Benefits for the standardization of Mission Planning Services from Advanced Planning Experiences

Simone Fratini, Mehran Sarkarati, Nicola Policella, Alessandro Donati, Mario Merri and Mariella Spada

22/03/2013

Background

1. IWPSS-11 on Prospects for Common Timeline representation
    a. SpaceOps 2012 paper

2. CCSDS Meeting on “Mission Planning Service Standardization”
    a. Darmstadt, summer 2012
    b. Chicago, autumn 2012

Conclusions (of the CCSDS Meeting)

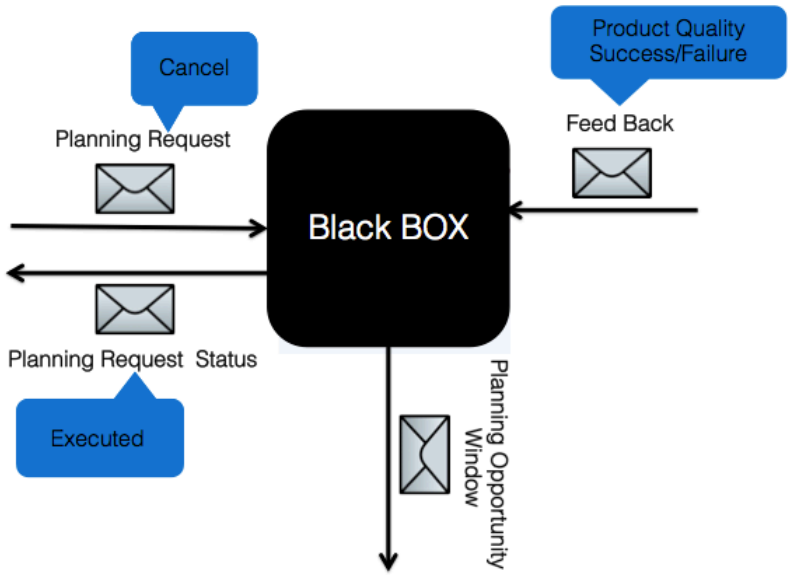
1. Is there a need to standardize Planning Services?
    a. *Probably yes*

2. If so, which ones (what information exchange or operations requests are candidates for standardization)?
    a. *At least low-level services for managing timelines*

3. What are the organizations/systems that would be providers/consumers of standard Mission Planning services?
    a. *Space agencies and 3rd parties software providers*

4. Are you aware of any pre-existing standards or information models that could be used as a starting point?
    a. *Yes, almost 20 years of deployments with timelines*

The Black Box



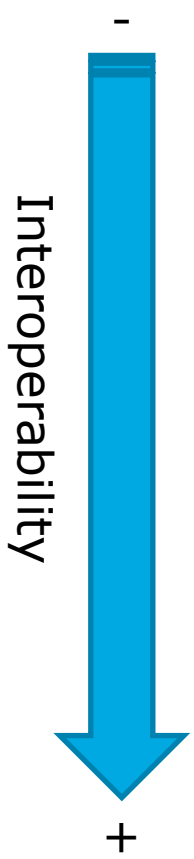
1. CCSDS recommended to start from the boundaries of the planning system (seen as a Black Box)

2. Handling planning "requests"

Outline

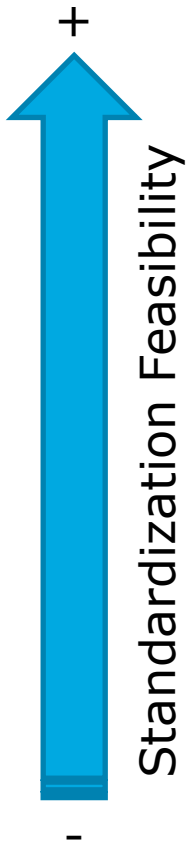
1. Analysis of the boundaries
2. Space planning experiences: current solutions
3. A Planning Service Oriented Architecture
    a. Categories of services
    b. Possible services
4. Conclusion

Framing the boundaries



1. All relevant semantics for planning is defined within the black box, we standardize only the syntax of the messages

2. The interfaces of the black box allow to semantically describe data, problems and processes



A middle point ?

(possible) agreement on the *syntax and semantic of a* ***limited set of services*** provided by the system to **manipulate** low level information:

- More effective than simply syntax…
- …less difficult than agree on a language or pre-defined semantic

- Do we need to reinvent the wheel? Probably not.

Advanced space planning experiences (not complete)

NASA AMES

- HSTS
- EUROPA
- MAPGEN

HST/JWST

- SPIKE

JPL

- ASPEN
- MUSE

ESA

- MEXAR2, RAXEM, SKeyP
- APSI (MrSpock, AIMS)

DLR

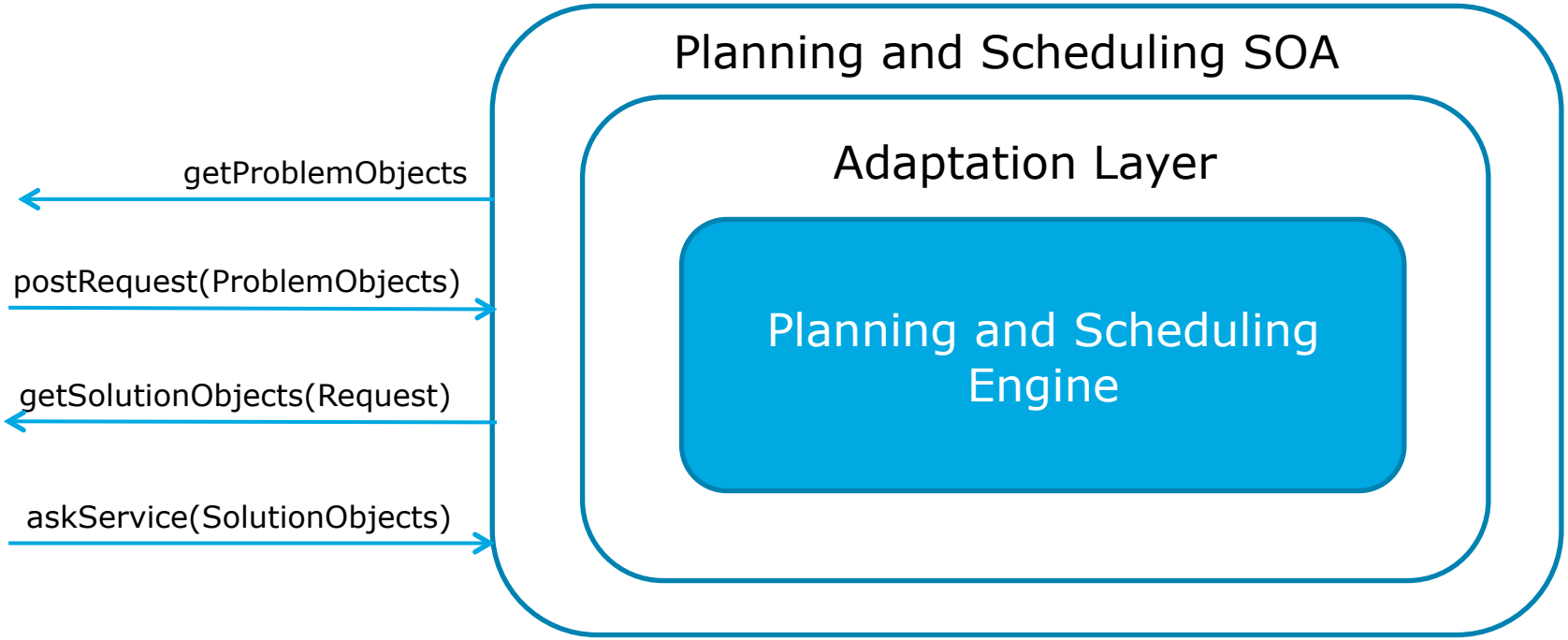
- TerraSAR-X/TanDEM-X MPS

Similar Approach

1. Same Background:
    a. Model-Based (symbolic entities, software deployment and test substantially independent from the specific mission)
    b. Timelines (short cognitive distance between modeling primitives and the objects to be modeled in space problems)

2. Similar set of services for (a subset of):
    a. Represent and Manage Activities and Timelines
    b. Define Domain Theories, Problems and Solutions
    c. Support Problem Solving with Timelines
    d. Support Timeline Validation and Verification
    e. Entail Timeline Execution

Planning Service Oriented Architecture



1. Define classes of basic services
2. Define levels of service for each class
3. Define dependencies among service levels

Classes of Services

(1) services to represent and manage the basic entities that constitute a planning problem and its solution:
    (1) Timelines
    (2) Tasks, Events and Activities
    (3) Constraints and Rules
    (4) Objective Functions

(2) services to interact with the system, to post problem, control the solving process and provide feedback to the system:
    (1) Timeline Extraction
    (2) Scheduling
    (3) Planning
    (4) Optimization

(3) services to use and manage solutions:
    (1) Storage
    (2) Management
    (3) Visualization
    (4) Execution

Service to represent the basic entities (1/3)

1. Requirements:
    a. Independent on the specific syntax for representing data, constraints and rules
    b. No assumption on specific language (NDDL, ANML, DDL and so on)

2. Needs an agreement on few concepts:

    a. **Problems** as sets of time tagged data and relations to be satisfied among them (Temporal, Data and Time-Data Relations):
        - Punctual: data + time point
        - Interval: data + time interval
        - Complex: data + implicit definition of time tags (i.e. every 2hrs)

    b. **Solutions** can be either:
        - Set of time tagged data
        - Set of timelines
        - Constraint Network on time and data variables

Service to represent the basic entities (2/3)

1. Timelines as sequences of time-tagged data:
    a. Time Model: Fix, Bounded, Flexible
    b. Data Type: Ground, Parameterized, Multi-valued, Functions
    c. Transition Constraint: Local, Global
    d. Specification: Complete, Partial

Complexity/Representational Power

Better to keep it as general as possible, i.e. at least multi-valued flexible timelines (then you can use time fix ground valued timelines for instance)

Service to represent the basic entities (3/3)

1. Temporal relations among time tagged data:
    a. Simple precedence
    b. Overlapping/Not Overlapping
    c. Qualitative General Temporal Relations (e.g. Allen's algebra)
    d. Quantitative that does not require search (e.g. STP)
    e. Quantitative that requires search (e.g. DTP)

2. Data Relations
    a. Simple Equality
    b. Equality/Inequality
    c. Set Relations
    d. Linear/Not Linear
    e. Conjunctive
    f. Disjunctive
    g. Global Constraints
    h. …..

Services to interact with the system (1/2)

1. Increasing levels of reasoning complexity:
    a. Timeline extraction & querying (no domain theory)
    b. Scheduling (no new activity generation)
    c. Planning & Scheduling (activity ordering and generation)
    d. Optimization

2. Timeline management, extraction & querying basic services:
    a. propagating constraints to query the value of the timeline at a specific time
    b. querying the status of the timelines
    c. detecting and reporting conflicts in the constraints
    d. extracting timelines from tasks, events, activities (at various levels of flexibility)
    e. querying if a specific placement of an activity/task/event will violate the constraints
    f. adding/retracting dynamically activities/tasks/events from timelines
    g. synchronizing timelines with events representing triggers or tasks/activities representing external inputs

Services to interact with the system (2/2)

1. Scheduling Services:
    a. Activity Ordering
    b. Activity Selection/Instantiation

2. Planning & Scheduling Type:
    a. State Variables (purely causal)
    b. State Variable and Renewable Resources (monotonic)
    c. State Variables and Consumable Resources (not monotonic)

3. Planning & Scheduling Output:
    a. Single Solution (Time Fix, Ground Timeline)
    b. Kernel (Only solutions)
    c. Envelope (Boundaries)

4. Planning and Scheduling Process Boundaries (and authorities):
    a. Optimization (in the kernel or envelope)
    b. Selection (in the envelope)
    c. Refinement
    d. Generation (in an envelope)
    e. Re-Scheduling
    f. Re-Planning

Services to use and manage solutions

1. Evaluate solutions (quality, robustness, flexibility, stability)
2. Executing timelines (monitoring the process):
    a. different levels of control/autonomy
3. Storage
4. Visualize timelines (less trivial than it looks like)
5. Intersect, synchronize, merge and split timelines
6. Support problems at various level of granularity and temporal horizon

Example of Service Level Requirements

1. An RCPSP-Max Scheduling problem would require:
    a. Activities (Interval data Problem Objects or more)
    b. Set of time tagged data as solution objects (or more)
    c. Quantitative Temporal Constraints (STN or more)
    d. Scheduling Capabilities

2. Planning for dumping with ground station visibilities would require (at least):
    a. Interval Data as Problem Objects
    b. Timelines as Solution Objects (time fix or more)
    c. Quantitative Temporal Constraints
    d. Planning Capabilities with State Variable and Renewable Resources (to model the memory allocation)

3. Planning for On Board Autonomy would require:
    a. Planning with State Variables (or more)
    b. Re-Planning Capabilities (requires an Envelope-Level Planning Solution)
    c. Dynamic insertion of activities into timelines (requires time-flexible timelines)
    d. Timeline Execution Services

Possible roadmap

Modelling Services	Planning Objects (Events, Activities, Tasks)	Problems (Goals, Objective Functions)	Solutions (Time Tagged Data, Timelines, Constraint Networks)	Timelines (Time Fix or Flexible, Ground or Multi Valued)	Domain Theories (Constraints, Rules, State Variables, Resources)
Problem Solving Services	Timeline Extraction and Querying	Planning and Scheduling	Re-Scheduling and Re-Planning	Optimization	Solving Process: Single Solution, Kernel, Envelope
Solution Management Services	storage	Evaluation Validation and Verification	Execution	Visualization	Intersect, synchronize, merge and split solutions

Conclusions

1. Standardize Services Data and Processes to Entail Interoperability
    a. Modelling Services
    b. Solving Problem Services
    c. Solution Management Services

2. Planning Service Oriented Architecture
    a. Define Classes of Service, Level of Service and Service Requirements

3. Use an Extensible Approach, start from simple things, evolve it
    a. Analyse current status
    b. Abstract Common Features
    c. Define General Class of Services
    d. Iterate and evolve the set of services