



Martin Kolombo, Martin Pecka, **Roman Barták**  
(Charles University in Prague, Czech Republic)

## AN AD-HOC PLANNER FOR THE MARS EXPRESS MISSION

## The Problem

The task is to automate the whole planning process  
 proposed as a challenge problem for ICKEPS 2012

### Given input:

- spacecraft orbital events (flight dynamics),
- ground station visibility,
- a set of Payload Operations (PORs).

Generate a schedule of all operations for:

- maintenance,
- observations,
- command uploading,
- data downloading,
- pointing transitions.



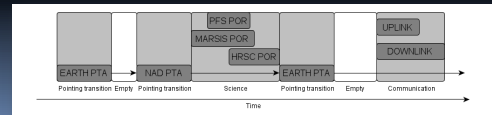
## Background



- ESA **Mars Express Mission** (MEX) launched in 2001
- spacecraft is orbiting around Mars since January 2004, producing 2-3 GBit of scientific data on a daily basis
- 7 scientific payloads** aimed at collecting data to study the Martian atmosphere and the planet's structure and geology:
  - ASPERA - Energetic Neutral Atoms Imager (AS)
  - HRSC - High-Resolution Stereo Camera (HR)
  - MARSIS - Mars Advanced Radar for Subsurface and Ionosphere Sounding (MI)
  - OMEGA - IR Mineralogical Mapping Spectrometer (OM)
  - PFS - Planetary Fourier Spectrometer (PS)
  - SPICAM - UV and IR Atmospheric Spectrometer (SI)
  - VMC - Visual Monitoring Camera (VM) - not any more in use

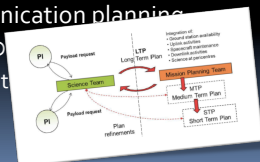
## The Core Model

- a **scheduling problem**
  - scheduled operations known in advance
  - only pointing operations are planned dynamically
- the schedule represented as a sequence of time windows containing particular (overlapping) operations - **TimeLine**



## The Situation

- current MEX planning
  - a collaborative problem solving process between a **Science Team** located at ESA-ESAC (payload requests) and the **Mission Planning Team** located at ESA-ESOC (spacecraft operation)
- automation of communication planning
  - MEXAR 2 - scheduling Data
  - RAXEM - scheduling Data



## TimeLine Modifications

- starting with a single empty window over the planning horizon
- when operations are added, the windows are **shifted** (shrunk/stretched) in time and **split up**



## The Scheduling Approach

### pre-scheduling

- distribute the **maintenance operations** uniformly to windows

### scheduling PORs

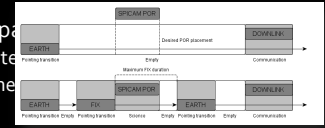
- find a window for an **observation operation**
- find windows for the **supporting operations** (command uplink and data downlink)
- add operations for **changing pointing** if necessary

### post-scheduling

- schedule the **ASPERA payload**
- schedule **auxiliary telecommands** and their uplink

## Scheduling Pointing

- Three types of pointing operations
  - NADir (time limited)
  - FIX (inertial) (time limited)
  - EARTH



### a new pointing operation

- insert right before the "main" operation that requires a different pointing (shift/split up the windows if necessary)

### a couple pointing operation (may be empty)

- insert after the "main" operation (shift/split up the windows if necessary)
- to restore the original pointing or
- to ensure the change of pointing for the time-limited pointing

## Scheduling Observations

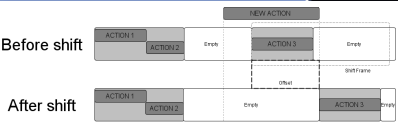
### find a window for the operation

- take the POR operations one by one
- explore windows from left to right
- allocate the operation to the selected window (shift/split up the windows if necessary)
- schedule auxiliary operations (communication, pointing)

### in case

#### Before shift

- ba
- wi



#### After shift



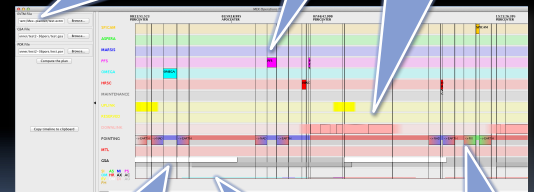
## Implementation

### Specification of input files:

- flight dynamics
- groundstations
- PORs

### Observation operations

### Communication operations



### Groundstation visibility and bitrates

### Memory allocation

### Pointing operations

## Scheduling Communication

when the observation (POR) operation is allocated to a window we need to schedule:

- operation to **uplink telecommands** (before the POR op)
- operation to **download data** (after the POR op)

### find a communication window

- the operation fits to the window (shift/split up the windows if necessary)
- groundstation bitrate defines the operation duration

### or make a new communication window

- made from an empty window (shift/split up the windows if necessary)
- respect the groundstation visibility
- may require adding new pointing operations

## Final Remarks

- Problem specification
  - **inaccuracies** in the description and data
- Implementation from scratch
  - **hard modifications**
- **Efficiency seems good** for under-constrained problems
  - not yet tested on real data
- **Generalization** to other problems?
  - YES in principle, NO in implementation
- **Future steps?**
  - test with real data
  - a new solver based on a CSP

