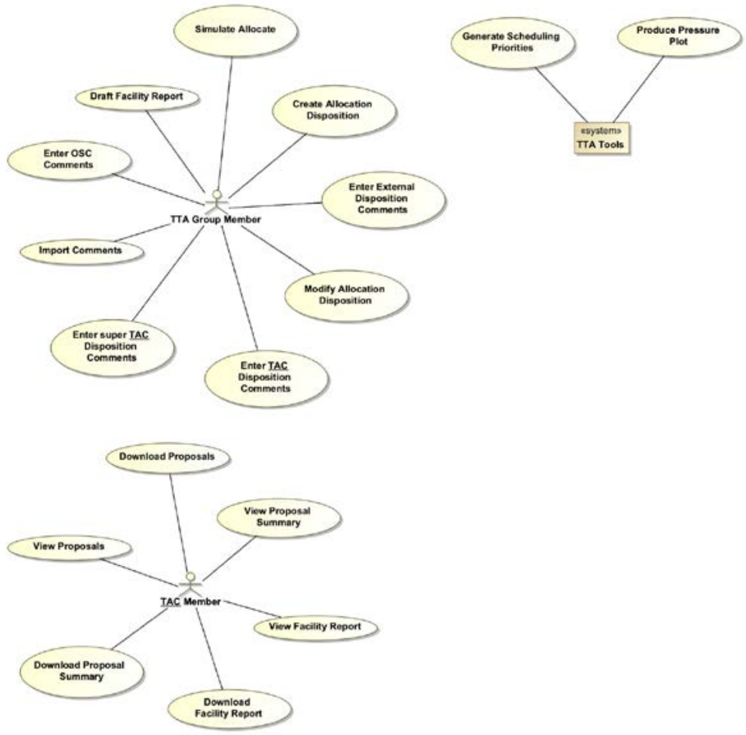
2.1.2.5.2.1 Primary Presentation



2.1.2.5.2.2 Element Catalog

Domain Object	Definition
Proposal Summary	A summary consisting of the PROPOSAL ID, NORMALIZED LINEAR-RANK SCORE, SRP NAME, TELESCOPES, PRINCIPAL INVESTIGATOR, CO-INVESTIGATORS, TITLE, ABSTRACT, PRELIMINARY PRIORITIES, COMMENTS FOR THE PI, and INTERNAL COMMENTS.
Facility Report	
Pressure Plot	A plot of the allocated hours as a function of LST (or GST) for a given Facility, broken down by scheduling priority and weather.

2.1.2.5.3 Use Cases



2.1.2.5.4 Requirement Mapping

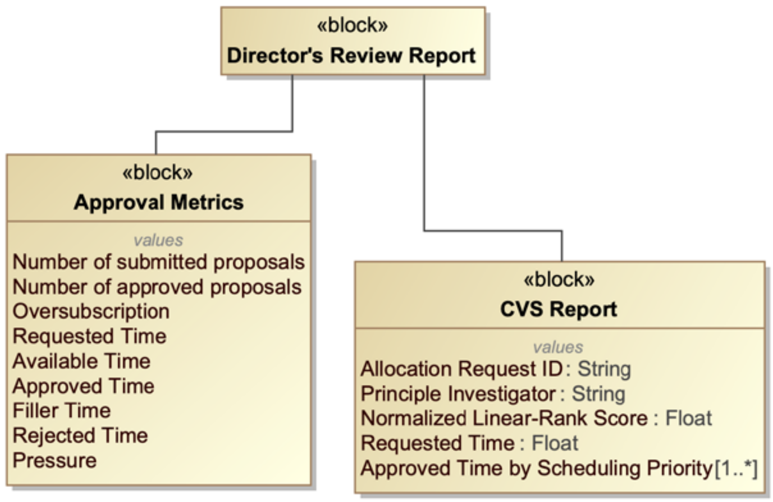
Legend
↗ Satisfy
⇢ Satisfy (Implied)

	TTA-L1-98 Panel Allocate Process	TTA-L1-99 Observatory Site Allocate Process	TTA-L1-100 Special Solicitation Allocate Process	TTA-L1-101 TAC Proposal View	TTA-L1-102 TAC Proposal Summary View	TTA-L1-103 TAC Facility Report View	TTA-L1-104 TAC Comments for the PI	TTA-L1-105 Super TAC Meeting Comments	TTA-L1-107 TAC Testing	TTA-L1-108 Create Allocation Disposition
Allocation Disposition	⇢	↗	⇢				↗	↗	↗	↗
Allocation Disposition State	↗	↗	↗							
Data Processing Disposition	↗		↗							
Facility Report						↗				
Observing Specification Disposition	↗		↗							
Pressure Plot						↗				
Proposal Repository				↗	↗					
Proposal Service				⇢	⇢					
Proposal Service Interface				↗	↗					
Proposal Summary					↗					
UX		↗	↗	↗	↗	↗	↗	↗	↗	↗

2.1.2.6 Approve

After committees make allocation recommendations, Directors (or their delegate) finalize allocation decisions which are expressed in reports. The Approve package contains the report-related entities.

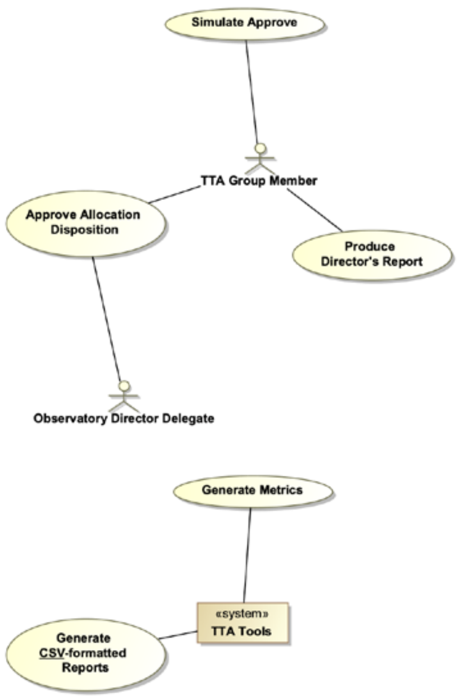
2.1.2.6.1 Primary Presentation



2.1.2.6.2 Element Catalog

Domain Object	Definition
Directors' Review Report	A report written by the TTA Group for the NRAO/GBO Director that summarizes the recommendations made by the TAC for semester Solicitations.
Approval Metrics	TTA Process Statistics
CSV Report	A CSV formatted version of a Director's report.

2.1.2.6.3 Use Cases



2.1.2.6.4 Requirements Mapping

Legend
↗ Satisfy
⇢ Satisfy (Implied)

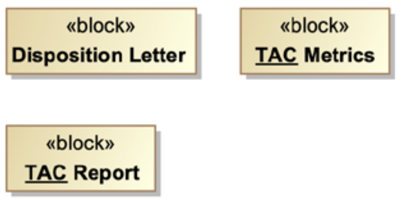
	TTA-L1-109 Generate CVS Spreadsheet	TTA-L1-110 Generate Metrics	TTA-L1-111 Panel Review Process Allocation Disposition Approval	TTA-L1-112 Allocation Disposition Testing	TTA-L1-113 Edit Allocation Disposition	TTA-L1-114 Director's Review Report	TTA-L1-115 Observatory Site Review Allocation Disposition Approval
Allocation Disposition State			↗	↗	↗		↗
Approval Metrics		↗					
CSV Report	↗						
Director's Review Report						↗	
Metrics Generator		↗					
Proposal Service			⇢	⇢	⇢		
Proposal Service Interface			↗	↗	↗		
UX	↗	↗	↗	↗	↗	↗	↗

Note: TTA-L1-109 should read CSV, not CVS.

2.1.2.7 Closeout

The Closeout package includes place-holder concepts related to the final steps of the TTA process. These concepts will be further refined in subsequent phases.

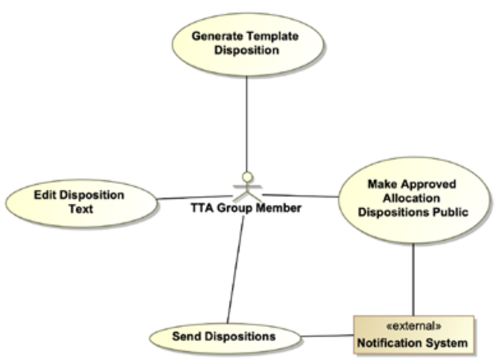
2.1.2.7.1 Primary Presentation



2.1.2.7.2 Element Catalog

Domain Object	**Definition**
Disposition Letter	A letter (or email) sent to the authors of a submitted proposal that summarizes the results of the review process.
TAC Metrics	Time Allocation Committee statistics.
TAC Report	Time Allocation Committee report.

2.1.2.7.3 Use Cases



2.1.2.7.4 Requirements Mapping

TTA-L1-124 involves an interface to the archive which has not yet been analyzed; this requirement will be addressed in the Logical Phase.

Legend
↗ Satisfy

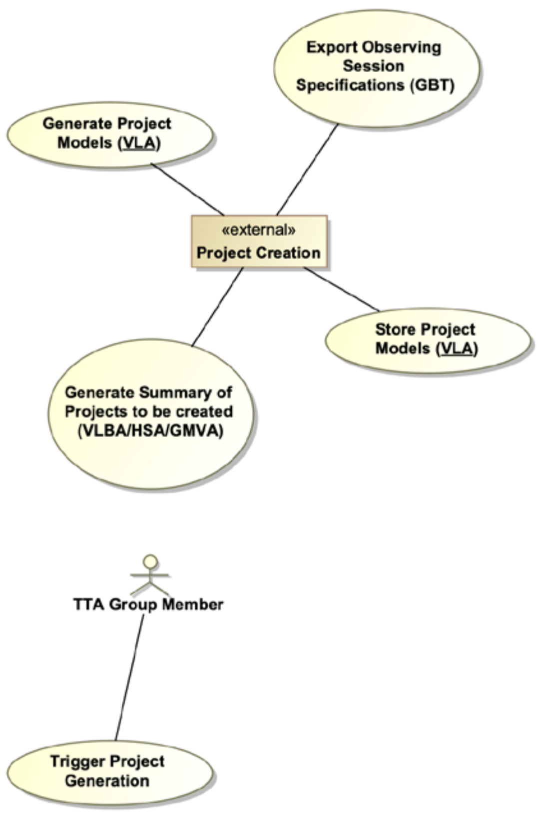
	TTA-L1-118 TAC Report	TTA-L1-119 Send Dispositions	TTA-L1-121 Generate Disposition Letter	TTA-L1-122 Disposition Template	TTA-L1-123 Edit Disposition Text	TTA-L1-124 Make Allocation Dispositions Public
TAC Report	↗					
Disposition Generator			↗			
Disposition Letter		↗	↗	↗	↗	
Notification System		↗				
TAC Metrics	↗					
UX	↗	↗	↗	↗	↗	

2.1.2.8 Create Project

As described in the Architectural Refinements section, each AUI/NA Facility has a unique project model and we plan to use the Anti-corruption Layer strategy to create an isolating layer providing other systems with information or functionality in terms of their own domain model. This strategy allows TTA to maintain a highly unified core conceptual model while supporting any existing or future facility.

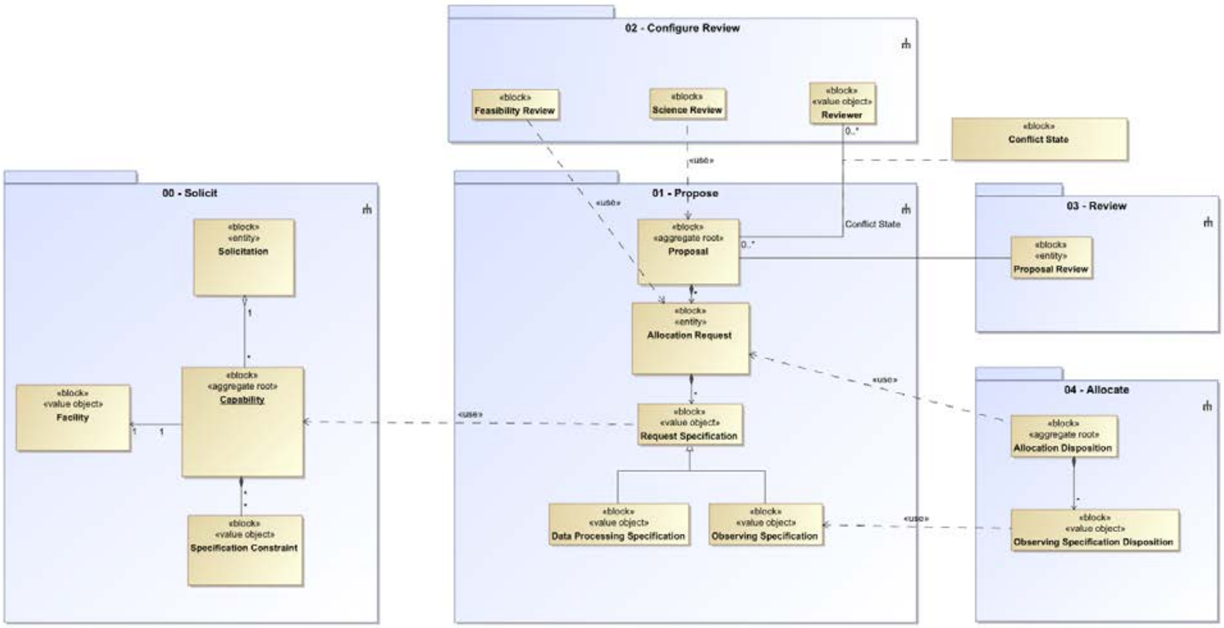
Refinement and development of this layer involves collaboration between different groups which will occur in a subsequent phase.

2.1.2.8.1 Use Cases



2.1.2.9 Package Dependencies

The purpose of this view is to show important conceptual dependencies between packages in the Domain Layer. Some of the details in each package have been suppressed to highlight the key dependencies.



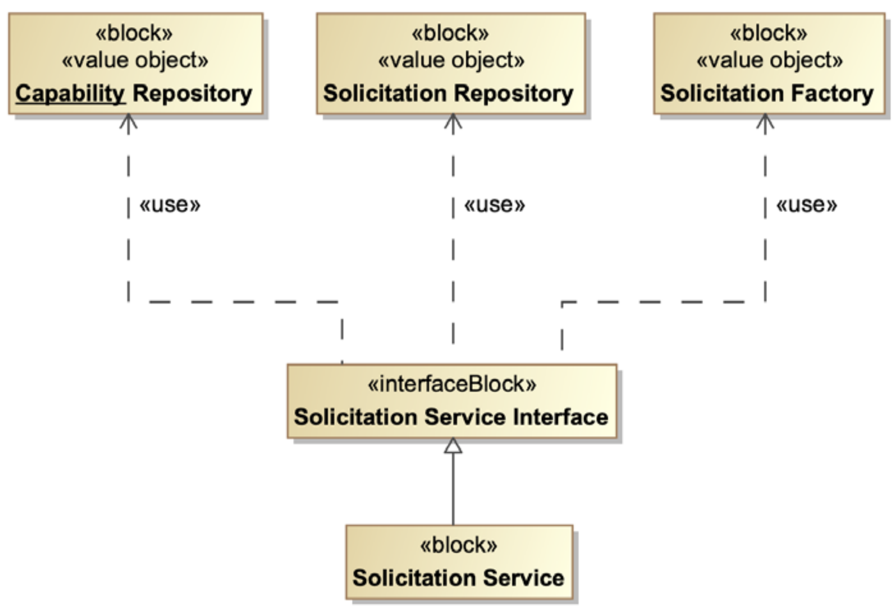
2.1.3 *Application Layer*

The Application Layer Model consists of a set of packages defining use cases which map to sections 3.1 to 3.9 in "TTA Tools System Description". In addition, the Application Layer Model includes entities in the Application Layer that orchestrate the use of entities found in the Domain Layer; these entities are modeled conceptually as Domain-Driven Design services (see 1.2.1.1). The services have been derived from the use cases.

2.1.3.1 Solicitation Service

The Solicitation Service represents the minimum design that supports configuring and opening a solicitation, modifying capabilities, and testing proposal validation. Initially, capability information will be provided in the Solicitation Configuration File. Solicitations are sufficiently complicated to require a factory as opposed to a simple constructor. A repository is required to support multiple concurrent solicitations.

2.1.3.1.1 Primary View

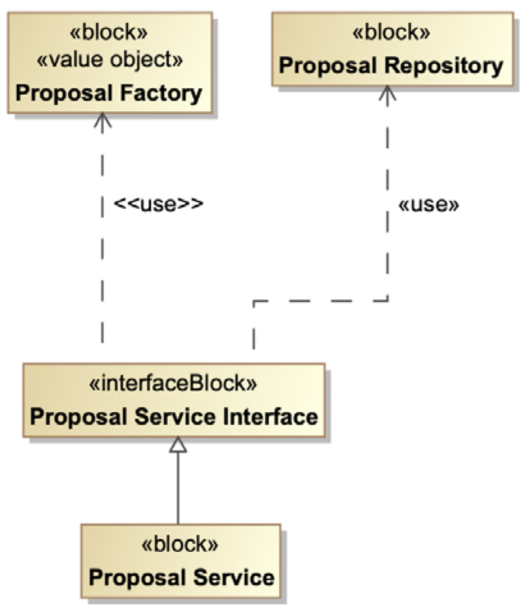


2.1.3.2 Proposal Service

The Proposal Service represents the minimum design that supports creating and vetting proposals.

Note that the Proposal Service has an unrealistically high number of allocations in various Satisfy matrices in the Domain Model. It is expected that the Proposal Service will be refined in subsequent phases to, for example, provide efficient access to entities associated with Proposals (e.g. Allocation Dispositions). A key advantage of the Logical Phase includes providing time for the DMS Architect to collaborate with the SSA Architect on issues like the Proposal Service.

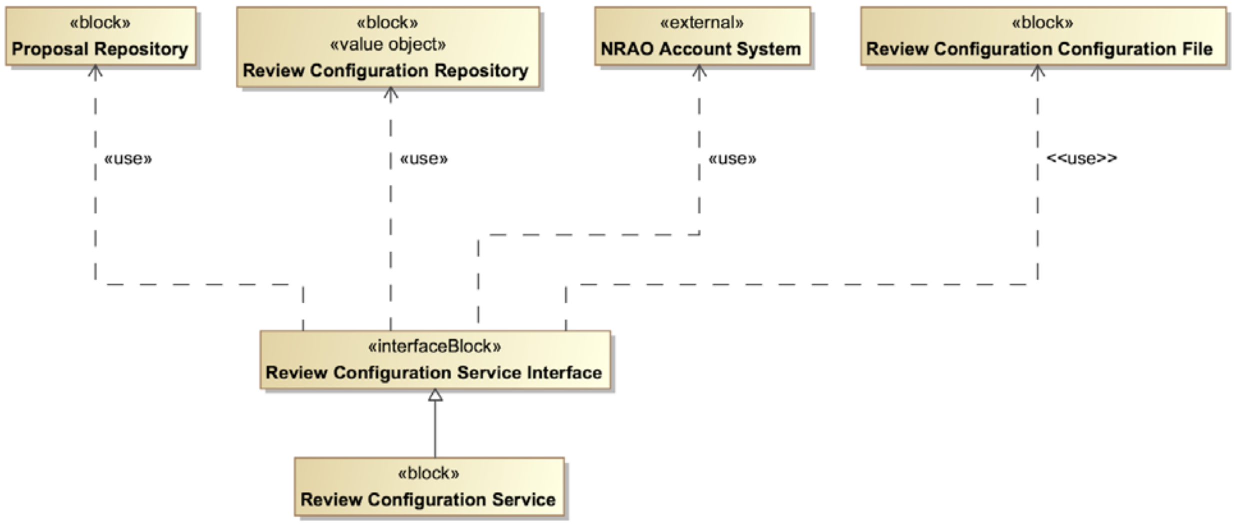
2.1.3.2.1 Primary Presentation



2.1.3.3 Review Configuration Service

The Review Configuration Service represents the minimum design that supports managing review groups, assigning reviewers to groups, and assigning proposals to reviewers.

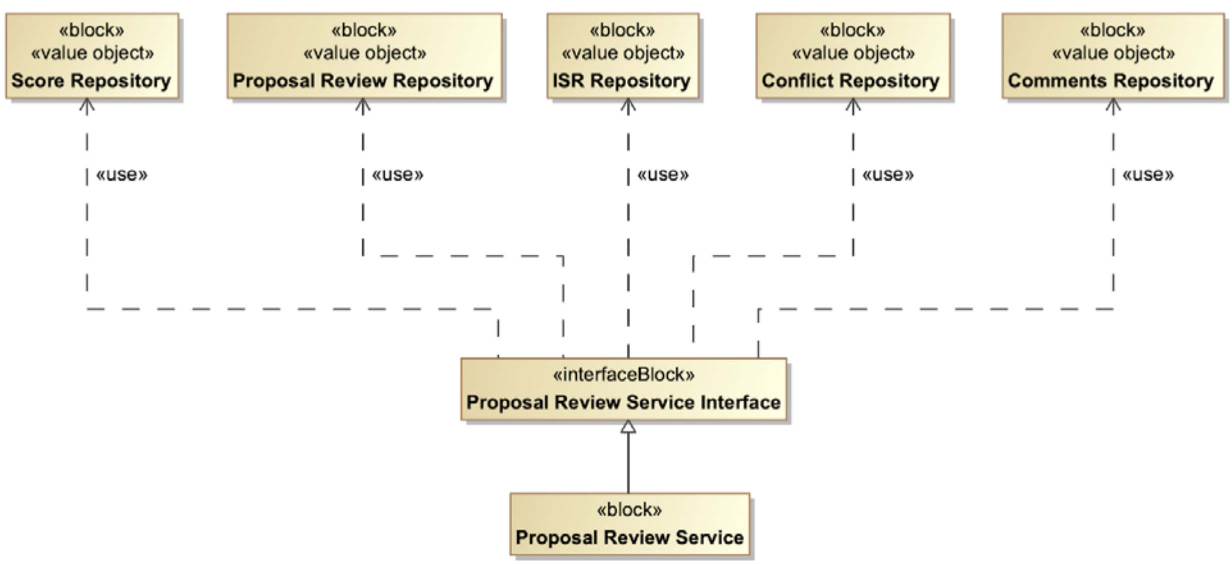
2.1.3.3.1 Primary Presentation



2.1.3.4 Proposal Review Service

The Proposal Review Service represents the minimum design that supports Panel Proposal and Observatory Site review processes.

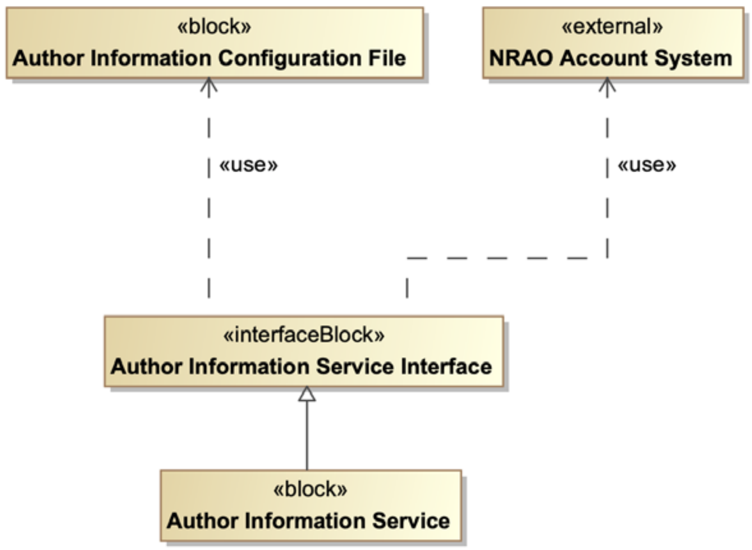
2.1.3.4.1 Primary Presentation



2.1.3.5 Author Information Service

The Author Information Service represents the minimum design that supports accessing author information via the NRAO Account System or configuration files. NRAO is in the planning phase of a project to update its user account system and this part of the design will be revisited at a later date.

2.1.3.5.1 Primary Presentation



2.1.3.6 Service Dependencies

Figure 7 shows the current relationships between Application Layer services and Domain Layer packages. It is expected that the services and their relationships to Domain Layer entities will change in the Logical and Physical phases.

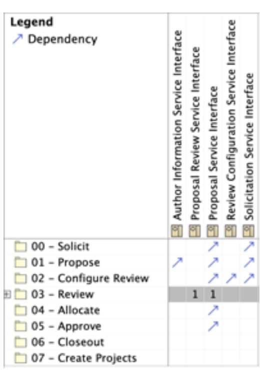


Figure 7 Dependencies between services and packages.

3 REFERENCED MATERIALS

1. "Telescope Time Allocation (TTA): Concept", Balser et al., 688-TTAT-002-MGMT, Jul. 02, 2019

2. “2019-03-TTA Tools Concept”, Kern et al., PowerPoint presentation

3. “2019-06-Project KickOff”, Kern, PowerPoint presentation

4. "Telescope Time Allocation (TTA): System Description", Balser et al., 688-TTAT-004-MGMT, Mar. 13, 2020

5. “Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing”, Vol 4, Buschmann et al., 2010

6. “Software Architecture in Practice”, Bass et al., 2013

7. “Designing Software Architectures: A Practical Approach”, Cervantes et al., 2016

8. “Domain-Driven Design: Tackling Complexity in the Heart of Software”, Evans, Eric, 2003

9. “Patterns of Enterprise Application Architecture”, Martin Fowler, 2003

10. “Growing Object-Oriented Software, Guided by Tests”, Freeman et al, 2009

11. “Telescope Time Allocation Tools Execution Plan”, Treacy, Kern, 688-TTAT-010-MGMT, Version 0.01