A GOAL-ORIENTED AUTONOMOUS CONTROLLER FOR SPACE EXPLORATION

Presented by Antonio Ceballos Roa, Advanced Space Systems and Technologies, GMV
AUTHORS

- A. Ceballos$^1$
- S. Bensalem$^2$
- A. Cesta$^3$
- L. de Silva$^4$
- S. Fratini$^3$
- F. Ingrand$^4$
- J. Ocón$^1$
- A. Orlandini$^3$
- F. Py$^5$
- K. Rajan$^5$
- R. Rasconi$^3$
- M. van Winnendael$^6$
AUTONOMY IN SPACE SYSTEMS

Autonomous system:
“a system capable of fulfilling its purpose without external intervention”

Motivation for autonomy:
- Challenges
  - Constrained communication links
  - Environment uncertainty
  - Cost
- Objectives
  - Mission performance
  - System survival
GOAC REQUIRED CAPABILITIES

- Deliberative
  - On-board planning and scheduling
- Reactive
  - Quick reaction to dynamic external and internal conditions
- Safe
  - Decision making and execution relying on fresh information
  - Correct behavior enforced at run-time
- Operations
  - Goal-based (E4)
  - Adjustability of ECSS autonomy level (E1 – E4)
- Open architecture
  - flexible, adaptable to mission needs, applicable to a wide range of space systems
T-REX: INTERLEAVING PLANNING AND EXECUTION

- Divide and conquer: partition of the control problem
- Functional scope
- Temporal scope
- Coordinated control loops: reactors
**T-REX: MAIN CONCEPTS**

**Agent**: set of coordinated concurrent reactors.

**Reactor**: control loop, encapsulating details of how to accomplish control objectives.

**Timeline**: a particular evolution over time of a state variable.

**Agent state**: state of all timelines.

**Observation**: current value of a timeline.

**Goal**: desired future value of a timeline.
T-REX: SENSE-PLAN-ACT

**Synchronization**
- Observations produced by reactors to guarantee a consistent and complete view of the agent state.

**Deliberation**
- Deliberative reactors produce goals as a result of solving planning problems.
- Relies on a timeline-based planner.

**Dispatching**
- Goals are dispatched to other reactors for execution.
Domain model (DDL.3)
- Set of state variables and allowed state transitions
- Causal and temporal relationships between state variables

Planning problem
- A future desired state of the system = set of goals

Plan
- Set of state transitions required to achieve the desired state
- Flexible plans
GOAC-APSI PLANNER

Deliberative Reactor

New Goals
Re-planning

Domain.ddl
Init_Conditions.pdl

GOAC-APSI Planner

APSI Framework

Timelines
FUNCTIONAL LAYER

- GenoM: generator of modules
  - A module
    - is responsible for a particular functionality of the robot
    - offers services which implement algorithms
    - exports some data
  - Automatically generated code
    - to cope with low-level software details
    - API library
    - test

- BIP: behaviors, interactions, priorities
  - Composition of heterogeneous components
  - Ensures correctness-by-construction of essential system properties
  - Automated support for component integration
A control architecture as a collection of coordinated control loops, with a recurring sense-plan-act cycle.

High level goals are decomposed into low-level commands.

Divide-and-conquer approach to complexity.

Correct-by-construction functional layer.

Observations flow from the low level to the high level.

Framework for development of modules and reactors.

High level goals are decomposed into low-level commands.

Divide-and-conquer approach to complexity.

Correct-by-construction functional layer.

Observations flow from the low level to the high level.

Framework for development of modules and reactors.
CASE STUDY: DALA SCENARIO

- Navigate safely in an a priori unknown environment.
- Take high resolution pictures of an operator-given list of locations (of scientific interest).
- Communicate with orbiters or a lander during some given visibility windows.
- Continuously monitor the environment for “opportunistic science,” and take appropriate actions when something interesting is detected.
- Monitor and control the proper heating of the platform and the payload.
- Monitor and control the proper power usage and energy consumption.
CASE STUDY: 3DROV SCENARIO

3DROV visualization tool
(Courtesy of TRASYS)

3DROV system architecture
(Courtesy of TRASYS)

Key features:
- Opportunistic science
- Running out of resources
- Dynamic goal injection
- Safety
- Autonomy levels

Navigation cycle:
- Traverse
- Panoramic view
- Zig-zag + pictures

DEM

Picture

DEM (Courtesy of TRASYS)
CONCLUSIONS

- “General purpose” solution: valid both for in-orbit and surface application domains
- Flexible wrt autonomy levels: handling from E1 to E4
- Deliberative and reactive in an efficient way
- Robust execution under uncertainty
- Verification and detection of system properties (e.g. deadlocks)
- Operators can focus on the domain level
- Improved mission performance:
  - Science return
  - Robustness of the robotic system

GOAC responds to the needs of space missions in which higher degrees of on-board autonomy are required
Thank you

Antonio Ceballos Roa
Advanced Space Systems & Technologies
Email: aceballos@gmv.com
www.gmv.com