Developing Autonomous AI Planning and Scheduling Technologies for Remote Planetary Exploration

Mark. J. Woods[(1)], Ruth. S. Aylett[(2)], D.P. Long[(3)], Maria Fox[(3)], Roger Ward[(1)]

*[(1)]SciSys Ltd.
Clothier Road
Bristol
BS4 5SS
UK*

*[(2)]Centre for Virtual Environments
Business House
University of Salford
M5 4WT
UK*

*[(3)]Department of Computer Science
University of Durham
Durham
DH1 4LE
UK*

Mark.Woods@scisys.co.uk, R.S.Aylett@salford.ac.uk, Roger.Ward@scisys.co.uk D.P.Long@durham.ac.uk, Maria.Fox@durham.ac.uk

Introduction

Remote planetary exploration has much to gain from the availability of advanced robotic technologies. Missions such as JPL's Pathfinder and the forthcoming Mars Express/Beagle 2 Lander demonstrate how discrete applications of robotic technology can enable comprehensive planetary science activity. The Aurora programme presents the space community with a significant set of technical challenges on a much larger scale. Aurora's objective of a manned mission to Mars in 2030 will require a series of co-ordinated complex, robotic pre-cursor missions. The longevity and magnitude of the programme will impose a need for highly advanced technologies, particularly in the area of autonomy.

AI planning and scheduling is well suited to providing autonomous control for robotic applications. Its use has been widely considered for robotic control in both the space and non-space domains [1][2]. There are a number of on-going programmes such as NASA Ames and JPL's AI planning and scheduling groups which are developing suitable planning and scheduling technologies for space. However, the application of planning and scheduling to this domain is relatively new and significant development will be required in order to meet the challenges of Aurora. This paper outlines work carried out as part of the preparatory phases of Aurora which aims to specify the on-board planning and scheduling requirements for the programme, reviews the state of the art and suggests a way forward for future development.

Planning and Scheduling Technology for On-Board Autonomy

The ability to create and execute detailed plans based on abstract goals has enormous potential benefit for remote planetary exploration missions. Typically such missions are characterised by severely constrained communications links which are limited in terms of frequency, data volumes, transmission rates and 'real-time' access. For example one-way light times to Mars are between approximately 4 and 21 minutes. In the Beagle 2 case communication outages can be in the order of days – due to the requirement of mutual visibility between lander, orbiter and the ground station - with uplink rates in the order of 8kbits/sec and communications session intervals of around 15mins. Such a constrained communications link precludes 'real-time' tele-operation – "Joysticking from Earth is no longer an option" [3].

Recent and current approaches to this problem generally involve communicating sequences of time-tagged, low-level or on-board macro-commands to a spacecraft or probe which execute until successful completion or a problem is encountered. The difficulty with this approach is two-fold. Firstly given the limited nature of the communications uplink, only a small number of commands can be communicated at a time. This is partly alleviated by the use of on-board control procedures as in the Beagle 2 case. However, remote environments are typically uncertain and this is particularly true of in-situ planetary exploration. When sequences fail the Lander or rover is usually put into safe-mode until the problem can be analysed and if possible resolved by ground based mission planners.

This is clearly inefficient as it can take many days before a problem is discovered and many more before it is resolved. Even under nominal conditions the process of planning operations on the ground is manually intensive and time-consuming. Furthermore without access to 'live' data, operations can take place which could endanger the safety of a spacecraft or probe, although safing procedures should prevent this.

Providing remote exploration units with the ability to create their own plans based on up-to-date state information and more importantly enabling them to re-plan in response to dynamic events would greatly improve the efficiency of a mission and potentially improve its safety. Operators can use the restricted communication channel to forward high-level goal requests which the on-board PS system can turn into detailed commands. Execution can be monitored continuously and re-planning invoked when a plan failure occurs. It is difficult to see how missions such as sub-surface Europa exploration could be achieved without this type of technology.

There are many approaches to robotic lander or rover control ranging from deliberative planning to reflexive, behavioural approaches [4][5]. However in applications where autonomy is required there is often a certain level of complexity above which deliberative planning is necessary. It is believed therefore that using planning technology for robotic control is essential in achieving complex goals. It should be noted that its use does not preclude the adoption of other intelligent techniques for lower-level robot control, indeed there are a number of examples of hybrid, deliberative/reflexive architectures which have been developed by the robotic community [6][7].

Aurora Requirements for Planning and Scheduling

Based on [8] it can be seen that Aurora is expected to consist of a number of distinct mission phases such as:

Number	Phase
1.	Surface mapping with optical/RF instruments and maximum extraction of information from orbit
2.	Local environment in-situ characterisation by an array of surface stations (i.e. immobile landers similar to Beagle 2 but more autonomous)
3.	Deployment of landers with mobile robotic support for sample in-situ analysis or return
4.	Installation/building of infrastructure to prepare for the arrival of human crews
5.	Human presence and mobility on planetary surface

Table 1: Aurora Phases

Clearly autonomous planning and scheduling capability could be put to use in many if not all of these phases. However it is difficult to comprehensively specify requirements for all of the individual phases given the technology immaturity and lack of experience in some areas. In order to provide a stable reference point, an autonomous Beagle 2 type mission is considered. An autonomous Beagle 2 mission would form part of phase 2 of Aurora. This class of mission has many of the characteristics of the types of autonomous missions required for the other Aurora phases. This allows specific and generic requirements to be defined and where appropriate matched to the individual phases. In addition, SciSys have specified the requirements and developed the on-board software for the actual Beagle 2 Lander thus allowing actual mission experience to be factored in to the definition process. A complete set of requirements are outlined in the final report of this work. A set of general requirements are presented here as a synopsis:

Num	Requirement
GR1	accept explicit, abstract goals and also higher level metric goals
GR2	specify behaviour or sequences of tasks which can achieve these goals while respecting resource and operational constraints
GR3	manage the execution of this behaviour
GR4	respond dynamically to conflicts which may contradict desired behaviour or violate constraints
GR5	response times (for both low and high-level reactions such as collision avoidance and re-planning) shall be prompt, order of seconds for on-line planning and order of hours for off-line
GR6	PS technology shall facilitate different levels of autonomy including closed-loop goal-based commanding, execution and monitoring of manually created plans and execution and monitoring of traditional command sequences

GR7	PS Technology shall be scalable and suitable for distributed deployment
GR8	PS technology development shall include validation and verification methods which are consistent with on-board deployment

Table 2: General Requirements

Perhaps the most prominent feature of Aurora is its lengthy duration. As the programme matures it is likely that a heterogeneous mix of rovers, landers, base stations and orbiters will be expected to co-operate on achieving a range of tasks and goals. At a lower level, a variety of hardware and software systems, with different levels of ability will be deployed. Algorithms which provide autonomy must therefore take account of this, i.e. should be scalable and able to accommodate distributed architectures. Having defined a set of requirements it is possible to review the state of planning and scheduling technology and assess how it should be developed to meet these complex challenges.

Review of Planning and Scheduling Technologies for Space

AI Planning and Scheduling Technology (PS) goes all the way back to research originally carried out in the 1970s in robot planning [9]. In the recent period, it has started to move out of the research community and into real world applications. This was facilitated in the US by the DARPA Planning Initiative of the 1990s, which focused on military logistics, but which can be seen as a root of the strong groups at JPL and NASA Ames that have taken the technology into NASA programmes.

Initially this work produced ground based applications such as supporting scientists using image processing software [10], or operators of the Deep Space communications network [11]. However PS technology was then flown in the Remote Agent Experiment (RAX) as part of Deep Space 1 [12]. It is also being used for ground based development of planetary rover systems [13]. In the absence of a unifying initiative such as the DARPA programme, ESA has so far lagged behind NASA in taking up this technology, with its use in Optimum-AIV [14] and an Astronaut Training package the only examples to date. However, as argued above, it is hard to see how the necessary level of autonomy and responsiveness needed in the AURORA programme can be achieved without PS technology.

The 1970s work introduced the idea of explicitly representing the actions which could be used to form a plan as *planning operators*, together with logical pre-conditions (what must be true for an action to be validly inserted into a plan) and effects (what new facts will be true and old facts false after such an action is inserted). This basic representational concept has proved very durable, and in slightly elaborated form is still present in the Planning Domain Design Language, PDDL [15]. PDDL is an important development in the AI Planning research community offering a standard way of representing planning problems and making it possible to compare the performance of different planning systems on a standard set of problems in the International Planning Competition [16] which has run biennially since 1998.

The development of PDDL as a community standard also offers the possibility of reuse of planning knowledge between different AURORA applications. It can form a base component in an overall planning toolbox which would allow planning software to be configured for different parts of the programme over its long timeline, rather than constructed anew each time. The importance of this objective is demonstrated by the development generic frameworks for NASA programmes [17, 33] as the US PS experts working on space applications move from one-off experiments to a more general use of PS technology.

Planning operators offer a more robust and application-focused alternative to the weaker expert system representations such as rules sometimes used for ad hoc intelligent planning systems. Apart from supporting modern algorithms that cope much better with the computational demands of automatically generating plans than simple rule chaining is able to do, planning operators also offer more support to humans in the planning process by making the cause and effect links between actions in a plan explicit. Thus they can be used to support an intelligent interface in which the human user is able to manipulate plans without violating any of the causal constraints. Planning operators can either be thought of as representing immediately executable actions, a style called flat planning, or as specified across an abstraction hierarchy, where more abstract operators can be expanded into less abstract ones until a primitive executable level is reached. This style of planning is known as Hierarchical Task Network (HTN) planning [18].

The use of planning operators means that GR1 - *accept explicit, abstract goals and also higher level metric goals* – is met by all PS technology as long as goals for a particular task are presented one set at a time to the planning process. If many tasks must be planned for in overlapping time periods, so that it is necessary to make a choice between working on current goals or giving new ones more priority, then the technology known as *continuous planning* [19] is required, an area researched in the recent period only.

In the classic formulation, planning was seen as the problem of selecting the right actions for a plan, while scheduling was seen as the problem of allocating resources – including time – to a known set of actions. Thus the two fields have until recently been researched separately, within a framework assuming that planning would first select some actions, scheduling could be carried out on the backend and execution would take place after that. However, it is clear that in most real world applications, the choice of actions may depend on the availability of resources, while the allocation of resources is determined by which actions are chosen, so that planning and scheduling activities are interleaved. This is clearly true for AURORA onboard applications where resource constraints – time, power, computational resource, communications bandwidth - are of prime importance. Some development of PS technology is therefore required to meet GR2 - *specify behaviour or sequences of tasks which can achieve these goals while respecting resource and operational constraints*.

The research is however clearly moving in this direction, underlined by the recent introduction of PDDL2.1. This version of PDDL, used in the 2002 International Planning Competition, incorporates explicit representation of time and duration and improved ability to handle numeric quantities, and alongside it was a new set of standard problem domains including a satellite domain [32] in which planning and scheduling were combined. Further work remains, but PS technology is likely to meet this requirement in the relatively short term.

GR3 - *manage the execution of this behaviour* – and GR4 - *respond dynamically to conflicts which may contradict desired behaviour or violate constraints* – involve a different type of interleaving, between planning and execution. Managing execution is partly a low-level scheduling task and partly a monitoring task of making sure that the effects required by the plan actually occur. The monitoring system is generally required to interface with a diagnosis system, since diagnosing an error is usually a requirement for a successful plan that deals with it. Responding dynamically to conflicts refers to the rescheduling or replanning activities required where execution monitoring detects a failure.

After an early and influential paper on interleaving planning and execution [20] only a limited amount of work has been carried out in the generic area, though its importance to AURORA applications is underlined by the integrated planning. execution monitoring and diagnosis system produced for the NASA RAX experiment [21]. Since replanning can be thought of as adding new goals to a planning system, research in continuous planning, cited above, subsumes this area. A less ambitious approach tried in robot planning [2] tries to limit errors to an intelligent execution system running at a lower level of abstraction, but there must be some concerns about how far this will scale up.

GR5, covering acceptable response times, is a largely un-researched area in PS technology apart from a little work on anytime planning. The issue here is that the very large search spaces handled by planning algorithms normally involve heuristic search methods for which the solution time is unpredictable but potentially very long in the worst case. One common way to achieve anytime behaviour is to use a local search strategy, where a seed plan is identified and then iteratively improved by considering local perturbations of the current candidate plan. as in Casper [22]. Decision theoretic approaches to handling contingencies, constructing *policies* rather than plans, have also been explored [23], but are not currently feasible for real world problems due to the computational requirements and representational problems.

GR6 - *PS technology shall facilitate different levels of autonomy* – is another area in which little research has taken place. The potential for different levels of interaction, from strategic control via goal setting, right down to operating joints and motors, is clearly there in multi-level robot architectures which separate out executive, tactical and strategic control. The NIST Reference Architecture of the early 1990s [24] and the ESTEC study [25] of the same period applied this idea to Space Robotics, but were never applied to actual systems. Meanwhile PS researchers have tended to concentrate on more efficient algorithms rather than mixed-initiative interaction between PS systems and human operators. Much remains to be done to turn the basic architectural frameworks into working systems.

GR7 - *PS Technology shall be scalable and suitable for distributed deployment* – requires the integration of PS technology with multi-agent (MA) systems, a related and very active area of research. Work has been carried out into multi-robot cooperation [26] and into more generic distributed PS technology [27], but the trade-off between distribution of decision-making and communication overhead has yet to be satisfactorily settled. In addition, all of the issues raised for single-agent PS need to be re-examined for MA PS, with replanning for example becoming yet more complex. This is an area in which long-term research still needs to be carried out.

Finally GR8 - *PS technology development shall include validation and verification methods* – raises a set of issues generic to all applications of AI technology in a domain such as Space, where the expensive consequences of software errors make risk avoidance essential. Work in viewing planning as a theorem-proving activity combines planning with logical correctness, but is computationally expensive and has limited expressiveness, a problem in common with formal methods in general. Some work has been carried out in developing static and dynamic checking tools for planning systems [28, 29] since the planning operator representation is in some respects easier to check than a rule-based system. Giving users confidence in the correctness of plans without being able to check every single possible plan requires much more thought and there is little work in this area. Indeed, compared to expert system technology, there is in general little life-cycle support for PS [however see 30,31], reflecting its longer gestation in application to real world problems.

Future Development – Technology Roadmap

The table below summarises the discussion of the previous section in relation to the requirements of section 1 by matching specific PS technology developments to each requirement and indicating the time scale on which each is likely to be achieved. The time scale is classified as achieved (A), meaning available now, short (S), meaning within the next five years, medium (M), within the next ten years, and long (L), within the thirty years projected for the AURORA programme. Necessarily the longer time scales are somewhat less definite than the shorter ones. In addition, the current state of the technology is categorised.

Req No.	**PS Technologies for this requirement**	**Time scale**	**Current strength of technology**
GR1	Goals for a single task Goals for many tasks concurrently (continuous planning)	A M/L	Strong Weak
GR2	Explicit representation of time and numerical values Interleaving planning and scheduling	S M	Becoming much stronger Still weak but improving
GR3	Execution monitoring Interfacing with diagnosis system	S M	Weak but improving Weak
GR4	Interleaving planning and execution	M	Weak – limited amount of work so far
GR5	Anytime planning	M	Weak apart from local search
GR6	Architectures for adjustable autonomy Mixed-initiative systems	S M	Existing work in robotics Weak, little work
GR7	Multi-agent distributed planning systems	L	Very weak
GR8	Verification and validation systems	M	Weak – little work

The range of time scales indicates that though much work remains to be carried out in producing PS technology that will meet the AURORA requirements, some components are either ready now or could be developed over the next five years. This could then create both momentum for the researchers and a degree of acceptance by the end-users which would lay the basis for the medium and long term developments required.

Conclusion

PS technology is an indispensable requirement for the ambitious objectives of the AURORA programme, especially in supporting the levels of on board autonomy required for spacecraft and planetary vehicles operating at great distances from the earth in environments which require timely responses to ensure mission success. Work in the US has already demonstrated that given a focus, PS technology can be developed in the directions required, and the evaluation summarised here shows that shorter-term objectives can be used as a jumping off point for meeting the more challenging medium-term and long-term requirements. For example, the NASA 2003 Mars Lander mission is currently actively considering the off-board use of PS technology for its Mars Rovers, incorporating human operator interaction with an automated system as a way, at this stage, of constructing low-level sequences more quickly. This type of confidence-building exercise is one way of building the acceptability of PS technology in operational environments, and might equally be considered by ESA, at least in experimental form, perhaps using the Beagle II command sequences as an example.

While some way from the complexity of the AURORA programme, PS technology could also usefully be applied in the short term satellite domain, where it is clear there are immediate gains to be made in operational flexibility and reduction of ground costs. Again, NASA are taking this path with the application of the CASPER [21] framework to the Three Corner Sat (3CS) [34] and EO-1 missions.

The role of the DARPA Planning Initiative in the US during the 1990s in building up cohesive teams of researchers and focusing them on real world problems should not be underestimated, and suggests that an integrated programme in Europe could also be very productive. Europe has many strengths in PS technology – witness the growth for example of the EU Network of Excellence PLANET – but has tended to concentrate more on theory and algorithm development. The lack of a unified applications focus may well be a factor here. The inclusion of Space in the EU Framework 6 programme and new organisational methods such as the Integrated Projects may provide an opportunity for ESA to contribute to a more generic development of PS technology which would be of benefit not only to AURORA but also to areas outside Space.

References

[1] R. Sherwood, A. Mishkin, T. Estlin, S. Chien, P. Backes, J. Norris, B. Cooper, S. Maxwell, G. Rabideau, "Autonomously Generating Operations Sequences For a Mars Rover using AI-based Planning," *Proceedings of the 2001 Intelligent Robots and Systems Conference,* Maui, HI, November 2001

[2] Firby R. J., “Adaptive Execution in Complex Dynamic Domains”, Ph.D. Thesis, Yale University Technical Report YALEU/CSD/RR #672, Jan 1989

[3] Doyle R., Editor’s note in – *NASA’s New Era in Space Exploration*, IEEE Intelligent Systems March/April Edition 2002

[4] Howard and Serajii H., “A Real-Time Autonomous Rover Navigation System”, World Automation Conference, 2009

[5] Nehmzow U., “Vision Processing for Robotic Learning”, Industrial Robot Vol. 26, No. 2, pp 121-130, 1999

[6] Shenker P. et al, “Robot Work Crews for Planetary Outposts: close cooperation and coordination of multiple robots”, SPIE Symposium on Sensor Fusion and Decentralized Control in Robotic Systems III, Vol. 4196, Boston, Mass, USA pp. 210-220, 2000

[7] Aylett, R., Barnes D., “A Multi-Robot Architecture for Planetary Rovers”, 5th ESA Workshop on Advanced Space Technologies for Robotics and Automation, ESTEC, Noordwijk, The Netherlands, 1998

[8] ESA, “Technologies for Exploration – Aurora Programme Proposal Annex D”, ISBN 92-9092-616-3, 2001

[9] Fikes, R.E; Hart, P.E. & Nilsson, N. (1972) "Learning and Executing Generalised Robot Plans'. Artificial Intelligence, 3:251-288, 1972

[10] Chien, S. (1994) "Using AI Planning Techniques to Automatically Generate Image Processing Procedures: A Preliminary Report" Proc. AIPS94, Chicago, IL, June 1994, pp219-224

[11] S. Chien, A. Govindjee, T. Estlin, X. Wang, R. Hill Jr., "Using Artificial Intelligence Planning Techniques to Automate Generation of Tracking Plans for a Network of Communications Antennas," Proceedings of the 1997 Conference on Innovative Application of Artificial Intelligence, Providence, RI, July 1997,

[12] Bernard, Dorais, Gamble, Kanefsky, Kurien, Man, Millar, Muscettola, Nayak, Rajan, Rouquette, Smith, Taylor, Tung. "Spacecraft Autonomy Flight Experience: The DS1 Remote Agent Experiment," Proceedings of the Space Technology Conference and Exposition (AIAA-99), Albuquerque, NM. September, 1999.

[13] P. Backes, G. Rabideau, K. Tso, S. Chien, "Automated Planning and Scheduling for Planetary Rover Distributed Operations," Proceedings of the IEEE Conference on Robotics and Automation (ICRA), Detroit, Michigan, May 1999

[14] Aarup, M; Arentoft,M.M; Parrod, Y; Stokes, I; Vadon, H. & Stager, J. (1994) "Optimum-AIV: A Knowledge-Based Planning and Scheduling System for Spacecraft AIV". In Intelligent Scheduling, M.Zweben & M.S.Fox (eds) Morgan Kaufmann 1994, pp451-469

[15] M. Ghallab, Adele Howe, Craig Knoblock, Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. PDDL---The Planning Domain Definition Language. Yale University, New Haven, CT, March 1998. Draft 1.0.

I[16] nternational Planning Competition: http://www.dur.ac.uk/d.p.long/competition.html

[17] S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B.Smith, F. Fisher, T. Barrett, G. Stebbins, D. Tran , "ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling," SpaceOps 2000, Toulouse, France, June 2000

[18] K. Erol, D. Nau, J. Hendler. "HTN Planning: Complexity and Expressivity." In AAAI-94, Seattle, July, 1994.

[19] M. Pollack, J.F.Horty, There's more to life than making plans: Plan management in Dynamic, Multi-agent Environments, AAAI 98 Workshop on Distributed Continual Planning

[20] Ambros-Ingerson, J. & Steel, S (1988) Integrating planning execution and monitoring. In Proceedings of the seventh national conference on artificial intelligence (AAAI 88), Saint Paul, MN, (pp. 83-88).

[21] S. Chien, N. Muscettola, K. Rajan, B. Smith, G. Rabideau 1998 " Automated Planning and Scheduling for Goal-Based Autonomous Spacecraft" IEEE Intelligent Systems, pages 50-55, September/October 1998.

[22] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, "Using Iterative Repair to Increase the Responsiveness of Planning and Scheduling for Autonomous Spacecraft," IJCAI99 Workshop on Scheduling and Planning meet Real-time Monitoring in a Dynamic and Uncertain World, Stockholm, Sweden, August 1999

[23] Boutilier, C., Dean, T. and Hanks, S. (1999) "Decision-Theoretic Planning: Structural Assumptions and Computational Leverage", JAIR, Volume 11, pages 1-94.

[24] J. S. Albus. A reference model architecture for intelligent systems design. In An introduction to intelligent and autonomous control, pages 27-56. Kluwer Academic Publishers, Boston, MA, 1993.

[25] Dornier GmbH; Baseline A&R Control Development Methodology Definition Report; study managed by P. Putz; European Space Agency Contract Report; October 1991

[26] Rachid Alami, Frédéric Robert, Félix Ingrand, Sho'ji Suzuki 1995. Multi-robot Cooperation through Incremental Plan-Merging . IEEE ICRA 95.

[27] . M. desJardins, E.Durfee, C.L.Ortiz Jr, M. Wolverton, *A survey of research in Distributed, Continual Planning,* AAAI Fall Symposium on Distributed Continual Planning, 1998

[28] Simpson, R.M., McCluskey, T. L., Zhao, W., Aylett, R. S. & Doniat, C. An Integrated Graphical Tool to support knowledge Engineering in AI Planning. *Proceedings of European Conference on Planning (ECP'01)*, Toledo, Spain. 2001

[29] Chien, S.A; Hill, R.W; Wang, X; Estlin, T; Fayyad, K.V. & Mortenson, H.B.(1996) Why Real-world Planning is Difficult: a Tale of Two Applications. In: New Directions in AI Planning, M.Ghallab & A.Milani, eds, IOS Press, Washington DC 1996 pp 287-98

[30] R.M. Simpson and T.L. McCluskey. 2002. A Tool Supported Structured Method for Planning Domain Acquisition. Springer Verlag LNAI 2366, p. 544 ff.

[31] R Aylett, C Doniat 2002 KA Tool and Domain Construction for AI Planning Applications, ES2002

[32] D. Long and M. Fox, 2002. Domain Modelling and the Satellite Observation Domain, 3rd International NASA Workshop on Planning and Scheduling in Space 2002

[33] Nicola Muscettola Gregory A. Dorais Chuck Fry Richard Levinson Christian Plaunt 2002. "IDEA: Planning at the Core of Autonomous Reactive Agents " The Eighteenth National Conference on Artificial Intelligence, AAAI 2002

[34] S. Chien, B. Engelhardt, R. Knight, G. Rabideau, R. Sherwood, E. Hansen, A. Ortiviz, C. Wilklow, S. Wichman , "Onboard Autonomy on the Three Corner Sat Mission," Proceedings of the 2001 International Symposium on Artificial Intelligence, Robotics, and Automation for Space, Montreal, Canada, June 2001.