# *EUROPA*: A Platform for Timeline-based AI Planning, Scheduling, Constraint Programming, and Optimization

**Javier Barreiro**[*], **Matthew Boyce**[*], **Jeremy Frank**[†], **Michael Iatauro**[*]
**Paul Morris**[†], **Tristan Smith**[*], **Minh Do**[*]

[*] SGT Inc., NASA Ames Research Center, Mail Stop 269-3, Moffett Field, CA 94035
[†] NASA Ames Research Center, Mail Stop 269-3, Moffett Field, CA 94035

## Abstract

*EUROPA* is a class library and tool set for building and analyzing planners within a Constraint-based Temporal Planning paradigm. This paradigm has been successfully applied in a wide range of practical planning problems and has a legacy of success in NASA applications. *EUROPA* offers capabilities in 3 key areas of problem solving: (1) Representation; (2) Constraint-based Reasoning; and (3) Search. *EUROPA* is a means to integrate advanced planning, scheduling and constraint reasoning into an end-user application and is designed to be open and extendable to accommodate diverse and highly specialized problem solving techniques within a common design framework and around a common technology core. While *EUROPA* is a complete tool set, in this paper, we will mostly concentrate on its timeline-based plan representation and the least-commitment partial-order planning approach operates on top of that representation.

## Introduction

*EUROPA* (Extensible Universal Remote Operations Planning Architecture) is a class library and tool set for building timeline-based planners (and/or schedulers) within a Constraint-based Temporal Planning paradigm (Frank and Jonsson 2003). Constraint-based Temporal Planning is a paradigm of planning based on an explicit notion of time and a deep commitment to a constraint-based formulation of planning problems. This paradigm has been successfully applied in a wide range of practical planning problems and has a legacy of success in NASA applications.

As a complete Planning & Scheduling platform, *EUROPA* offers capabilities in 3 key areas of problem solving:

1. **Representation**: Externally, *EUROPA*'s main input modeling language is the New Domain Definition Language (NDDL) (pronounced 'noodle'). NDDL is a high-level object-oriented modeling language that can describe a number of concepts based on Variables and Constraints. Internally, *EUROPA* allows a rich representation based on *durative tokens* on *timelines* for actions, states, resources and constraints that allows concise declarative descriptions of problem domains and powerful expressions of plan structure.
2. **Constraint-based Reasoning**: *EUROPA*'s main reasoning module contains constraint-processing and inference algorithms to enforce domain rules/constraints and propagate consequences as updates are made to the problem state. Specialized techniques for reasoning about temporal quantities and relations included in *EUROPA* are particularly useful to deal with real-life problem domains.
3. **Search**: by default, *EUROPA* supports chronological-backtracking search with the ability to integrate heuristics into the basic search algorithm and for developing new customized search algorithms.

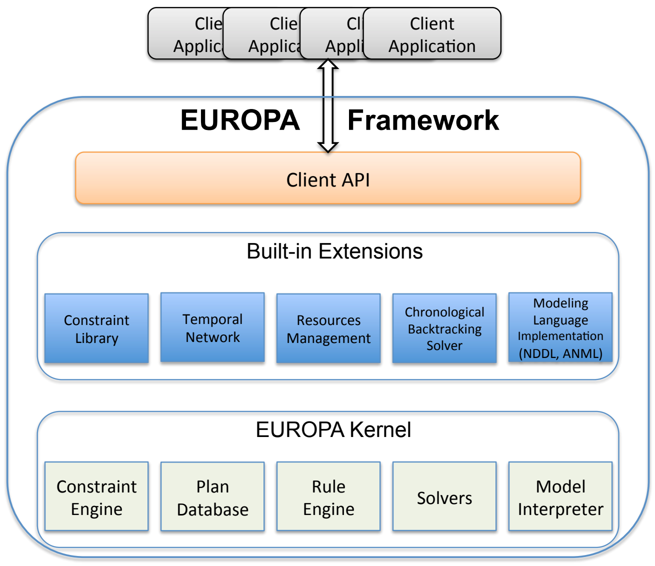

Figure 1: *EUROPA* Architecture

*EUROPA* is designed to be open and extendable to accommodate diverse and highly specialized problem solving techniques within a common design framework and around a common technology core. Figure 1 shows the main components of *EUROPA* and the hierarchical relationships between them: through the *Client API*, applications can utilize or extend either the core components of the *EUROPA kernel* or the built-in commonly used components that are extensions of the kernel components. *EUROPA* is now at version 2.6 which embodies significant evolution from the original *EUROPA*, which in turn was based upon HSTS (Muscettola et al. 1998). It has been made available under an open-source license. The source code and extensive documents on *EUROPA* are available at: *http://code.google.com/p/europapso/*.

## Timeline-based Planning

**Modeling:** The NDDL input modeling language representation includes state and activity descriptions, as is common in planners using traditional modeling languages like the Planning Domain Definition Language (PDDL) that support the variable-value formalism. *EUROPA* thus takes its heritage from planning formalisms like IxTeT and SAS+. The NDDL models are parsed/translated into an internal representation with the following core components:

- **Timeline & Token:** *EUROPA* state variables are represented using *timelines*, and the changing values on timelines represent sequences of states. Specifically, states are represented as a set of temporally extended predicates called *tokens* and the temporal and logical constraints between them. Each token consists of a proposition and a list of parameters, which by default includes the *start*, *end* and *duration* temporal variables. Timelines consist of totally ordered sequences of connected tokens; hence, a timeline can be in only one state at any instant. Note that while tokens mostly represent temporally-extended predicates, which resemble world-state, tokens also represent actions in *EUROPA* and thus may not be on some particular timelines.
- **Compatibility:** The final component of NDDL model is a set of compatibilities/rules that govern the legal arrangements of states on, and across, timelines. These compatibilities are logical implications asserting that if a timeline is in a state (represented by a given token), then other timelines must be in one of a set of compatible states. Compatibilities can incorporate (1) explicit logical and arithmetic constraints on the parameters of the states/tokens or (2) temporal constraints between tokens. *EUROPA* provides a library of such constraints (e.g., all of the Allen's Temporal Relations), and this library can be extended if new constraints are needed.

**Least-Commitment Partial Planning Refinement Search:** EUROPA follows the *lifted plan-space refinement* planning approach. It uses a series of refinements to convert an initial partial plan into a final plan that is complete with respect to the requirements of the planner. A partial plan consists of a set of timelines containing a partially instantiated and ordered set of tokens with the possible flaws of: (1) unbound variables; (2) open condition; and (3) temporal threats. Flaws are resolved one at a time following user-defined flaw filter and flaw selection strategy and a complete plan is returned when there are no flaws. The main flaw types and the resolution strategy for each of them are:

- **Unbound Variable:** a variable in the partial plan whose domain is not a singleton. A unbound variable flaw is resolved by specifying a value from the domain of that variable.
- **Open Condition:** when a new token is added to the partial-plan, its status is *inactive* and it represents an open-condition flaw. This type of flaw can be resolved by: (1) *merge* with an existing active token; (2) *activate* and add to timeline; or (3) *reject* (if this option is allowed for the flawed token). When a token is "activated", it can introduce a new set of (inactive) *slave* tokens. For example, if the activated token represents a durative action, then (slave) tokens representing (pre)conditions and effects are added as new inactive tokens (new open conditions).
- **Threat:** once a token has been placed in the partial plan it may impact other tokens indirectly through possible overlapping requirements on objects, or by creating resource oversubscription. Threats are resolved by imposing ordering constraints among tokens.

There are extendable built-in flaw-filter and flaw-selection strategies to decide which flaw to be handled next. For the built-in chronological backtracking (depth-first) search solver, when a flaw is selected, it represents a branching point in the search graph with the next child search node to be selected/explored basing on the resolution-selection strategy (which *EUROPA* also provides extendable built-in options). The flaw-resolution strategy is backed by strong constraint propagation routines with special emphasis on temporal and resource reasoning.The search process stops when there is no additional flaw to handle. The built-in solver is only one of many search strategies that can be implemented using *EUROPA* all of the solver components are designed to allow the *EUROPA* user to implement other search algorithms and heuristics as required by the problem domain being addressed.

## Current State & Future Work

*EUROPA* and its supporting tools have been going through a long period of development. Besides the core modeling and reasoning capabilities, *EUROPA* also provides a streamlined API to integrate with native client applications, Eclipse's plugins to help build and debug NDDL/ANML models, and visualization capabilities to support plan comprehension and diagnosis. Overall, *EUROPA*'s current main strengths include: (1) proven track record of addressing real life planning and schedulingproblems; (2) expressive modeling capability; (3) flexible framework; (4) strong support for integration with other applications; (5) open-source license. It has been used for a variety of missions, mission-oriented research, and demonstrations, including: DS1: RAX Remote Agent Experiment, Support for operation of the International Space Station's solar arrays, Bedrest study at Johnson Space Center, MER Tactical Activity Planning, Advanced Spaceflight Training Systems Development, MBARI's underwater autonomous vehicle, and Willow Garage's autonomous robot navigation.

Nevertheless, we still have a long list of improvements for *EUROPA*. The most important ones in our opinion are: significant improvements for search (especially heuristic guidance) and inference capabilities, support the ANML and PDDL modeling languages, improve the visualization and debugging tools and allow *EUROPA* extensions to be written in other languages. Given that *EUROPA* is open-source software, we welcome contributions from planning and scheduling researchers and practitioners.

## References

Frank, J., and Jonsson, A. K. 2003. Constraint-based attribute and interval planning. *Journal of Constraints Special Issue on Constraints and Planning* 8(4).

Muscettola, N.; Nayak, P.; Pell, B.; and Williams, B. 1998. Remote agent: To boldly go where no ai system has gone before. *Artificial Intelligence* 103:5–47.