SCHEDULING AND PLANNING IN EXECUTIVE SYSTEMS

CONNOR LANGE

Overview

- Background
- Previous Systems
- GDMFAS
  - Task Manager
  - Scheduler
  - System Monitor
- Implementation Issues

Research/Topic Area

- Executive Systems in Space!
- Special Considerations
  - Extreme Risk – **Money** and Time
  - Communication – Little to None
  - “Weak” hardware – Flight Heritage
  - Mistrust of AI – Flight Heritage / Risk

Executive System Architecture Overview

- Agent architecture that acts as the “brain” of an autonomous system (AS)
- Handles all planning and scheduling of tasks and mission objective
- Monitors system health and consolidates information about the agent to preserve the system

Implications

- Significant resource usage
- Additional complexity
  - Software
  - Risk
  - Hardware
- Increased development time

Why?

- Human error due to:
  - Limited visibility of system state
  - Accidents (wrong commands, etc.)
  - Humans being slow and inefficient (other things)
- It’s necessary… for the precise reasons it’s dangerous

Overview

- Background
- Previous Systems
- GDMFAS
  - Data Items
  - Task Manager
  - Scheduler
  - System Monitor
- System Concerns

List of Previous Systems

- GENIE
- LOGOS
- ASOF
- ASPEN
- NMRA
- ASE
- CASPER
- NEAT

Ground Station Approach

- Systems: GENIE, LOGOS, ASOF
- Benefits:
  - Significant advantages all-around
    - Hardware
    - Error Correction
    - Software Complexity/Testing
- Cons:
  - Comms!

Platform Independent Approach

- Systems: ASPEN
- Benefits:
  - Generic/Abstract Implementation
  - Versatility in mission planning
    - Ground/Space Focus
  - Easier to interface with since humans think in terms of “high-level” operations
- Cons:
  - Doesn’t inherently address specific spacecraft concerns (hardware, etc.)

Spacecraft Approach

- Systems: NMRA, ASE
- Benefits:
  - Considers unique spacecraft problems
  - Immune to communication problems
  - Counteracts human error
  - Quick response time
- Cons:
  - Added complexity, risk, development time, etc.

Partial Solutions

- Systems: CASPER, NEAT
- Benefits:
  - UNIX philosophy – do 1 thing and do it well
  - Forces modular architecture
  - Provides versatility in system design and decreases future development time
- Cons:
  - No assumptions about system design
  - Communication with other modules

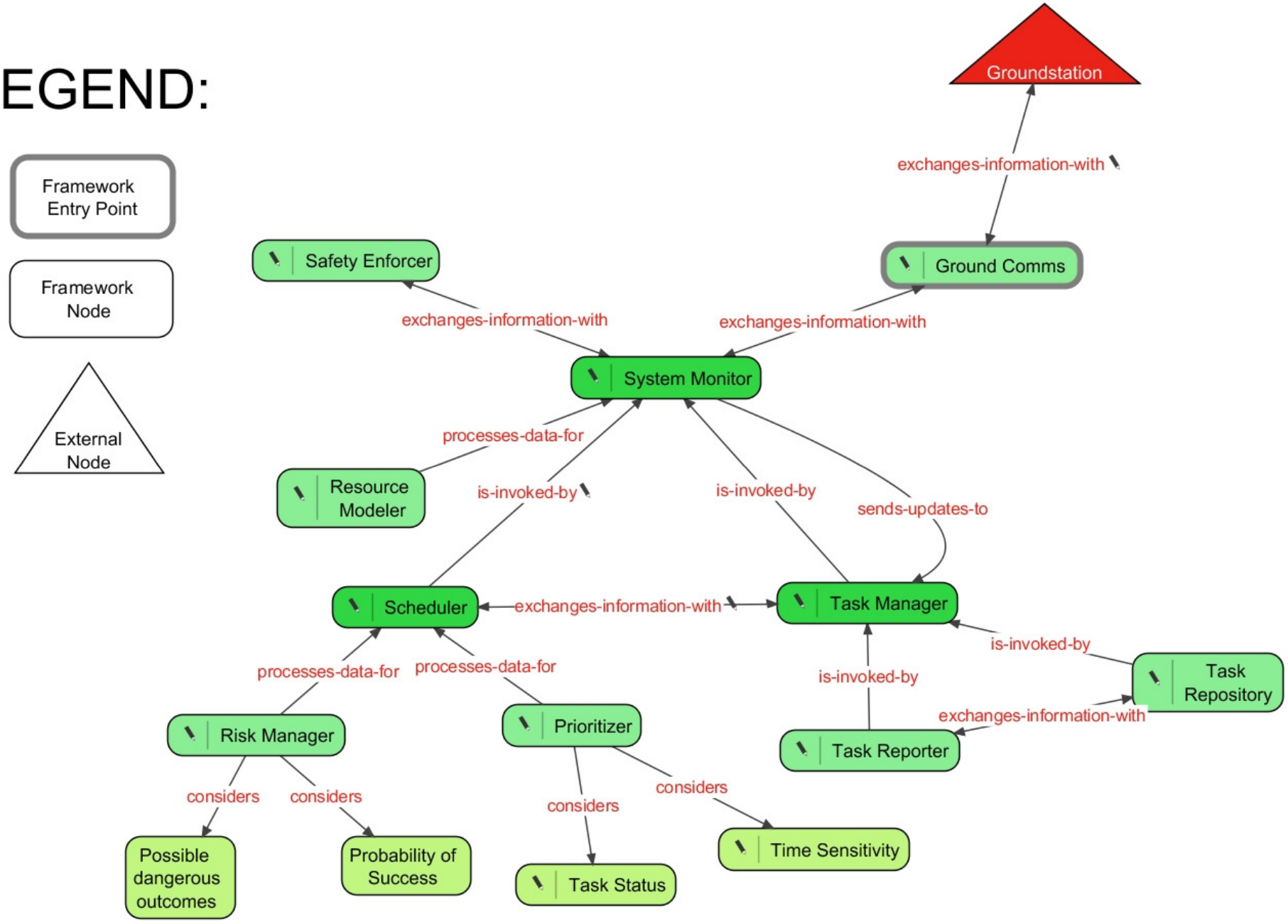
Overview

- Background
- Previous Systems
- GDMFAS
  - Data Items
  - Task Manager
  - Scheduler
  - System Monitor
- System Concerns

Generic Decision Making Framework for Autonomous Systems ("the framework")

- Relieve the headache
- Make development simpler and more cost effective
  - Don't reinvent the wheel
- Don't waste time with the "obvious"
  - Ex: C program headers, function declaration

LEGEND:


Framework Entry Point
Framework Node
External Node
Groundstation
exchanges-information-with
Safety Enforcer
Ground Comms
exchanges-information-with
exchanges-information-with
System Monitor
processes-data-for
Resource Modeler
is-invoked-by
is-invoked-by
sends-updates-to
Scheduler
exchanges-information-with
Task Manager
is-invoked-by
Task Repository
processes-data-for
processes-data-for
is-invoked-by
exchanges-information-with
Risk Manager
Prioritizer
Task Reporter
considers
considers
considers
considers
Possible dangerous outcomes
Probability of Success
Task Status
Time Sensitivity

Overview

- Background
- Previous Systems
- GDMFAS
  - Data Items
  - Task Manager
  - Scheduler
  - System Monitor
- System Concerns

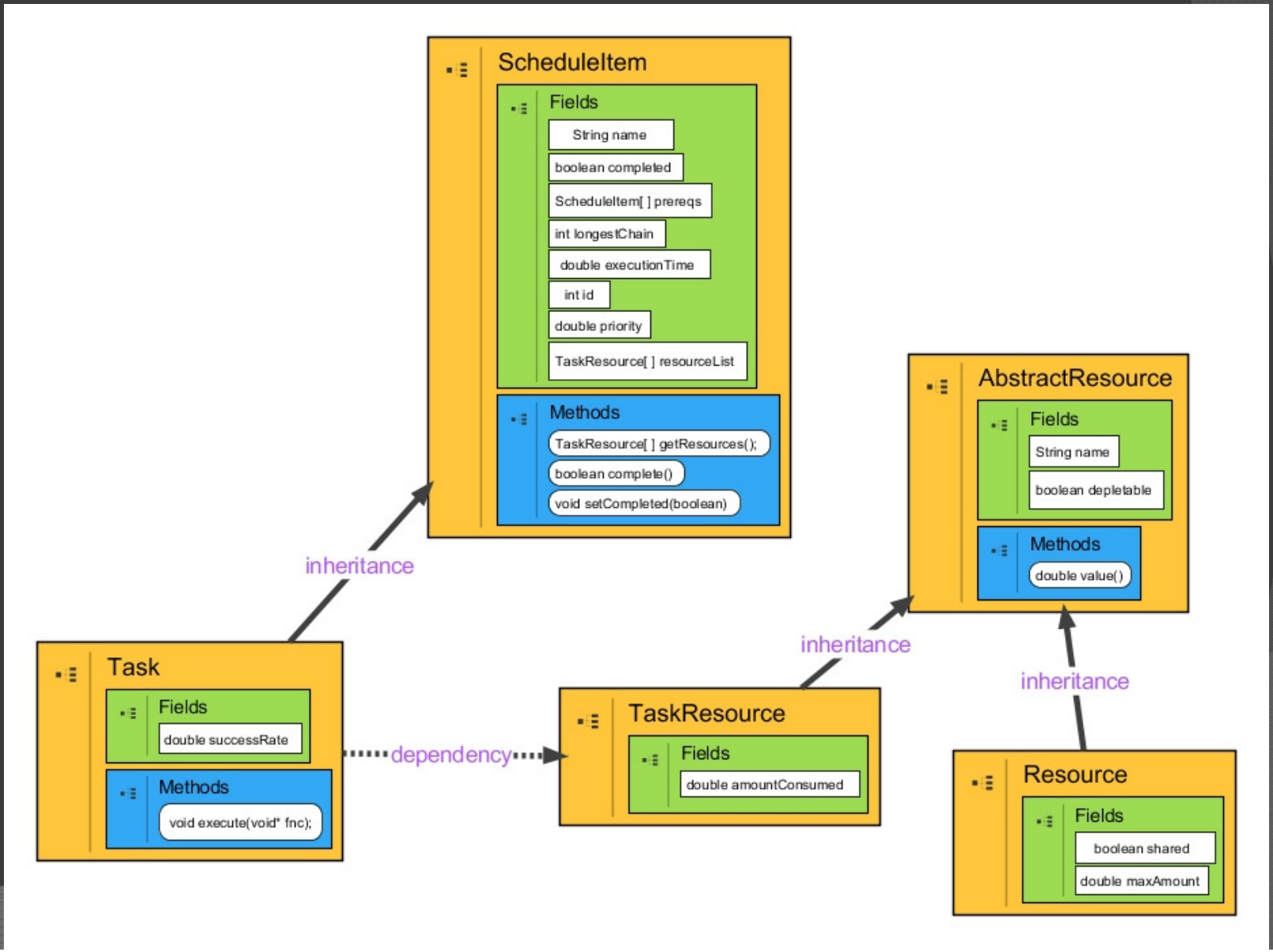
Data Item Design Decisions

- Executables
  - Effects on the system are all that matter
  - Tasks executed through user-code
- Resources
  - Numerical values and binary use indicators
  - Safety
    - Minimum safe level
    - Total amount

Data Items - Tasks

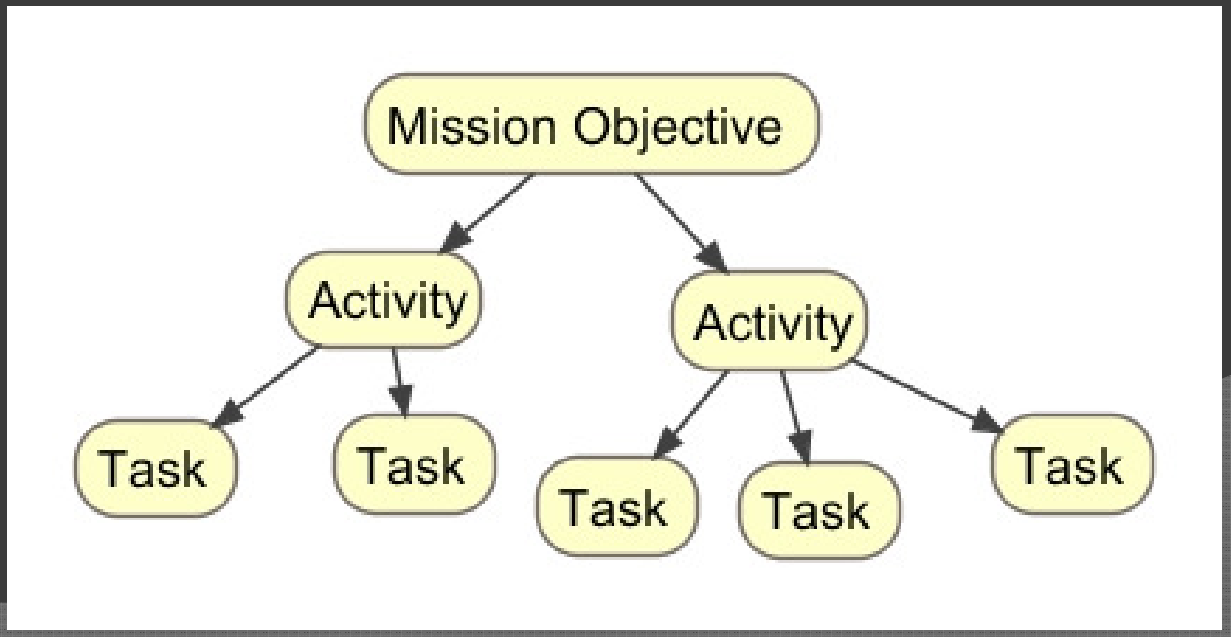
- Most basic schedulable unit
- Most common unit
- Use resources
- Only directly executable unit

Task Class Diagram



Data Items: Activities and Mission Objectives

- Derived from ScheduleItem
- **NOT** directly executable
  - Executes the Tasks or Activities contained within



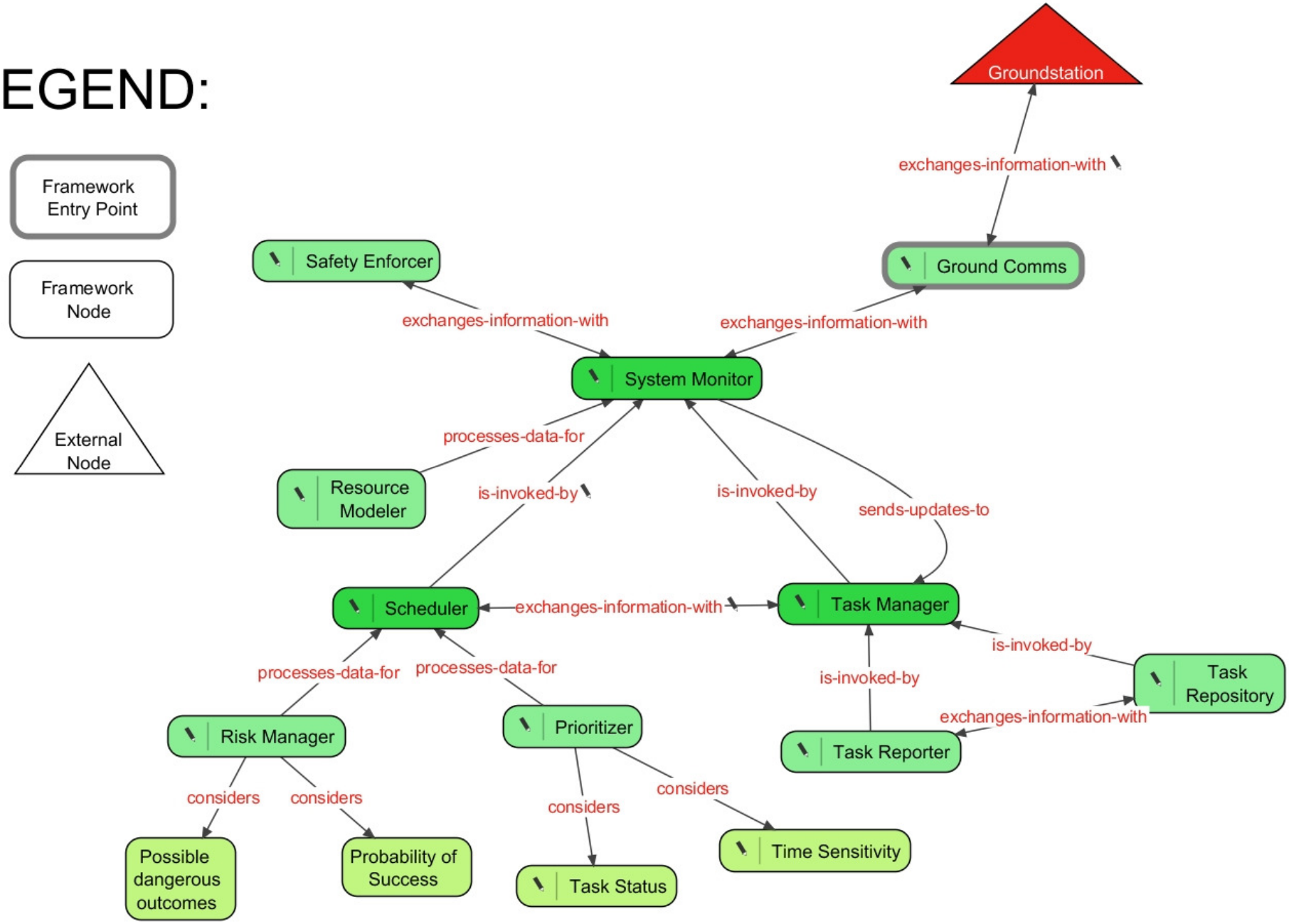
Data Items - Resource

- Used by the *System Monitor* to watch resource levels of all resources
- Stores:
  - The maximum capacity of the resource in the system
  - If the Resource is sharable or not

Overview

- Background
- Previous Systems
- GDMFAS
  - Data Items
  - Task Manager
  - Scheduler
  - System Monitor
- System Concerns

LEGEND:


Framework Entry Point
Framework Node
External Node
Groundstation
exchanges-information-with
Safety Enforcer
Ground Comms
exchanges-information-with
exchanges-information-with
System Monitor
processes-data-for
Resource Modeler
is-invoked-by
is-invoked-by
sends-updates-to
Scheduler
exchanges-information-with
Task Manager
is-invoked-by
Task Repository
processes-data-for
processes-data-for
is-invoked-by
exchanges-information-with
Risk Manager
Prioritizer
Task Reporter
considers
considers
considers
considers
Possible dangerous outcomes
Probability of Success
Task Status
Time Sensitivity

Task Manager

- Repository for all possible Tasks in the system
  - Keeps track of what has been completed and what hasn't
- Adds new Tasks
- Updates Tasks
- Relays repository information to the ground

Task Manager – Adding Tasks

- All Tasks that the developers are aware of at system creation time are automatically read
- Future Tasks must be transmitted
  - Mission completed, changed, or aborted
- Operator sends the data required for a Task to the system (“Task factory”)

Task Manager - Updates

- *System Monitor* sends updated Task properties to the repository where they are saved
  - Execution Time
  - Resource Usage
- Updates completion status
  - Task → Activity → Mission Objective

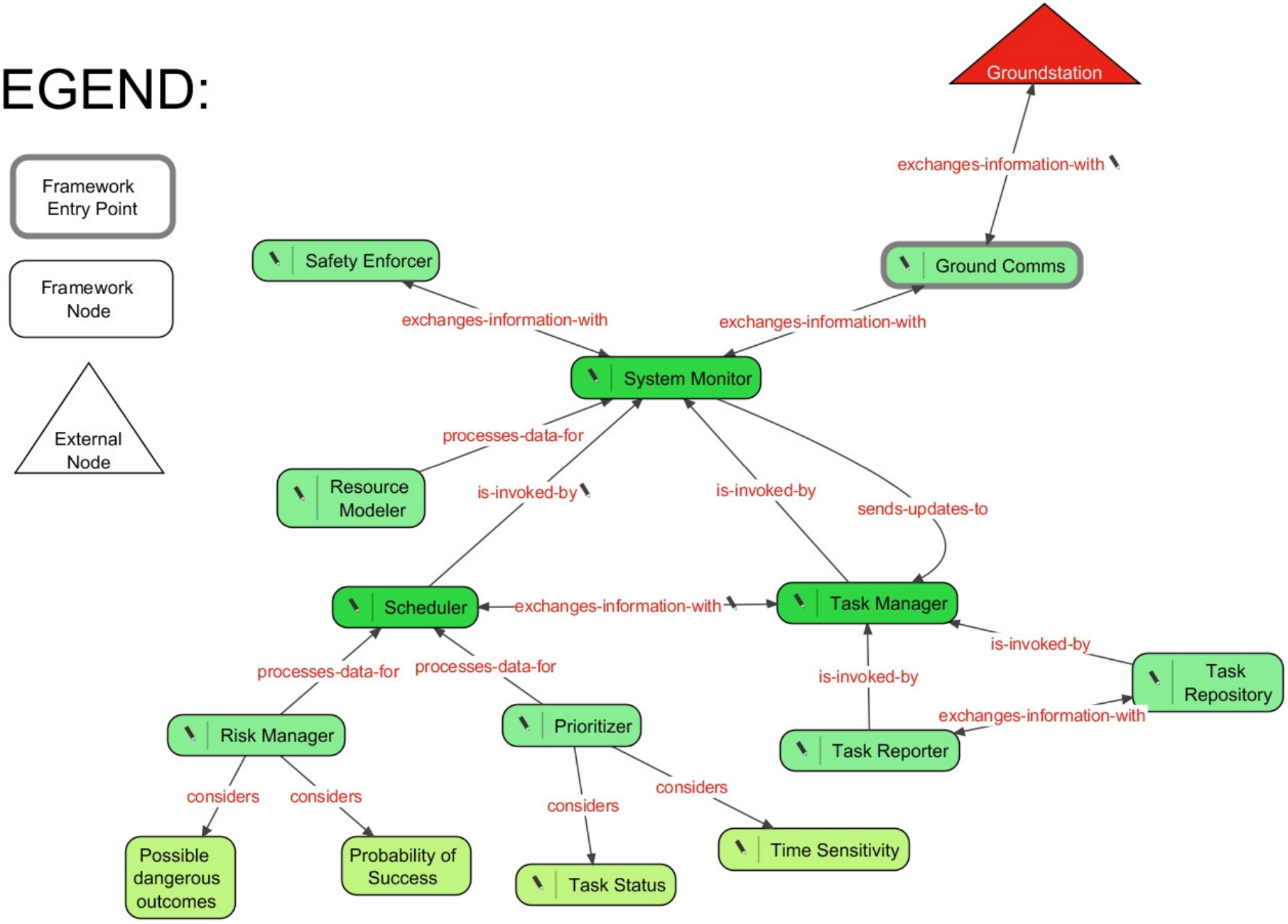
Task Manager – Relaying Status

- Creates a report of the current status of the Task repository
  - Overall Mission Status %
  - Mission Objectives %
    - List
  - Activities %
    - List
  - Tasks %
    - List

Overview

- Background
- Previous Systems
- GDMFAS
  - Data Items
  - Task Manager
  - Scheduler
  - System Monitor
- System Concerns

LEGEND:


Framework Entry Point
Framework Node
External Node
Groundstation
exchanges-information-with
Safety Enforcer
Ground Comms
exchanges-information-with
exchanges-information-with
System Monitor
processes-data-for
Resource Modeler
is-invoked-by
is-invoked-by
sends-updates-to
Scheduler
exchanges-information-with
Task Manager
is-invoked-by
Task Repository
processes-data-for
processes-data-for
is-invoked-by
exchanges-information-with
Risk Manager
Prioritizer
Task Reporter
considers
considers
considers
considers
Possible dangerous outcomes
Probability of Success
Task Status
Time Sensitivity

Planning/Scheduling Considerations

- Time
  - Time sensitive tasks
- Resources
  - Memory
  - Power
- State
  - Broken hardware
  - Within communication range
- Priority
  - External tasks
  - Control Tasks

Scheduler

- Constructs the schedule but doesn't execute it
- Processing
  - Risk Manager
  - Prioritizer
- Schedule Construction
  - Scheduling Algorithm

Scheduler – Risk Manager

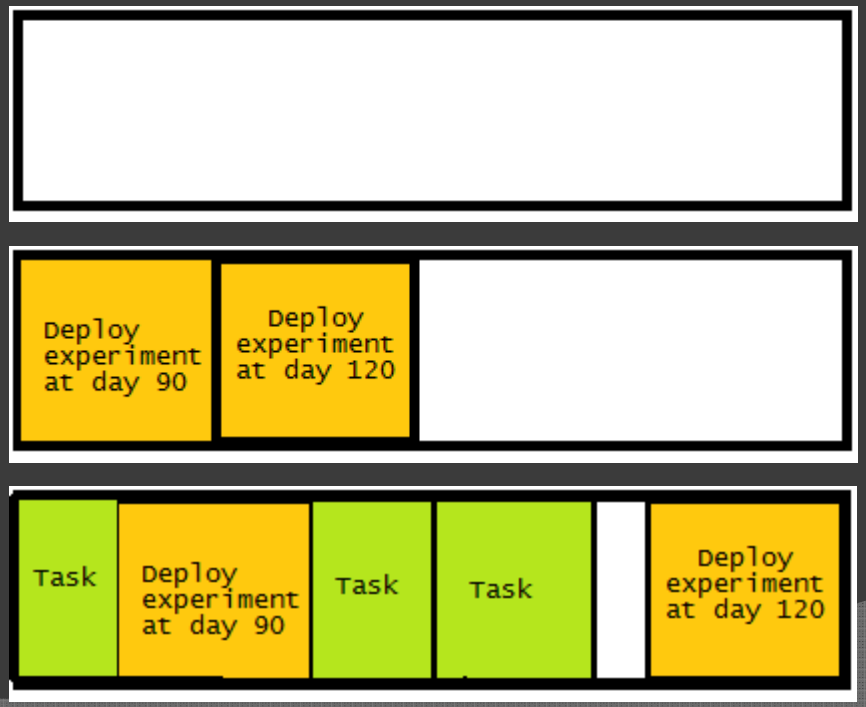
- Removes Tasks from the selection pool
  - Dangerous tasks
    - Resources
    - Probability of success (< 65%) → consequences
    - Mission status (≥ 70%)

Scheduler - Prioritizer

- Assigns a ranking to Tasks in the selection pool
  - Time - time of day
    - Absolute deadlines exit prioritizer
  - Fairness - starvation
  - Contribution to Mission - completion of tiers
  - User priority - importance to developer

Scheduler - Algorithm

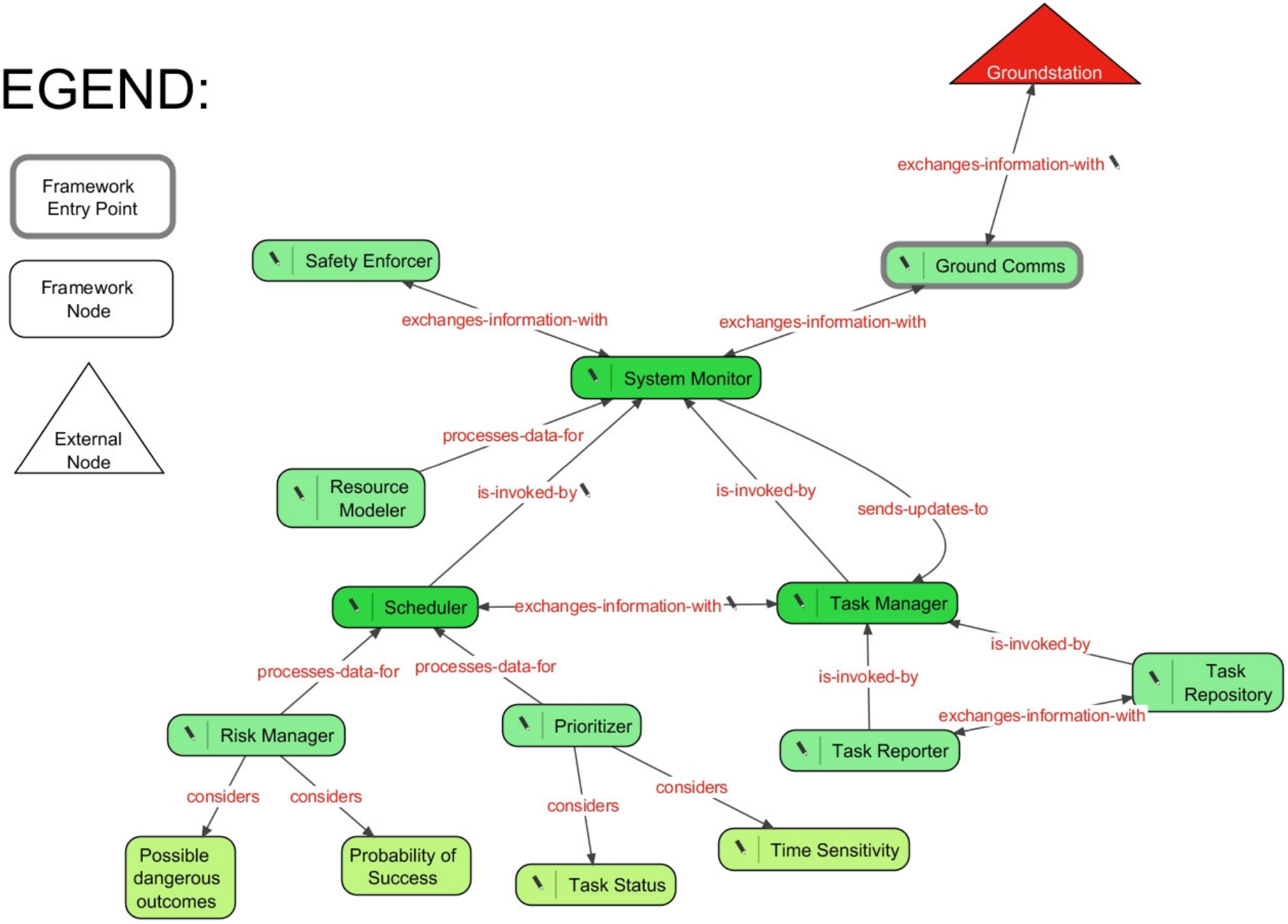
- Place Tasks with absolute deadlines first
- Place rest of Tasks in selection pool by ranking and time length



Overview

- Background
- Previous Systems
- GDMFAS
  - Data Items
  - Task Manager
  - Scheduler
  - System Monitor
- System Concerns

LEGEND:


Framework Entry Point
Framework Node
External Node
Groundstation
exchanges-information-with
Safety Enforcer
Ground Comms
exchanges-information-with
exchanges-information-with
System Monitor
processes-data-for
Resource Modeler
is-invoked-by
is-invoked-by
sends-updates-to
Scheduler
exchanges-information-with
Task Manager
is-invoked-by
Task Repository
processes-data-for
processes-data-for
is-invoked-by
exchanges-information-with
Risk Manager
Prioritizer
Task Reporter
considers
considers
considers
considers
Possible dangerous outcomes
Probability of Success
Task Status
Time Sensitivity

System Monitor

- Ensures system safety
  - Enforce manual override
  - Kill or halt a Task if it becomes dangerous
  - Force critical tasks to execute
- Monitor Resources
  - Enforce sharing of resources
  - Prevent Tasks from running due to Resource problems
- Invoke other modules
- Comms

Overview

- Background
- Previous Systems
- GDMFAS
  - Data Items
  - Task Manager
  - Scheduler
  - System Monitor
- System Concerns

System Concerns

- Performance
  - Memory
    - Unnecessary modules
    - Memory overhead
  - Complexity
  - Effectiveness
- Usability
  - Easy to understand, integrate
- Practicality of Implementation

QUESTIONS?

Sources in Paper