

# EUROPA

# Knowledge Engineering Tools



JAVIER BARREIRO\*, MATTHEW BOYCE\*, MINH DO\*,  
JEREMY FRANK<sup>†</sup>, MICHAEL IATAURO\*  
TATIANA KICHKAYLO<sup>‡</sup>, PAUL MORRIS<sup>†</sup>, JAMES ONG<sup>+</sup>,  
EMILIO REMOLINA<sup>+</sup>, TRISTAN SMITH\*, DAVID SMITH<sup>†</sup>



# Lots of stuff – little time



We will be at the System Demo for further questions!  
**Javier Barreiro – EUROPA Lead will be there!**

EUROPA Invited Talk @ PSTL Workshop

**June 26 (Tomorrow) @ 10:00 AM**

# What is EUROPA?



- Class library and tool set for building, customizing, and analyzing planners
  - Constraint-based Temporal Planning
  - Lifted Partial-Order Planning framework
  - Timeline-based state/plan representation
- Mainly used for NASA applications
- NDDL modeling language
- Open-sourced

# EUROPA History



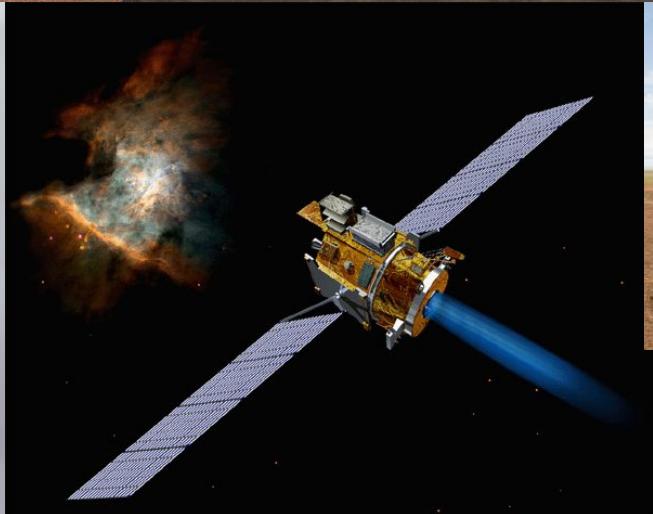
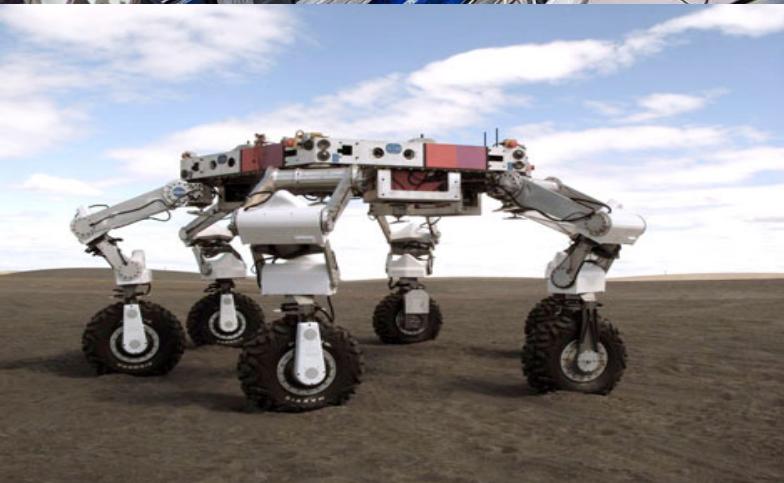
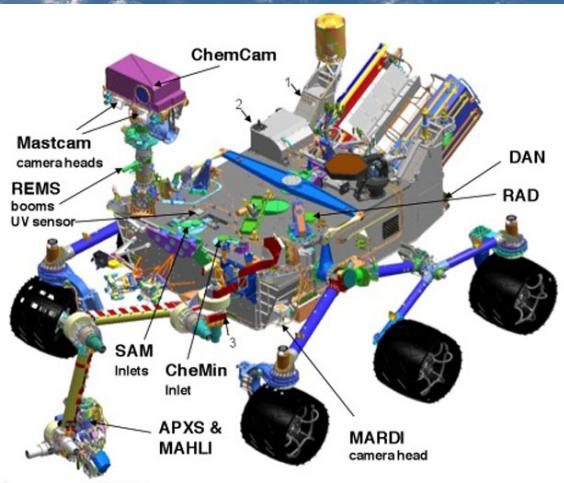
- HSTS (1998) - Initial DDL and planning paradigm.
- RAX (1999) – first deployment
  - RAX won NASA software-of-the-year
- EUROPA (2002) - Initial implementation of current approach
- EUROPA 2 (2005) - Modular architecture, more robust implementation
- EUROPA 2.1 (2007) - **Open source**, architecture evolution, moved to Google Code, strong documentation
- EUROPA 2.6 (2012) – API, KE tools

# Applications



- 15+ NASA Applications, such as:
  - Science activity planning
    - Airborne observatory SOFIA
    - Remote Agent Experiment (RAX)
    - Mars Exploration Rovers(MER)
    - Phoenix Mars Mission.
  - SACE
  - Planning and scheduling support for experiments
    - Life in the Atacama (LITA)
    - Bed Rest study.
- MBARI: underwater autonomous vehicle
- Willow Garage: autonomous robot navigation
- Other applications noticed in the EUROPA mailing list

} Ex-Ames Members



# EUROPA Architecture

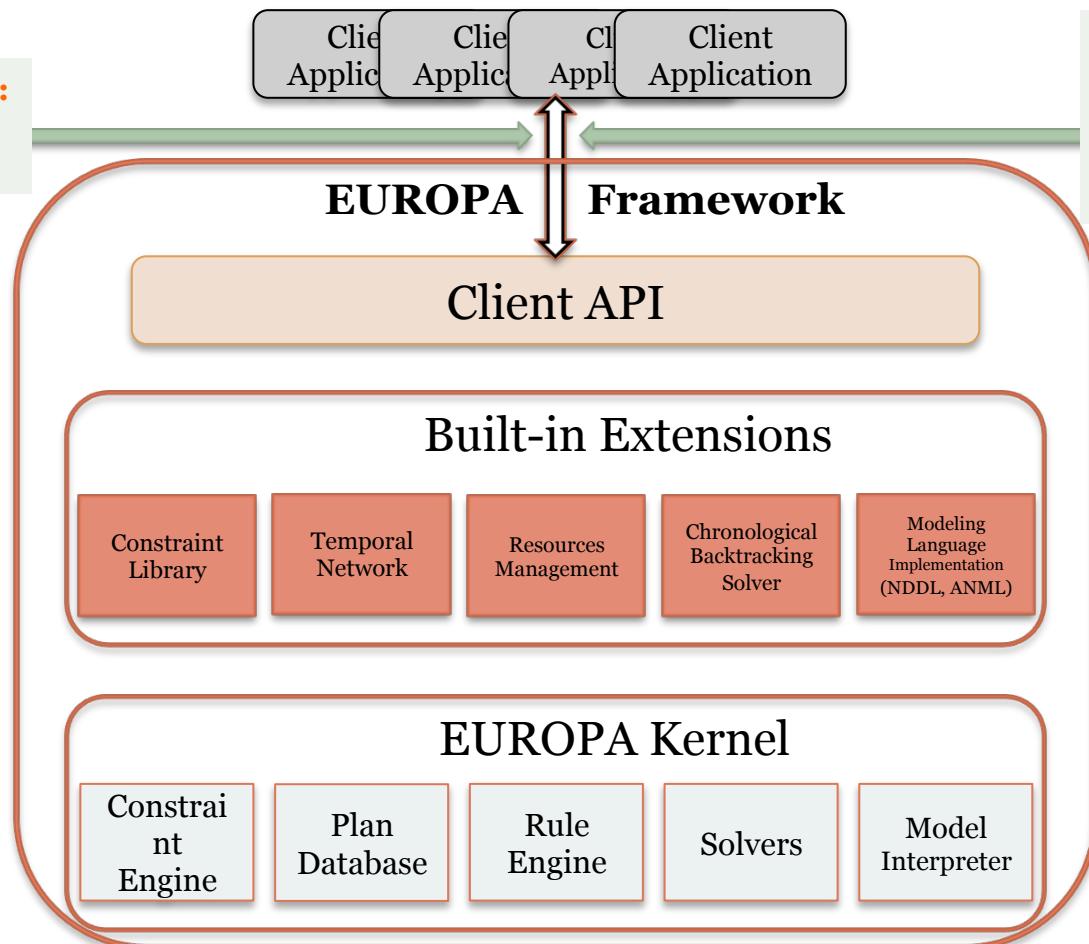


## KE Tools - Analysis:

- Plan progress
- Plans

## KE Tools - Preparation:

- NDDL & ANML model support
- Client application integration assistant



# Design Process Track



1. ~~Elicitation~~ & Modeling
2. Model Validation & Verification
3. Plan/Schedule Analysis

A little bit



# KE Tools “Platform”

## Eclipse Plugins

- Take advantage of Eclipse bells & whistles  
(we use Eclipse a lot)

## PSUI Package

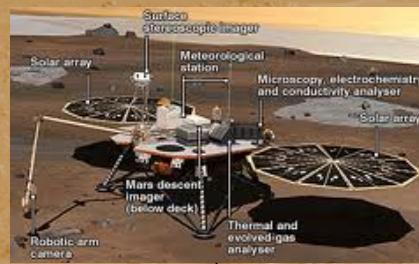
- No Eclipse requirement
- Much more customizable through BSH

# Presentation Outline



- • Example: Mars Rover
- Modeling support:
  - NDDL
  - ANML
- Plan visualization and analysis
- Planning process analysis support
- Client Application Integration Support

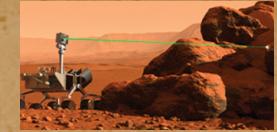
# Mars Rover Example



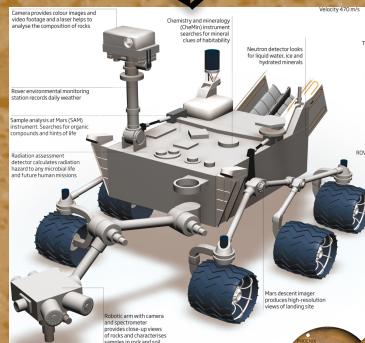
picture



analyze



communicate



sample



# Presentation/Demo Outline



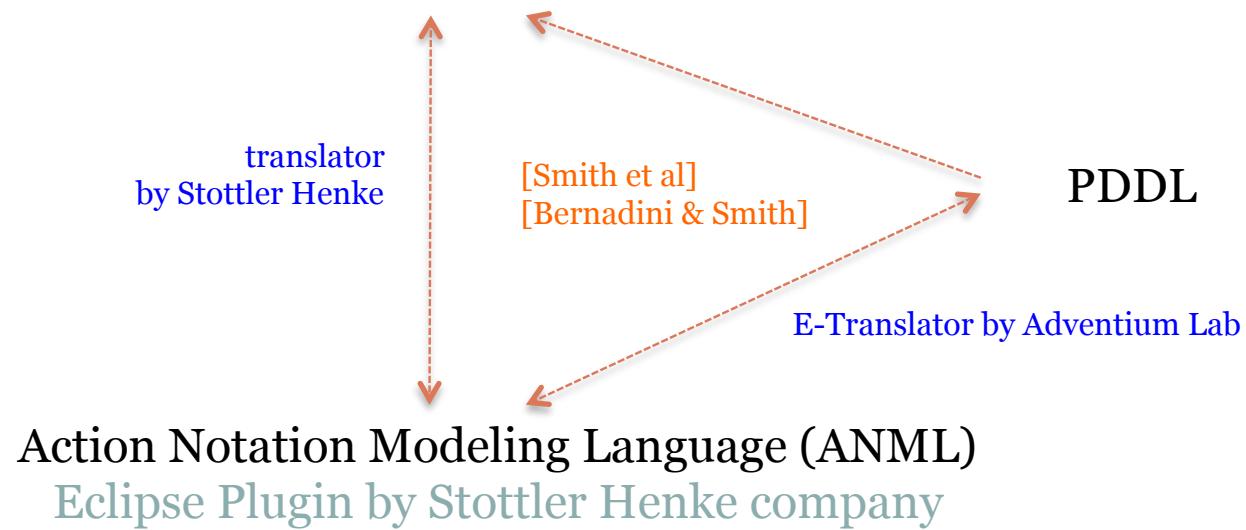
- Example: Mars Rover
- Modeling support:
  - NDDL
  - ANML
- Plan visualization and analysis
- Planning process analysis support
- Client Application Integration Support



# Modeling Support: Eclipse Plugin (no PSUI)



New Domain Description Language (NDDL)  
Eclipse Plugin by Ames PSG



# NDDL Example



## NDDL

```
class Rover {
    Navigator navigator;
    Instrument instrument;
    Battery mainBattery;

    action Go { Location dest; }
    action TakeSample { Location rock; }
    action PhoneHome{}
    action PhoneLander{}

}

class Navigator extends Timeline {
    predicate At { Location location; }
    predicate Going {
        Location from, to;
        neq(from, to);
    }
}

Rover::Go {
    contained_by(condition object.instrument.location.Stowed);
    met_by(condition object.navigator.At _from);
    meets(effect object.navigator.At _to);
    eq(_to.location, dest);

    equals(effect object.navigator.Going going);
    eq(going.to, dest);

    starts(effect object.mainBattery.consume tx);
    eq(tx.quantity, path.cost);
}
```

## PDDL

```
:types
    Navigator Instrument Battery – RoverComponent
    RoverComponent Rover – object

:predicates
    component_of(RoverComponent,Rover)

:constant
    (Going – location)

:functions
    At(Rover) - Location

:durative-action Go
:condition
    (and (over all (stowed(instrument))
        (at start (= At(?Rover) ?_from)))
:effect
    (and (at start (= At(?Rover) Going))
        (at end (= At(Rover) ?_to)))
        (at start (decrease MainBattery(?Rover) tx)))
```

Object-oriented Language

Lisp-like Language

# Modeling Support: NDDL



## Switch to Eclipse Environment

- Syntax highlight
- Object outline
- Error markers

# ANML Example

ANML ← NDDL + PDDL + ADL

PDDL

```
type roverComponent := {Navigation, Instrument, Battery};  
type location < string;
```

```
function bool component_of(roverComponent,Rover);  
function location At(Rover)
```

```
action Go (location from, to)  
{  
    duration := 5  
    [all] { instrument == stowed;  
           location == from :-> to;  
           MainBattery :consumes tx;}  
}
```

## :types

Navigator Instrument Battery – RoverComponent  
RoverComponent Rover – object

## :predicates

component\_of(RoverComponent,Rover)

## :functions

At(Rover) – Location

## :durative-action Go

:duration (= ?duration 5)

## :condition

(and (over all (stowed(instrument))  
 (at start (= At(?Rover) ?\_from)))

## :effect

(and (at start (= At(?Rover) Going))

(at end (= At(Rover) ?\_to)))

(at start (decrease MainBattery(?Rover) tx))

# Modeling Support: ANML



## Switch to Eclipse Environment

analyze  
action-fluent  
relationships

- Text-based ANML editor
  - Highlight, quick-fix, type-hierarchy, auto-completion suggestion
- ANML to PDDL translation
- Action Timeline Summary
- Fluent Action Timeline Summary
- Action Variable Matrix
- Causal chain
- Resource analysis



**Stottler Henke**  
Smarter Software Solutions

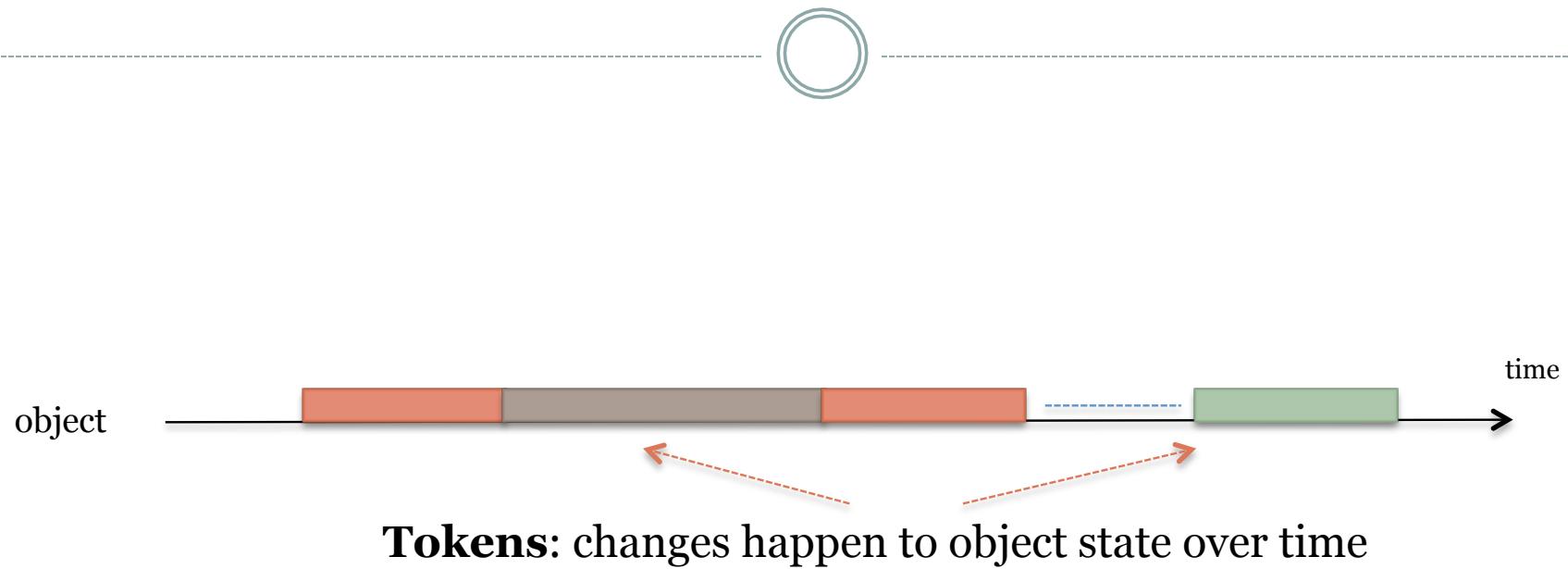


# Presentation Outline



- Example: Mars Rover
- Modeling support:
  - NDDL
  - ANML
- • Plan visualization and analysis
  - Planning process analysis support
  - Client Application Integration Support

# Timeline-based Plan Representation



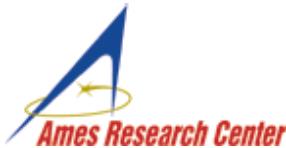
# Plan Visualization & Analysis: EclipseIDE (SWT) & PM/IDE

---



## Switch to Eclipse Environment

- From either NDDL or ANML editor
- Gantt charts on plan timeline
- Plan viewer with pop-up details about plan & token
- Graphs on resource profiles



**Stottler Henke**  
Smarter Software Solutions

# Plan Visualization & Analysis: PSUI



Switch to PSUI Package (no Eclipse) for Rovers Domain

- **PSGantt:** tokens on timeline as a Gantt chart
- **PSChart:** resource profiles as charts
- **Action Details & Action Violation**

# Presentation Outline



- Example: Mars Rover
- Modeling support:
  - NDDL
  - ANML
- Plan visualization and analysis
- • Planning process analysis support
- Client Application Integration Support

# Planning Process Analysis: PSUI

---



Switch to PSUI Package for Visualization Of:

- **Search statistic:** # of steps, time for each step
- **Individual step analysis:** # of flaws, details of each flaw  
possible resolutions, selected resolution

# Planning Process Analysis: Eclipse Plugin

---



Switch to Plugin to run/analyze EUROPA within Eclipse

- **Search statistic:** # of steps, time for each step
- **Individual step analysis:** # of flaws, details of each flaw, possible resolutions, selected resolution

# Presentation Outline



- Example: Mars Rover
- Modeling support:
  - NDDL
  - ANML
- Plan visualization and analysis
- Planning process analysis support
- Client Application Integration Support



# Assist Client Application Integration



- **Makeproject:** quick integration template, template C++ & Java code for utilizing API for various important functions (solving, showing/analyzing flaws, displaying)
- **BSH:** quickly customize the domain analysis & debugging graphical interface

# Summary



- **Modeling support: NDDL & ANML**
  - Error highlight, code auto-completion, PDDL auto-translation (limited), type hierarchy, object outline, references (reader/writer) lookup, filtering, online help
  - Multiple views: action timeline, fluent-action timeline summary, action-variable matrix, action causal relationship analysis, resource profile analysis
- **Plan Analysis:**
  - Plan = tokens on timeline: Gantt charts
  - Resource profiles: graphs/charts
- **Planning process Analysis:**
  - Graphical and text details on: processing time, #flaws, #flaws handled, possible/selected resolutions
- **Client (C++ or Java) application integration:**
  - Makeproject: template building
  - BSH shell: easy shell-script experimenting

# Conclusion



- Pros:
  - Long history of integrating in multiple NASA missions:
    - Well-documented (EUROPA Wiki), well-tested
    - Automated build & unit-test for all popular platforms (Windows, Mac, Linux 32/64 bits)
    - Different ways to test and integrate with native applications
    - **Open-sourced**
- Cons:
  - KE tools spread between several different environments
  - No native support for PDDL (yet)

# Questions?



EUROPA Invited Talk @ PSTL Workshop  
June 26 (Tomorrow) @ 10:00 AM

We will be at the System Demo for further questions!  
Javier Barreiro – EUROPA Lead will be there!