Automation and autonomy techniques are key elements in realizing the vision for space exploration. Intelligent automation of systems on crewed vehicles is instrumental for decreasing workload, reducing dependence on Earth-based support staff, enhancing response time, and reducing operations cost. Increased system autonomy for unmanned and manned vehicles reduces operations costs, while increasing operations efficiency and spacecraft capability by reducing the time required for humans to staff flight control positions and interact with the vehicles. To enable the application of intelligent automation and autonomy techniques, configuration and validation issues need to be addressed.

Reusable automation software must be adaptable to new applications without undue difficulty, and easily adjusted as the application operations change. The overhead of applying automation techniques to new applications is one of the two key obstacles to acceptance of such techniques in operations. A variation of the same issue is that of adjustment as requirements and application contexts change, which is inevitable in spacecraft operations.

The software and the adaptation to a given application must also be trusted before it can be accepted. Testing and other techniques are keys to establishing such trust and ensuring the correct function of automation systems. However, in both testing and validation, the complexity of intelligent software has proven to be a major obstacle. This has led to trust and correctness issues being another key obstacle to adoption of intelligent automation systems in both unmanned, and most importantly, in crewed vehicles.

Proposals in this area should include autonomy and automation software architectures that facilitate adaptation and ensure correctness. Specifically, proposals in the following technical areas are of high interest:

- Architectures for decision-making and closed-loop control that can be adapted to new applications with minimal reliance on intelligent systems expertise;
- Methodology and techniques for adapting autonomy software to applications, as well as for reconfiguring the software in response to changes;
• Representation and reasoning techniques for specifying properties for application interfaces, operations, flight rules and autonomy software behaviors, and for deriving overall properties for autonomy software applications.