A Generalized timeline representation, Services, and interface for Automating Space Mission Operations

Steve Chien, Mark Johnston,
Jet Propulsion Laboratory, California Institute of Technology

Jeremy Frank, NASA Ames Research Center

Mark Giuliano, Space Telescope Sciences Institute

Alicia Kavelaars, GMV

Christoph Lenzen, DLR

Nicola Policella, ESA

Copyright 2012 © All Rights reserved.

Many (Semi) Automated Space Mission Planning Systems

- Most use a timeline based representation for operations modeling
- Most model a core set of state, resource types
- Most provide similar capabilities on this modeling to enable (semi) automated schedule generation
- In this paper we explore the the commonality of: ***representation*** & ***services*** for these timelines

A range of timeline-based systems

Timeline modeling

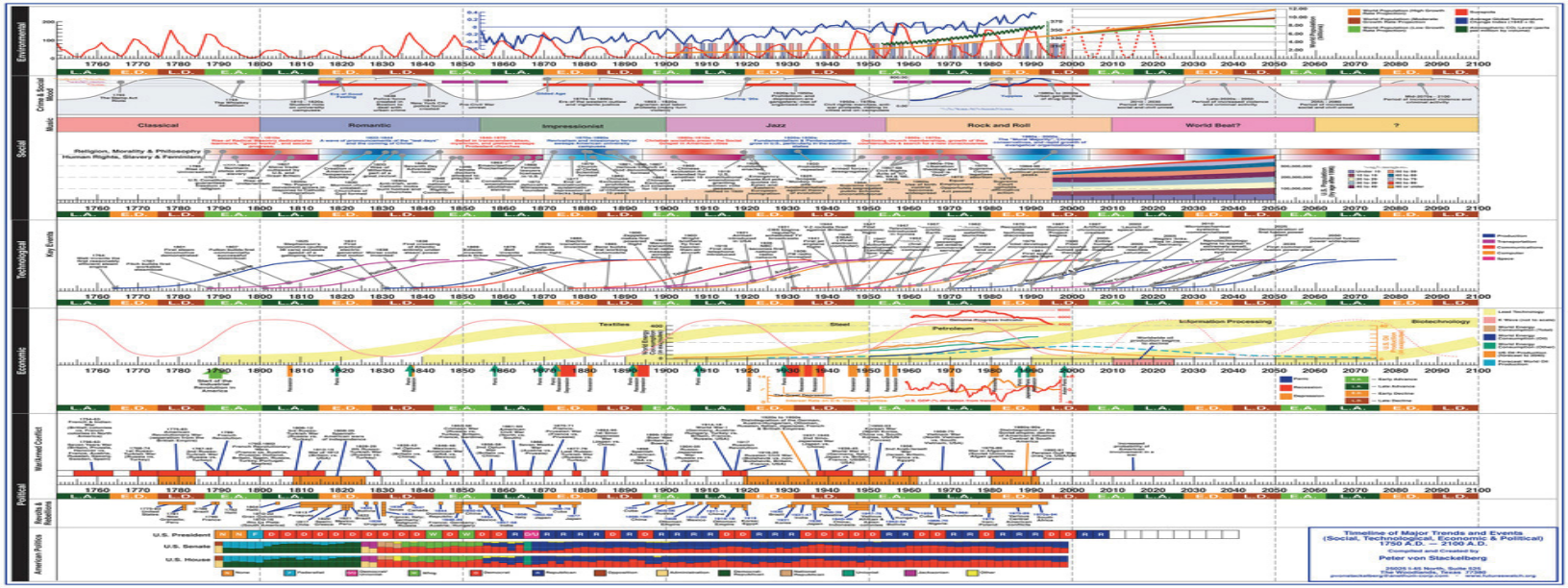
System	States, infinite	Resources	Relative timing	Parametric constraints
APSI	Finite Infinite (by means of parameters)	Yes, depletable and non-depletable	Supported	Yes
ASPEN	Finite, Infinite	Yes, unit, depletable, non-depletable, integral	Yes	Yes
EUROPA	Infinite	Yes	Yes	Yes
flexplan	Supported	Supported	Supported	Supported
Mexar2	finite states	reusable resources. Cumulative resource, and binary resource	yes	no
MUSE	Yes	Yes	No	No
Pinta/Plato	Yes	Yes	Yes	No
SKeyP	finite states	reusable resources. Cumulative resource, and binary resource	yes	no
SPIFE	Infinite	Yes	Yes	Yes
SPIKE	No	Yes	Yes	No

A range of timeline-based systems: common search interfaces

System	detect constraint violations in the above constraint types. typically O(n)	place an activity (and re model)	will a specific placement of an activity will violate the constraints	what are the valid times for which placing this activity will not violate any of the above constraints?	escape into arbitrarily coded constraint models: maneuvers, power, mobility, thermal.	Delete, move, abstract an activity and check constraints
APSI	X	X	X	X	X	X
ASPEN	X	X	X	X	X	X
EUROPA	X	X	X	X	X	X
flexplan	X	X	X	X	X	X
Mexar2	X	X	X	X	X	X
MUSE	X	X	X	X	X	X
Pinta/Plato	X	X	X		X	X
SKeyP	X	X	X	X	X	X
SPIFE	X	X	X		X	X
SPIKE	X	X	X	X	X	X

Advanced Planning & Scheduling Initiative - ESA

APSI Timelines representation



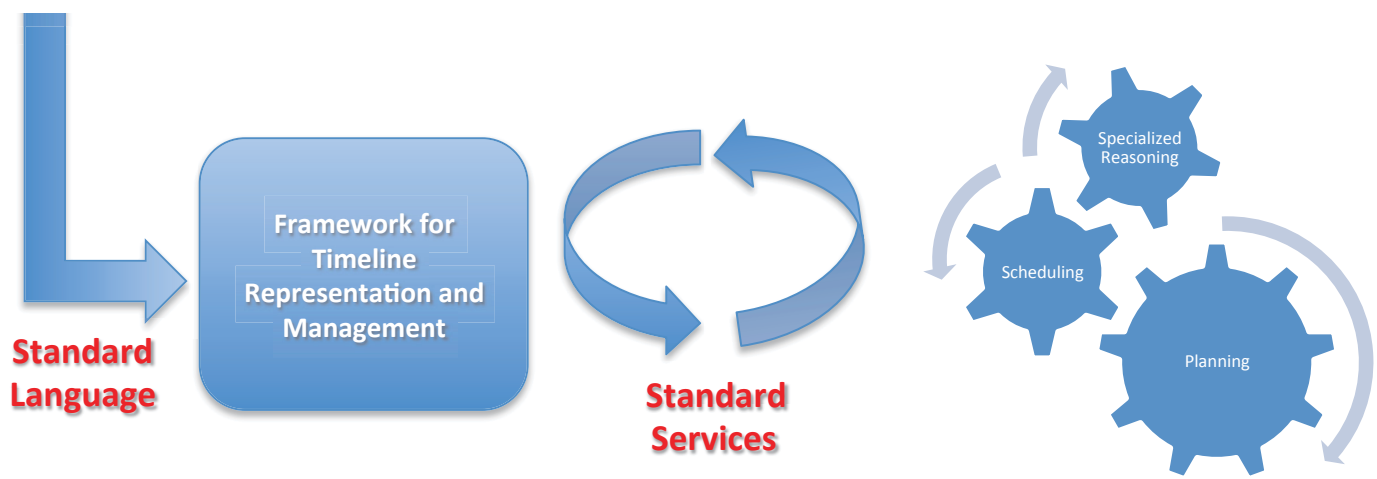
Numeric Profiles

Different levels of Abstraction

Synchronizations among Timelines

Sequences of States

- Flexible Temporal Constraints
- Cause-Effect Relationships
- Controllable /Uncontrollable Transitions
- Hierarchies
- Uncertainty
- Autonomy Levels

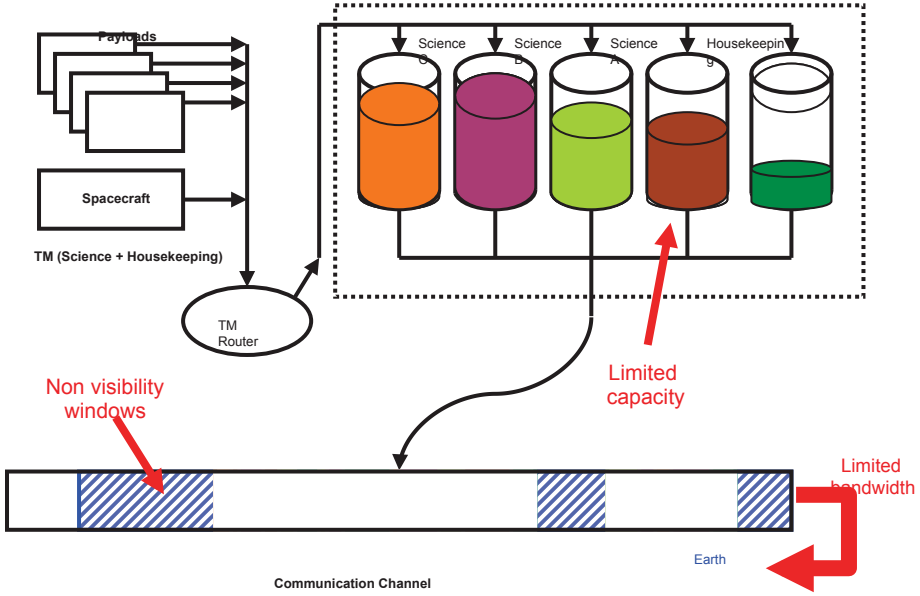


APSI Timelines representation

- Specifically the ability to support:
  - linear, possible grounded timepoints
  - finite states, infinite states, depletable and non-depletable resources
    - Finite states yes, infinite states by means of parameters, resources yes
  - variable relative time constraints
    - Not directly, we can currently state only that the bound of a constraint is not hardcoded in the domain but is an external function of the status
  - functional parametric dependencies among usages and activity parameters
  - the ability to detect conflict in the above constraints
  - to place an activity (and separately or simultaneously propagate/model)
  - the ability to query if a specific placement of an activity will violate the constraints
    - Yes, for temporal, parameter, sequence, resource and duration.
  - the ability to query for valid time placements of an activity with constraints
    - Yes, for temporal, parameter, sequence, resource and duration.
  - the ability to escape into arbitrarily coded constraint models, for maneuvers, power, mobility, thermal, etc

Mexar2

- The problem: generation of spacecraft operations for efficient on-board mass memory dumping for MEX
- The downlink activities were synthesized manually by a team of people continuously dedicated to this task
- Several constraints & requirements: limited on-board memory, limited communication capability, avoid data overwriting

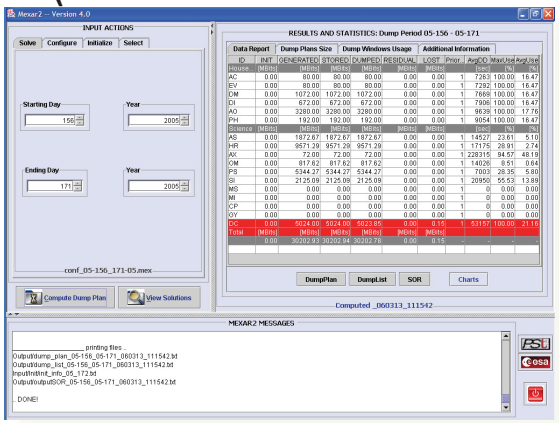


INSTITUTE FOR COGNITIVE SCIENCE & TECHNOLOGY

Mexar2

Technical features and performance

- Software design
  - Object-oriented
  - Two modules:
    - Problem Solver (PS)
    - Man-Machine Interface (MMI)
- Implemented in Java
  - Multiplatform: works under UNIX, Windows, Mac OSX
- Interactive problem style allowing what-if analysis
- **MEXAR2** is a configurable tool (e.g., adding a new packet store)
- Efficient solving algorithms (e.g., a dump plan over a period of 30 days is computed within 1 minute of computation)
- **MEXAR2** has reduced by 50% the time needed to generate dump plans
- Produces plans of higher quality without data loss (robustness)
- Allows to spot in advance resource bottlenecks (increased science return)



SKeyP

SOHO Keyhole Planner

- The problem: to generate plans for SOHO Keyhole periods operations
- Keyhole period: The HGA pointing capability, recorder dumping capabilities (possible only with DSN 34/70 m antennas) and recorder capacities are not sufficient to downlink all data,
  - selection and prioritization
- Plan :
  - What to store in the on-board memory
  - Data Downloading Activities

Requirements & Goals

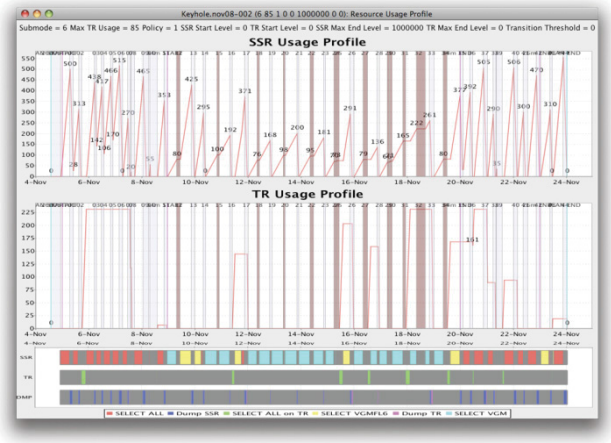
- satisfy the different constraints (e.g., finite recorder capacity, DSN antenna limitations, robustness)
- flexibility in recorder usage, switching commands timings, etc.
- allow exploration of options
- reduce planner's mechanical and repetitive tasks (and time) needed to produce a baseline solution
- reduce dependence on planner experience
- Integration with the current workflow

SKeyP

Achievements

- SKeyP solves the problem and reduces the working time
  - It produces a plan in under 10 seconds
- Rapid what-if analysis, parameter set comparisons
  - Manual fine-tuning of solutions
  - Better understanding of algorithm's behaviour
- SKeyP permits a fast handover between operational users
- It has been easily integrated with the current workflow

- Different guidelines contributed to the current result
  - Users (mission planners) integrated in the development team
  - Spiral iterative prototyping & validation cycles
  - Solved problems in compatible time constants



Automated Scheduling and Planning Environment (ASPEN)

Jet Propulsion Laboratory
California Institute of Technology

ASPEN – Application framework

- Timelines
  - States, resources, integral, *generalized**
- Constraint Engines
  - Timeline, parameter, temporal constraint
- Modeling language
  - Activity centric

Support Timeline Modeling

- Search Engines
  - Systematic, repair, user-defined*

Use Timeline Search Interfaces

ASPEN Applications

- Modified Antarctic Mapping Mission (2002)
  - Radarsat Interferometric Mapping of Antarctic Ice Sheet
- Earth Observing One (aka Autonomous Sciencecraft, Sensorweb)
  - Flight and Ground (2004-present)
  - Enabled $1M USD cost avoidance, +40% science return
- Orbital Express (2007)
- Rosetta (ongoing)
  - In development to support science planning for ESA's mission to Comet C-G

Europa

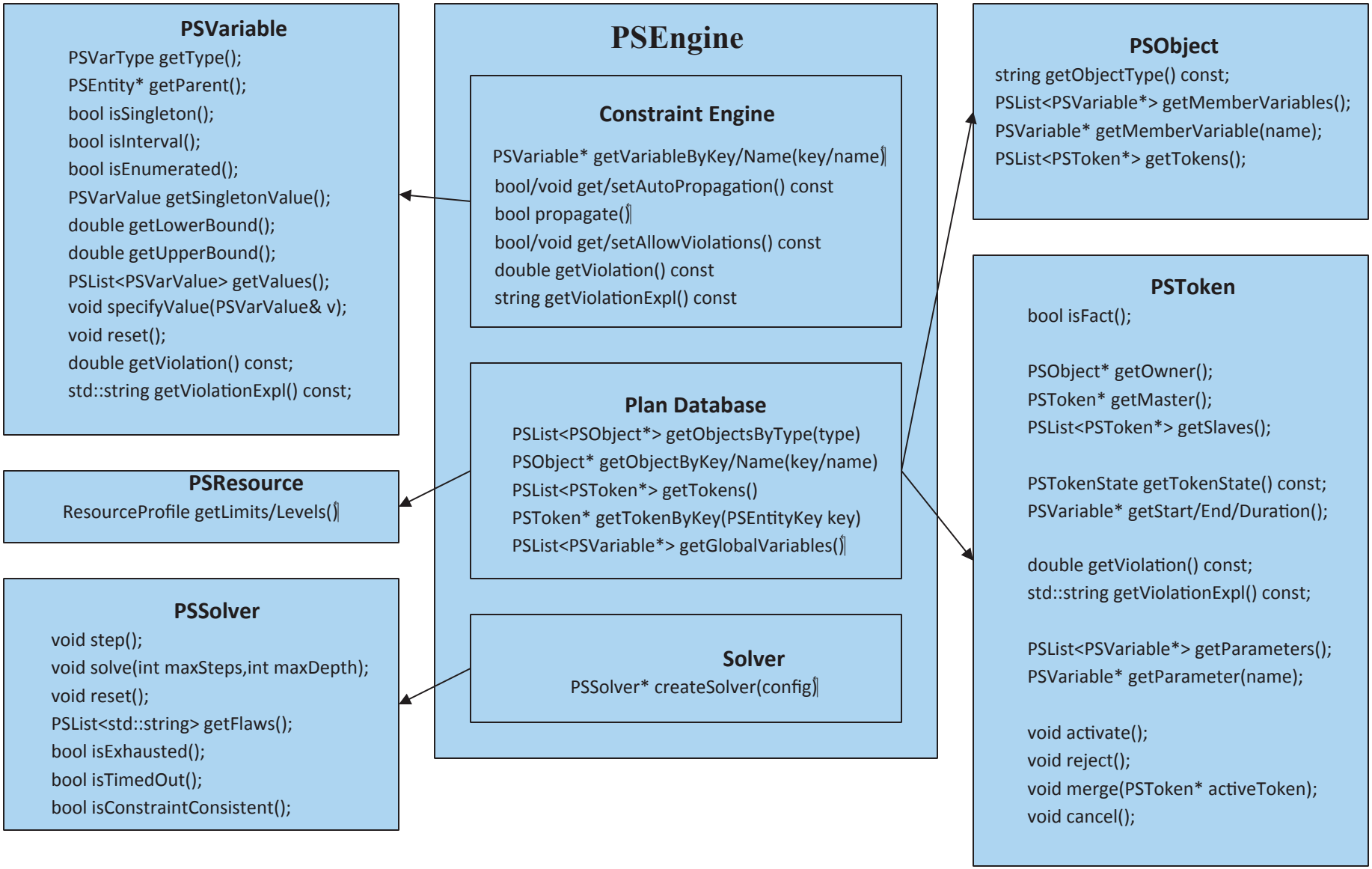
NASA Ames Research Center

EUROPA: NDDL and the Plan Database

- Classes
  - Inheritance
  - Constructors
- Objects
  - Contain Tokens, Variables, other objects
- Tokens
  - Object-scoped
  - Temporal entities (start, end, duration). Typically represent activities or states.
- Variables and Constraints
  - Object attribute, Token parameter, Global variables
  - Constraints on these
- Mapping NDDL to Reasoning
  - Temporal relations mapped to STN
  - Timelines and Resources mapped to special classes
  - Parameter constraints mapped to special classes

```
class LightSwitch extends Timeline{
predicate turnOn  { duration=1; }
predicate turnOff { duration=1; }}

class LightBulb extends Timeline {
        LightSwitch mySwitch_;
        LightBulb(LightSwitch b)
        { mySwitch_ = b;  }
        predicate On  {} predicate Off {}}

LightBulb::On{
        // Bulb must be Off to be turned On
        met_by(object.Off);
        met_by(object.mySwitch_.turnOn);}

LightBulb::Off{
        // Bulb must be On to be turned Off
met_by(object.On);
        // Must be turned off through the switch
met_by(object.mySwitch_.turnOff);}

goal(LightBulb.Off g1);
eq(g1.start,10);
```

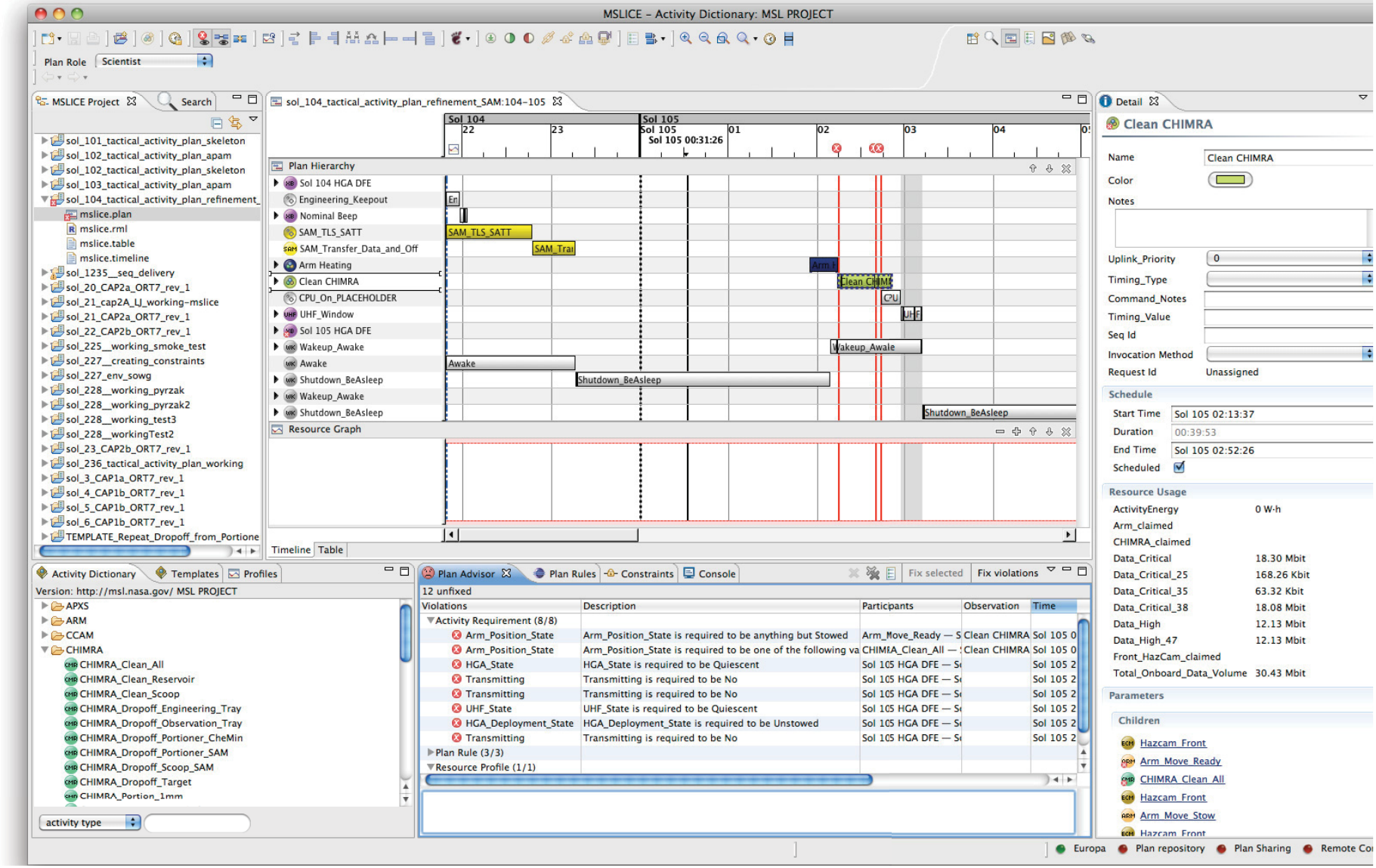
EUROPA: Architecture and Interfaces



EUROPA - Applications

- MAPGEN (MER)
  - *High degree of automated search*
  - Goal decomposition
  - Temporal reasoning
  - Reasoning about state conditions
- Phoenix Science Interface (PSI)
  - *Low degree of automated search*
  - Temporal reasoning
- **SACE (ISS Solar Array Planning)**
  - *High degree of automated search*
  - Temporal reasoning
  - Activity decomposition
  - Reasoning about state conditions
  - Optimization
- **T-REX**
  - *High degree of automated search*
  - Goal decomposition
  - Temporal Reasoning
  - Reasoning about state conditions

Ensemble – Architecture



Ensemble – Domain Description and Reasoning

- Activity Dictionary
  - Internal representation
    - Eclipse Modeling Framework (EMF)
    - Javascript formulas
  - Description of activity, activity details, and timelines
  - Description of conditions
  - Description of resources
  - Description of relationships
- Plan Advisor
  - Temporal reasoning built in
  - Visualization of feedback from external reasoners
    - E.g. EUROPA, Power analysis engines
  - External reasoners perform all checks of ‘Timeline’ conflicts
    - E.g. two activities that can’t overlap, condition checks

Ensemble - Applications

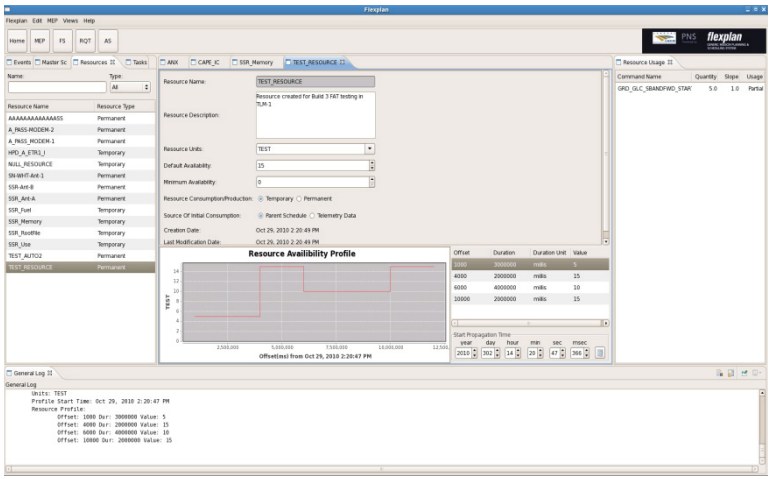
- Phoenix Science Interface (PSI)
  - *Low degree of automated search*
  - Temporal reasoning
  - Reasoning about state conditions
- MSLICE
  - *Moderate degree of automated search*
  - Temporal reasoning
  - Reasoning about state conditions

flexplan

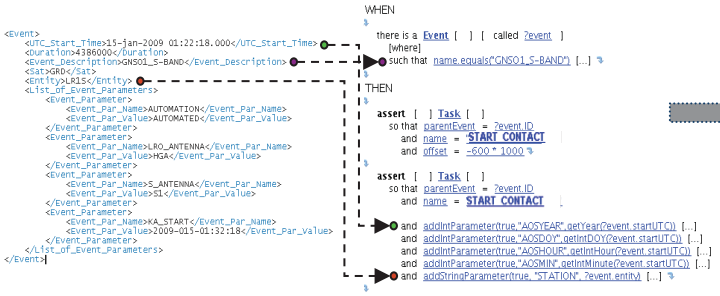
GMV

TIMELINE GENERATION IN *flexplan*

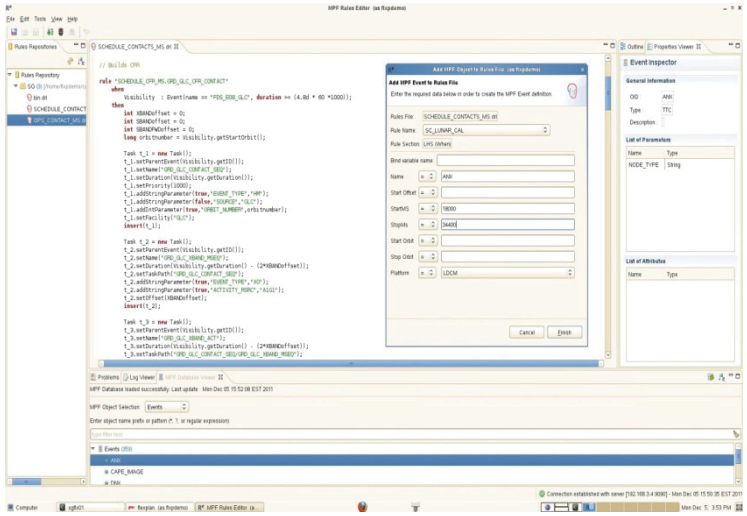
- Mission configured by defining 4 key elements:
  - Resources: Space, Ground, Manpower …
  - Events: FDS, Instrument, Geometry, Service Request Inputs
  - Tasks: Schedule Activities, Hierarchical Levels
  - Rules: Logic Based rules that combine Resources and Events to generate Tasks or detect Mission Constraints/Conflicts (Temporal, Resource and Limit Violation Constraints)
- Two modes of schedule timeline generation:
  - Logic Programming Rules: User configurable through a Rules Editor
    - Guided rule generation
    - Database integration
    - Syntax and consistency validation
  - Optimization Algorithms: Code based algorithms for more complex scheduling optimization



Resources, Events and Task Configuration



Events folded into Logic Programming Rules

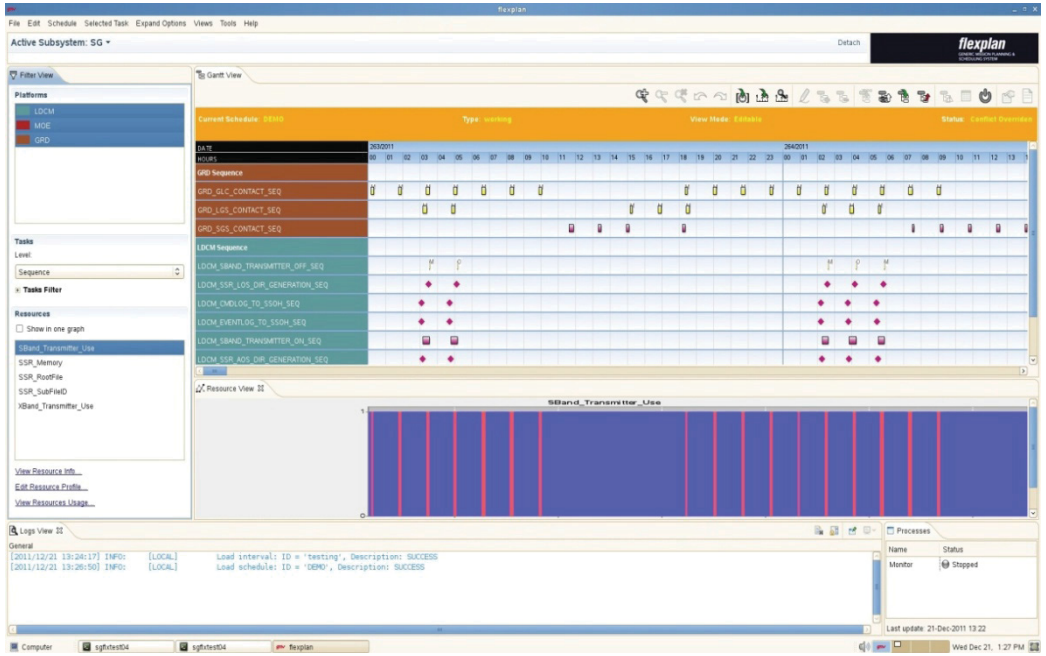


Rules Engine Editor

TIMELINE REPRESENTATION IN *flexplan*

- Each Task (Schedule Activity) represented in the Timeline, Task duration and exclusion windows shown
- Timeline Tasks classified by Mission Platforms: Spacecraft, Ground Site, Operations Center …
- Timeline Visual Cues Available: Zoom, Slide, Time Toggle (UTC, Spacecraft Time), Double-click to access Task Details
- Timeline Filtering by: Mission Platform, Task hierarchical level
- Resource Availability vs. Consumption Filtering and Representation in line with timeline
- Conflict Detection and Conflict Resolution Capability embedded in Timeline

- Manual Timeline Interaction
  - Task Insertion
  - Task Move or Deletion with Constraint Checks and Conflict Detection and Resolution
- Automated Timeline Generation
  - Full Automation/Lights Out operation
  - Scheduling, Constraint Detection and Conflict Resolution Rules and/or Optimization Algorithm applied to generate Conflict free Timeline



flexplan Timeline Representation

flexplan MISSIONS

Mission	Customer	Start Date	Automation
MetOp	EUMETSAT/ NOAA	2006	Automated input ingestion and processing. Status indicators in timeline indicate Activity Execution Status
LRO	NASA	2009	Automated input ingestion and processing. CCSDS Command Loads generated directly by ***flexplan***. Parameters fed back to timeline for inclusion in next Command Load.
SMOS	ESA	2009	Automated input ingestion and processing.
LDCM	NASA/ USGS	2013	Automated input ingestion and processing. Automated timeline generation based upon command directives. Automated conflict checks. Automated timeline output products export.
Sentinel 1	ESA	2013	Full automation the entire timeline lifecycle, from input ingestion and processing to timeline generation, conflict detection, resolution and output product export.
Sentinel 3	ESA/ EUMETSAT	2013	Automated input ingestion and processing.
Paz	INTA	2013	Automated input ingestion and processing. Automated output file generation.
SGSS	NASA	2015	Full automation the entire timeline lifecycle, from input ingestion and processing to timeline generation, conflict detection, resolution, timeline optimization and output product export.
Ingenio	CDTI/ESA	2015	Full automation (see Sentinel 1).

Global Team

- Primary ***flexplan*** teams in USA and Spain
- Systems deployed in both NASA, ESA and other Institutions/Agencies

MUSE

Jet Propulsion Laboratory
California Institute of Technology

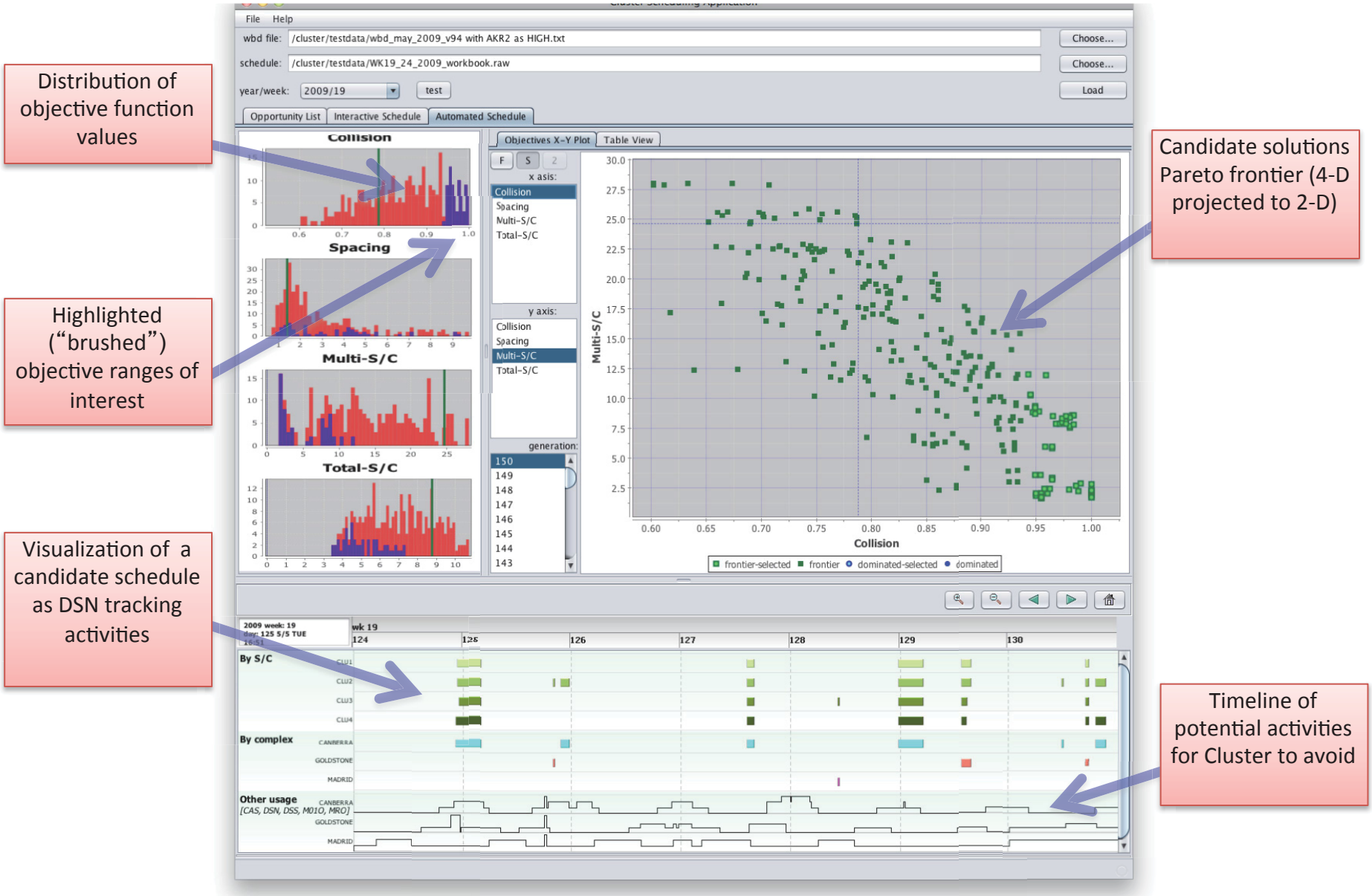
MUSE – timelines

- MUSE timelines
  - Emphasis is on support for multi-objective schedule optimization
  - Represented in code (Java) as piecewise-constant or -linear valued function of time
  - Java interfaces provide query functionality, algebra and set operations and various other timeline manipulation
  - Model elements include
    - Activities and groups of activities
    - Resource availability and consumption
    - Constraint and preference intervals and values
  - Objective and constraint evaluation functions are typically phrased in terms of timeline APIs for use by multi-objective optimizer

MUSE – usage

- MUSE is infused operationally for Cluster II WBD scheduling
  - Objectives considered include
    - Avoiding resource collisions with other mission activities
    - Spacing
    - Multiple-spacecraft activities preferred
    - Science activity priority mix preferences
  - Interactive GUI for trade-off exploration
- Other MUSE adaptions
  - JWST science planning to optimize for angular momentum accumulation, min gaps, min missed science opportunities
  - Cassini science planning with onboard recorder capacity constraints

MUSE – Cluster II WBD GUI



Pinta/Plato

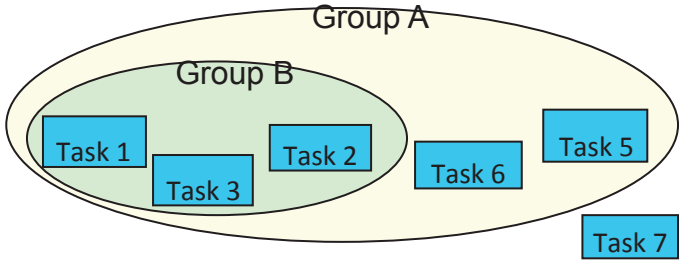
DLR/GSOC

Planning Modeling Language at DLR/GSOC (1)

Hierarchical structure:

Tasks may be given timeline entries

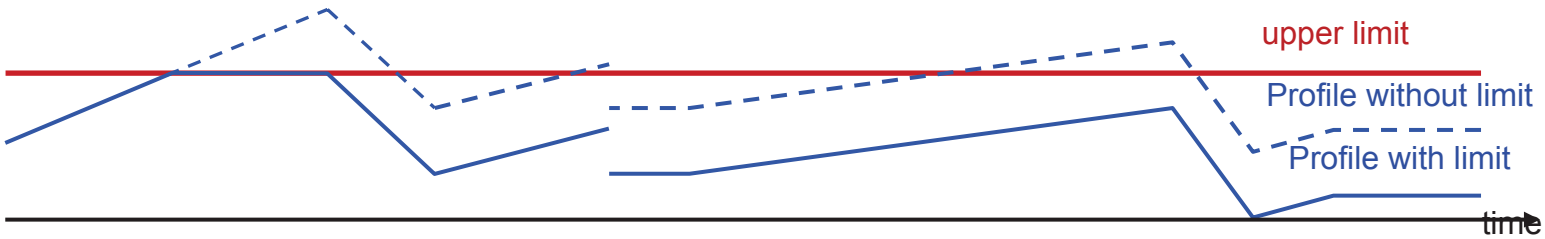
Groups contain Tasks and other Groups



Resource:

Resource profile (maps time to value)

Optional: Resource limits (resource profile is cut off at the limit)

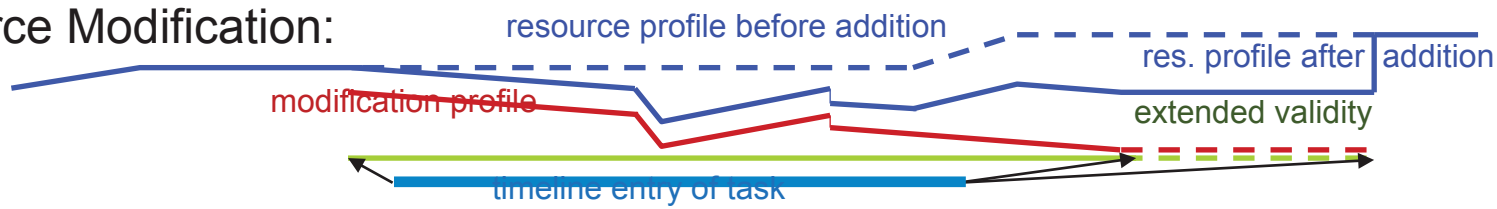


- Constraints (1):
  - Task: min/max duration
  - Group: min/max number of elements to schedule
  - Time Dependency: min(/max) separation in between tasks‘ start/end times

Planning Modeling Language at DLR/GSOC (2)

- Constraints (2):
  Resource Comparison:

Resource Modification:



Suitability: prefer timeline entries where a certain resource is maximal

Interfaces of Pinta/Plato

API of Plato's class library (currently .NET) supports heuristic algorithms
(calculate possible scheduling times, conflict traces, etc.)
Generic algorithm of Plato supports heuristic search
Plugin architecture of Pinta for import and export

Missions at GSOC/DLR using Pinta & Plato

- TerraSAR-X/TanDEM-X (since 2007/2010)
  - Fully automated system, generating 2 timelines per day
  - Custom algorithm implemented on top of Plato
  - Schedules payload commands and routine bus commands, excl. maneuvers
  - All scheduling relevant constraints covered by the planning model, including power, thermal windows, inter-satellite constraints
  - Pinta used for manually checking the timeline (during commissioning, anomalies)

- TET (2012)
  - Generic repair algorithm manually applied via Pinta on given payload timeline
  - generic algorithm schedules bus commands

- OnCall shift scheduling
  - Schedules the multi-mission staff for 24/7 on-call support
  - Generic algorithm for plan generation and major modifications
  - Manual interaction using Pinta for minor modifications

- Upcoming:
  - BiROS: payload and bus commands, support of on-board automation
  - EuCROPIS: bus commands
  - TerraSAR-2: payload and bus commands
  - EnMAP: payload and bus commands

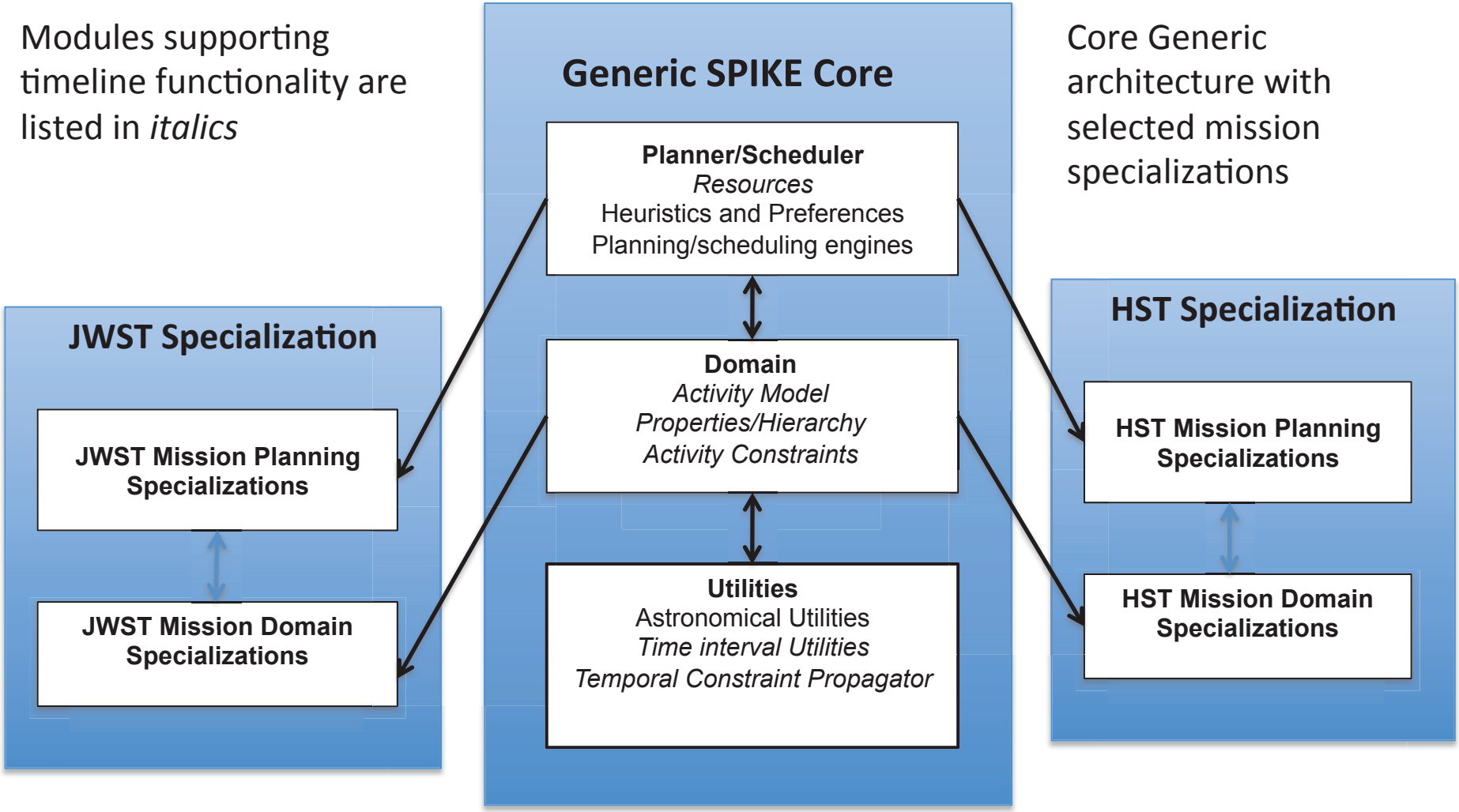
Spike

Space Telescope Sciences Institute

SPIKE High Level Architecture

Modules supporting timeline functionality are listed in *italics*

Core Generic architecture with selected mission specializations



SPIKE Time Line Support

- The temporal constraint propagator module provides most timeline support for SPIKE
  - Definition of tasks and constraints
    - Including simple and disjunctive constraints and spacecraft roll constraints
    - Can specialize constraint definitions for trade offs between dynamic changes of constraint parameters and runtime
  - System automatically maintains sets of legal time intervals for tasks
- SPIKE object model provides for modeling astronomical activities and planning states
- Scheduler module handles resources

SPIKE Missions Supported

- The SPIKE system achieved an over 30% increase in observation utilization for the Hubble Space Telescope.
- SPIKE has been used in multiple orbital and ground based astronomical missions including:
  - FUSE, Chandra, Subaru, and Spitzer.
- SPIKE has been re-factored for use in JWST long range planning.
- The TRANS system allows astronomical scientists to efficiently utilize the Hubble Space Telescope by planning how exposures schedule in abstract orbits.
  - TRANS uses the same core timeline utilities as SPIKE.

Summary

- Many (semi-) automated mission planning systems for space mission operations
- Many of these use a core set of timeline based modeling functions
  - Timeline modeling
  - Search services
- These commonalities offer potential to be harmonized to enable interoperability, re-use