CASE FOR SUPPORT

A. Research Context and Aims

Background: Simulating *low-level* cognitive behaviour, such as reaction to stimuli, has been a major focus of research and development in the autonomous systems (AS) community for many years. Automated assessment of sensor data, and reactive selection of actions in the form of condition-action pairs, is well developed in robotic and control application areas. In contrast, a characteristic of more intelligent, *high level* cognitive behaviour is to be able to reason with self-knowledge: an AS knows about the actions it can perform, the resources it has, the goals it has to achieve, the current state and environment it finds itself in; and it has the ability to reason with all this knowledge in order to synthesise, and carry out, plans to achieve its desired goals. So for example an unmanned vehicle on the Mars surface might be requested to collect a rock sample at some position, or a spacecraft might be required to take a photograph of some star constellation. These tasks require an AS to generate or be given detailed plans to achieve them. Enabling applications involving AS to have the general ability to synthesise plans in this manner is a great challenge, because of the difficulty of creating plans fast enough in real-time situations, and the problems in representing and keeping up to date the AS's domain knowledge [34].

Control systems in autonomous vehicles, however, such as in exploration robots or space satellites, need to be capable of planning, scheduling and carrying out long term tasks. For over 20 years scientists at NASA have been developing systems that can, for example, plan activities for spacecraft [4], schedule observation movements for the Hubble Telescope [19], and control underwater vehicles [17]. They have been successful in applying research from the area of automated planning and scheduling (here called *APS*), where domain knowledge is represented declaratively and manipulated via symbolic reasoning (Chien et al [27] gives a good overview of the potential benefits for spacecraft technology). The *APS* research community has been successful in overcoming some of the theoretical problems to do with computational complexity of generative planning, and scale-up of proposed solutions, which dogged the community in the last century. This is evidenced by the wide range of planning applications featuring at this year's annual international conference in *APS*, called ICAPS[1] which included fire fighting, satellite control, emergency landing, aircraft repair scheduling, workflow generation, narrative generation, and battery load balancing. ICAPS also hosts regular competitions leading to the development of optimised planning tools which can be embedded in applications software.

Challenges of Fielding *APS* Applications: The basic challenges of utilising symbolic reasoning systems such as deliberative planners within real time AS are well known, and were neatly summarised some time ago by Wooldridge and Jennings [34]:
(a) the transduction problem: that of translating the real world into an accurate, adequate symbolic description, in time for that description to be useful; (b) the representation/reasoning problem: that of how to symbolically represent information about complex real-world entities and processes, and how to get agents to reason with this information in time for the results to be useful.
The reasoning problem alluded to in (b) is what many in the *APS* community are aiming to solve, and a measure of their success is the growing range of applications referred to above. It is expected that this ongoing research will lead to yet more efficient solvers, which can reason with more expressive input knowledge. Just as challenging, and the subject of this proposal, is the transduction and the related representation problems of (a) and (b).

For an AS to produce plans and decisions rationally using symbolic reasoning, it has to have explicit knowledge of its domain's actions, resources, goals, objects, states and environment. A representation of such knowledge is called a *domain model*, and separation of the concerns of creating a domain model, and the creation of a planning algorithm, is the basis of what is termed *domain independent planning.* While the development of automated planning algorithms has been encouraging, a major problem remains in *APS* applications, which limits their adaptability, and makes them difficult to maintain and validate: much of the AS's self-knowledge has to be encoded in a domain model before its operation. Experience has shown that eliciting and validating domain models involves a great deal of expert time and effort. It also means that if the AS's capabilities change, for example if the preconditions or effects of an action change, then new knowledge describing this must be re-entered into the system by human experts. In fact acquiring, validating and maintaining a domain model for the purposes of automated reasoning is a key research challenge, and has long been a limiting factor in the exploitation of domain independent planning. While currently domain models are hand crafted and maintained, in AS they are required to be automatically learned and subject to adaptation over run time. The aim of this project is to work towards overcoming this research challenge, expressed in the research hypothesis:
Automatically learning and adapting an accurate and adequate domain model for the purposes of symbolic reasoning, in particular for the processes of APS, enables effective, sustained goal-directed behaviour for real time dynamic AS.
By the end of this project we aim to have demonstrated to prototype the feasibility of using a self-adapting domain model to support real time deliberative planning in AS, in applications supplied by the collaborating partners in the AIS Programme Call. If this challenge is achieved, then it will open the door to implementing high-level cognitive behaviour in real time dynamic AS.

In the following section we survey the state of the art in representing domain models, focusing on *adequacy*, that is the expressiveness of domain model languages, and *accuracy*, the validation of the model. In the succeeding section we focus on machine learning techniques which can be used to initially learn and then adapt the domain model.

Domain Model Languages: The control mech-

[1] icaps11.icaps-conference.org

anisms of ASs need to be able to represent and reason with rich and detailed knowledge of such phenomena as movement and resource consumption in the context of uncertain and continuously changing environmental conditions [11]. Traditionally, physical systems with discrete and continuously-varying aspects have been represented using the mathematical notion of a *hybrid* dynamical system. This is a system that has a state made up of a set of real and discrete-valued variables that change over time according to some fixed set of constraints. Hybrid systems are used for modelling in applications such as embedded control systems [5].

The research-led standard domain model language in planning is PDDL (planning domain description language), which is based around a world view of parameterised actions and states, where it is assumed that a planner generates a collection of instantiated actions to solve some goal posed as state conditions. It has been extended to cope with real applications such as crisis management [8] and workflow generation [25], and has versions which can represent time and resources. More expressive modelling languages such as PDDL+ have been developed for applications where reasoning about processes and events in a hybrid discrete/continuous world is necessary [9]. PDDL+ was recently used in an application for developing multiple battery usage policies [18]. Although PDDL is designed for logical precondition achievement, specialist forms of planning can be incorporated into the language using *procedural attachment* [7]. Using this kind of mechanism low level planning procedures such as real time geometrical reasoning or path planning, which benefit from a range of specialist techniques [20], can be incorporated within PDDL.

Despite its widespread acceptability, a serious problem with PDDL is that it reflects the concerns of those working on plan generation algorithms, rather than the execution and scheduling orientation of many applications. In contrast, scientists at NASA Ames developed the application-oriented language families HSTS [21] and then NDDL [16] for their applications in the Space arena. NDDL differs from PDDL in that encodings are based around representations of objects and object instances, which persist in predefined timelines of continuous activities. Each activity has a start and end time interval (to represent uncertainty of duration), and the distinction between *action* and *state* is effectively blurred. Plan generation and execution are therefore linked to a much greater degree than with PDDL. NDDL's concept of timelines is related to the idea of crafting abstract plans as in the input languages to HTN systems [15]. The idea of pre-written hierarchical plans to formulate possible behaviours has long been a popular type of formalism in which to encode dynamic knowledge for *APS* applications. A related view of how one could formulate dynamics comes from the area of Cognitive Robotics [24], which also seeks to emphasise the integration of planning and execution. The idea here, though, is to start with an axiomatisation of the application environment using a variant of situation logic, then hand craft generic plans (so-called 'action programming') from which concrete plans can be efficiently derived using deduction. Systems used in Cognitive Robotics such as GOLOG require more engineering for individual applications than in classical planning, but appear more appropriate for the control of robotics devices.

Another strand of research, closely linked to HTN and practical planning, has focussed on *rich plan representations* [22, 29, 30]. These representations are intended for the sharing of plans between agents. The richness of these languages stems from the underlying ontology that contains generic concepts from the planning domain. They have been used in a number of application domains such as emergency response [23] and personnel recovery [33].

The common role of these rich and expressive language families is to enable engineers to formulate an adequate representation of structural, dynamic and heuristic knowledge for applications involving action and change. In real time autonomous systems these languages have been used to represent a high level knowledge layer. *The key limitation here is the hand coded nature of this kind of knowledge, and the difficulty of validating and maintaining the model* - all current applications rely on teams of knowledge engineers to encode and validate the domain model [13]. To meet the challenge of domain modelling in NDDL, recent work by NASA scientists is aimed at developing an interactive domain model editor which uses a simulator to short circuit the loop between the model and validation of the model [3]. This work also points to the use of machine learning techniques (some developed by the authors of this proposal) to assist in engineering the model. Another promising method that can be used to automatically synthesise a planning domain model is to translate from an existing formal model in an *application lanaguage*. The ICKEPS-09 competition was devoted to this area, with applications including e-Learning, web services composition, and business processing [26]. While this line of work is important in the context of embedding planning components in applications such as workflow planning, this is not so suitable for AS where no formal model exists a priori. Also, in AS the domain model is subject to refinement and adaptation over time, in order that a goal directed planning function will remain effective. We propose to adopt machine learning techniques to effect both the initial acquisition of the domain model, and its evolution over its lifespan.

Machine Learning of Domain Models: Machine Learning applied to *APS* has attracted a long history of research, and we point the reader to a recent survey for a full account [12]. There have been many events on the subject in recent years including workshops adjunct to AI international conferences (including ICAPS), and elements of the ICAPS competition series (ICKEPS/IPC). In the context of domain independent planning, as well as research aimed at learning a domain model representing the physics of the world, much of the machine learning work is aimed at learning heuristics to make the use of a planning engine more efficient.

Domain model learning can be separated into three concerns: (i) what language is the learned domain model going to be expressed in? (ii) what inputs (training examples, observations, constraints, partial models etc) are there to the learning process? (iii) what stage is the learning taking place - initial acquisition, or incremental, online adaptation? For much of the work done up to now the answers to (i) are "some variant of PDDL forming a domain model that can be input to planning engines" and to (iii) is initial acquisition. However, adaptation can be viewed as a special case of initial acquisition, where input to the learning process includes the current domain model as well as training examples etc, and output is the updated model.

Regarding (ii), systems that learn very expressive domain models tend to demand most detailed input. Work in learning domain models for robotic agents [1, 2] assumes that a training mechanism exists with rich feedback mechanisms. Typically, much a priori knowledge is assumed, such as predicate descriptions of states, and partial or total state information before and after action execution. With such rich inputs, systems such as Amir's SLAF [1] can learn actions within an expressive action schema language.

Some recent work on learning domain models has concentrated on learning from example plan scripts but with little or no input domain knowledge. The LAMP system [36] can form simple PDDL domain theories from example plan scripts and associated initial and goal states only. It inputs object types, predicate specifications, and action headings, and from plan scripts taken from planning solutions, it learns a domain model. The domain model is synthesised using a constraint solver, inputting two sets of constraints: one set is based on assumed physical, consistency and teleological constraints - for example, every action in the example plan script adds at least one precondition for a future action, actions must have non-empty effects, and so on. The other set of constraints is generated using a type of associative classification algorithm which uses each plan script as an itemset, and extracts frequent itemsets to make up constraints. While LAMP is aimed squarely at helping knowledge engineers create a new domain model, LOCM is an algorithm which learns from plan scripts only [6]. As with ARMS, it outputs a planning domain theory in a PDDL format but it inputs *only* plan scripts - it does not require representations of initial and goal states, or any descriptions of predicates, object classes, states etc. LOCM has been used in a system that learns to play the Freecell game by observation, with no a priori knowledge of the game [6].

There have been several other notable developments in learning in uncertain or partially known domains. *Reinforcement learning*, traditionally used in single goal or policy learning planners, has recently been developed for symbolic or relational learning, though its potential for learning full models of the PDDL variety is not yet proven [12]. A promising approach towards learning incomplete and uncertain domain models is ongoing in the *Model-lite* project [35]. Here the authors use probabilistic logic as the basis for the language of the learned domain model.

B. Summary of Aims and Objectives

To summarise, before AS in real time dynamic applications can attain high-level cognitive skills there are still major challenges to be overcome in the acquisition, validation and adaptation of domain knowledge. To be able to perform deliberate reasoning in new or changing domains, we propose that an AS needs to be able to learn and incrementally adjust its understanding of the world, encapsulated in a domain model. It needs to ensure the accuracy of the evolving domain model with the help of internal verification checks and external validation constraints. The project aims to work towards the solution of these challenges within a programme involving collaborator applications (identified as *CAs* below) put forward by collaborators in the consortium behind the AIS Programme Call. Hence, we set up the following objectives, the achievement of which will be measured using the criteria following each one:

1. *research and develop an expressive domain model language (here called AIS-DDL) for AS*
   Criteria: AIS-DDL will be a generic language, adequate to capture domain models for at least three CAs. It will be capable of capturing knowledge about actions and change at a human-understandable level of abstraction, and allow for efficient reasoning as required for learning, planning and validation.

2. *research and develop methods for automated learning and online adaptation of models in AIS-DDL*
   Criteria: the methods will be generic to the CAs, and will maintain the accuracy and adequacy of the domain model, and develop heuristic knowledge to support planning functions;

3. *determine methods and develop tools for knowledge analysis, verification and validation (V&V)*
   Criteria: the methods will be able to detect inconsistencies in the learned models, derive new knowledge, and inform further knowledge acquisition and learning cycles. Further, V&V criteria will be in terms humans can understand, thus enabling a mixed-initiative approach to knowledge engineering where appropriate.

4. *deliver a prototype demonstrator system*
   Criteria: The system will exhibit deliberative planning within the CAs in a virtual world, and therein demonstrate the efficacy of domain model acquisition and online adaptation.

Relation with AIS Programme Call: This proposal will advance the state of knowledge in four areas of the Call's Research Interests table as detailed in the Pathways to Impact Document attached.

C. Method and Technical Plan

Overview: This research project's method will be based around the following activities:
–the creation/acquisition of a simulation environment tailored to each CA, analogous to that proposed by Scientists from NASA/JPL to explore

mixed-initiative knowledge engineering [3]. This will provide the necessary environment for experimention with the acquisition, verification and validation of domain models, and resulting synthesised plans;
–utilising an hierarchical approach to the simulation of AS architecture, with abstract symbolic knowledge at a high level to enable long term planning, with detailed knowledge at a lower level to enable path planning or manipulator planning;
–the creation of verification axioms and processes based on the *ontological constraints* intrinsic to the design of AIS-DDL (analogous to those developed for PDDL [32]);
–the engineering of a set of immutable validation constraints capturing some of the physics of the CAs;
–drawing on the techniques used in existing domain model learning tools such as SLAF and LAMP referred to above, the proposer's own recent research in learning domain models [6, 14], and early experiences of learning models in reactive AS [2];
- utilising the rich sources of relevant literature, for example the Space workshop series[2].

With these developments in place, it will be feasible to meet the main challenge of automated learning of domain models in AIS-DDL. These models will be translated into the input language of existing planning engines in order to test generated plans in real time using simulation, and the simulator will be used as the basis of the subsequent demonstrator system. The workpackages (WPs) making up the programme are detailed below:

WP1. Analysis of CAs and State of the Art: Determination and analysis of requirements of the set of CAs which cover the high level planning and decision making function of the AS, drawn from members of the AIS consortium; meetings with domain experts, acquisition of documentation and other appropriate resources describing CAs. For each CA: determination of required planning function, collation of sample required plans, state representations and sensor/effector information, and scope of application.
Distill the state of the art in APS from the literature as applicable to the case studies. Acquisition and testing of applicable tools eg specialist and general planners, learning tools, with potential for use in the project.
Construction of project web site and consideration of routes to transfer technology and exploit research outputs. Consideration of potential for integration of project results with other funded research in the AIS programme.
Delivered: Agreements on the detail and scope of the CAs, such as I/O from/to a deliberative planning function, and a set of detailed criteria with which to measure success [D1]; a collection of literature and summary overview of applicable state of the art in planning and learning techniques[D2], a repository of potentially applicable research tools, project website, and initial report on the integration of research results within the AIS programme[D3].
Evaluation: Scope of CAs to be sufficiently testing to measure all the planned features of the domain model language, the learning method, online adaptation, validation etc. The survey will be of publishable standard, and the tools repository will be used to demonstrate to collaborators the potential of current automated planning and learning technology.

WP2. Configuration of Simulation Environment: Using D1, D3 and collaborator resources where applicable, configure or acquire a simulation environment, for example based on a virtual world platform (such as Second Life), to simulate CAs. Identify the abstractions made and the effort required to transfer systems developed in the virtual world to a real scenario.
Delivered: report on abstractions made in the virtual world[D4]; working application simulator, and well defined interfaces [D5],
Evaluation: simulator configured to showcase the chosen CAs, with the ability to embed APS technology such as generative planners, demonstrate the execution of plans based on learned domain models, and handle user interaction during execution. The visualisation should be adequate to satisfy the owners of the CAs.

WP3. Domain Model Representation and Ontology: Utilising D2, gain insights from the major AI approaches to domain model representation (e.g. in classical planning, action programming, constraint-based planning), and formalisms used in hybrid systems design [5], SAT-based mixed discrete/continuous systems [28], classical-based formalisms [9], and situation-calculus-based work [10]. Clarify the relationship between high level notations and low level reactive planning knowledge as used in the CAs, and specify a generic I/O language for the planning component. Combine with insights from D1, and anticipating the need to learn domain models, create the first version of AIS-DDL.
– Define a rich ontology of domain independent planning concepts for representing processes, events, actions, uncertainty, and continuously changing variables that will provide the abstract vocabulary for AIS-DDL;
– Design and implement algorithms that map AIS-DDL to known languages such as variants of PDDL to utilise state of the art planning technology.
Delivered: specification of generic planner I/O [D6], AIS-DDL[D7], specification of domain model language ontology[D8], translators[D9].
Evaluation: D6 and D7 will fit the requirements of the planning function and model represention (respectively) of the CAs (evaluated by hand encodings of collaborator problem domains). D8 will be evaluated by peer reviewed publication and in combination with D9 using dynamic testing (in WP4 and WP5).

WP4. Verification and Validation: This WP will research and develop methods and tools for the verification and validation of AIS-DDL domain models, resulting in more accurate and robust domain models, and a way of validating the doman model learning processes(WP5). The work will draw on D6,D7 and D8 and relevant literature [32, 13, 15], and investigate:

[2]http://www.congrexprojects.com/11c05/

a) automated verification analysis: the creation of verification axioms and verification tools based on the ontological constraints intrinsic to the design of AIS-DDL
b) automated validation checks: the engineering and encoding of a set of immutable validation constraints capturing the underlying physics of each of the CAs, and a set of validation tools
c) a visualisation and mixed-initiative knowledge engineering tool to allow users to validate by inspection, and manipulate, the domain models.
These tools will be used to identify and help remove bugs, and in particular:
– to provide additional input knowledge during the knowledge acquisition process, and to inform each cycle of domain model adaptation;
– to provide information relevant for the efficiency with which planners can solve planning problems, advice on the best planner to use, and help in optimizing the representation to support efficient automated planning.
– to augment learned models with knowledge useful for the human user (to make them more understandable and intelligible), and useful for enabling translation to other formalisms;
Delivered: verification axioms and tools [D10], validation knowledge and tools [D11], knowledge engineering tool [D12], report on specification and computational properties of tools[D13]
Evaluation: D10-D12 will be evaluated taking into account number of errors identified from test scenarios, the quality of the additional knowledge created, and the success in integrating the output with learning functions in WP5, D13 will be submitted for peer review.

WP5. Acquisition and Adaptation of Domain Models Utilise D2 and D3 to further investigate forms of knowledge acquisition and learning, and methods for domain model creation. Assemble a number of sources of input to machine learning, for each of the CAs: (i) sets of sample information available to the CAs, from simulated sensor data, (ii) engineered and derived knowledge from D10,D11 in WP4, such as domain invariants used for validation checking. Utilise knowledge engineering tools as appropriate to create sample domain model encodings for the CAs, to be used to evaluate the acquisition process. Utilising D7 (the planning ontology), and insights from the literature e.g. [36, 6], and dependent on the kind of learning data available:
a) create an initial domain model acquisition tool
b) create an adaptation tool for evolving the domain model through its online use.
The acquisition tool is likely to be based on a training approach, where sample plans are supplied to it (or observed by it) and in the context of its domain invariants, it induces action structures. The adaptation tool is likely to utilise theory revision or incremental learning techniques, where feedback from the failure of a plan helps to identify and remove bugs in the domain model.
Delivered: Hand engineered domain models[D14], learning[D15] and adaptation[D16] tools, specification and computational properties of tools[D17]
Evaluation: Learned domain models will be compared to D14; the process of adaptation of domain models will be evaluated operationally within the demonstrator(WP6), D17 will be sent for peer review publication.

WP6. Demonstrator Systems, Project Evaluation and Exploitation: Development of the simulation environment to incorporate autonomous behaviour in order to demonstrate system learning and adaptation capabilities: this will fully simulate CAs, using for example a hybrid architecture [31] to integrate learning and planning components with virtual sensor and effector capabilies, leading to extensive testing using CAs scenarios (testing will run in parallel with development of learning techniques in WP5). There will follow an overall evaluation of the project; future development exploration, including integration with other results in the AIS progamme; identification using D3 of effort need to transfer results from the virtual world to the real, and determination of exploitation routes of developed technologies.
Delivered: final versions of simulation environments and demonstrator scenarios[D18]; pathway to research exploitation document[D19]; final project report[D20]
Evaluation: evaluate D18, D19 against success measures identified in D1 and take up of research results by commercial partners; peer reviewed journal publications derived from D20.

Project Management: The project's Workplan (see attached) illustrates deliverables, milestones, WP duration, and approximate WP resource at each of the Universities. Huddersfield will lead WP3, WP5, WP6; Edinburgh will lead WP2, WP4; WP1 will be jointly led. For WP6, researchers at Edinburgh in the third year will work on creating the interfaces necessary to make the Simulation Environment suitable as a demonstrator, and this work will be carried on in the fourth year at Huddersfield. Milestone meetings will take place at each 6 monthly milestone M0 - M8 with all University project staff attending, and other stakeholders and collaborators invited as required. The meetings will be used to review deliverables produced at each milestone, and to keep under review a detailed workplan for the remaining part of the project. Prof McCluskey will provide overall leadership of the project, and convene/chair milestone meetings.

Project Risks: We identify major risk areas in the project as (a) feasibility of creating simulations of CAs (b) poor degree of fit between planning technology and the application requirements (c) difficulty in obtaining and eliciting underlying knowledge. The range of potential CAs (as demonstrated in the Programme Call) and the similarity of them to existing planning developments (eg Mars Rover) mitigate against (a). The wide experience of the Proposers in applying APS, and in the knowledge engineering aspects in general, will help resolve problems arising in (b) and (c) by judging what is feasible in terms of the scope and range of the CAs given the timescale.

References

[1] E. Amir. Learning partially observable deterministic action models. In *Proc. IJCAI 05*, pages 1433–1439, 2005.

[2] S. Benson. *Learning Action Models for Reactive Autonomous Agents*. PhD thesis, Stanford University, 1996.

[3] Bradley J. Clement, Jeremy D. Frank, John M. Chachere, Tristan B. Smith, Keith Swanson. The challenge of grounding planning in simulation in an interactive model development environment. In *Proc. KEPS Workshop, ICAPS*, pages 23–30, 2011.

[4] J. L. Bresina, A. K. Jónsson, P. H. Morris, and K. Rajan. Activity planning for the Mars Exploration Rovers. In *Proc. ICAPS*, pages 40 – 49, Monterey, California, USA, 2005.

[5] L.P. Carloni, R. Passerore, A. Pinto, and A. Sangiovanni-Vincentelli. *Languages and tools for hybrid systems design*. 2006.

[6] S.N. Cresswell, T.L. McCluskey, and Margaret M. West. Acquiring planning domain models using LOCM. *Knowledge Engineering Review (To Appear)*, 2011.

[7] P. Eyerich, T. Keller, and B. Nebel. Combining Action and Motion Planning via Semantic Attachments. In *Proc. ICAPS*, 2010.

[8] J. Fdez-Olivares, L. Castillo, O. Garcia-Perez, and F. P. Reins. Bringing users and planning technology together: experiences in SIADEX. In *Proc. ICAPS*, pages 11 – 20, Cumbria, UK, 2006.

[9] M. Fox and D. Long. Modelling mixed discrete-continuous domains for planning. *Journal of Artificial Intelligence Research*, 27:235 – 297, 2006.

[10] H. Grosskreutz and G. Lakemeyer. On-line execution of cc-golog plans. In *Proc. IJCAI*, pages 12 – 18, 2001.

[11] J. Bresina, R. Dearden, N. Meuleau, S. Ramakrishnan, D. Smith and R. Washington. Planning under Continuous Time and Resource Uncertainty: A Challenge for AI. In *Proc. Conference on Uncertainty in AI*, 2002.

[12] S Jiménez, T. De la Rosa, S. Fernández, F. Fernández, and D. Borrajo. A review of machine learning for automated planning. *The Knowledge Engineering Review (in press)*.

[13] D. Long, M. Fox, and R. Howey. Planning domains and plans: Validation and analysis. In *Proc. Verification and Validation in Planning workshop, ICAPS*, 2009.

[14] T. L. McCluskey, S. N. Cresswell, N. E. Richardson, and M. M. West. Action Knowledge Acquisition with Opmaker2. In *Agents and Artificial Intelligence*, volume 67 of *Communications in Computer and Information Science*, pages 137–150. Springer Berlin Heidelberg, 2010.

[15] T. L. McCluskey, D. Liu, and R. M. Simpson. GIPO II: HTN Planning in a Tool-supported Knowledge Engineering Environment. In *Proc. ICAPS*, 2003.

[16] C. McGann. How to solve it: Problem solving in Europa 2.0. Technical report, NASA Ames, 2006.

[17] C. McGann, F. Py, K. Rajan, J. Ryan, and R. Henthorn. Adaptive control for autonomous underwater vehicles. In *Proc. AAAI*, pages 1319–1324. AAAI Press, 2008.

[18] M.Fox, D.Long, and D.Magazzeni. Automatic Construction of Efficient Multiple Battery Usage Policies. In *Proc. ICAPS, Frieburg, Germany*, 2011.

[19] G. E. Miller. Planning and scheduling the hubble space telescope: Practical application of advanced techniques. In *Artificial Intelligence, Robotics, and Automation for Space Symposium*, pages 339 – 343, 1994.

[20] M.Naveed, A.Crampton, D.Kitchin, and T.L.McCluskey. Real-Time Path Planning using a Simulation-based Markovian Decision Process. In *31st SGAI International Conference on AI (to appear)*, 2011.

[21] N. Muscettola. HSTS: Integrating planning and scheduling. In *Intelligent Scheduling*, pages 169–212. Morgan Kaufmann, 1994.

[22] A. Pease and T. Carrico. Object model working group core plan representation. Technical Report AL/HR-TP-1996-0031, United States Air Force Armstrong Laboratory, Wright-Patterson AFB, OH, 1996.

[23] S. Potter, A. Tate, and G. Wickler. Using I-X process panels as intelligent to-do lists for agent coordination in emergency response. In *Proc. 3rd Information Systems for Crisis Response and Management (ISCRAM)*, 2006.

[24] R. Reiter. *Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.

[25] A. Riabov and Z. Liu. Scalable planning for distributed stream processing systems. In *Proc. ICAPS*, Cumbria, UK, 2006.

[26] Roman Bartak, Simone Fratini, and Lee McCluskey. The third competition on knowledge engineering for planning and scheduling. AI Magazine, Spring 2010, 2010.

[27] S.Chien, G. Rabideau B. Smith, Nicola N. Muscettola, and K. Rajan. Automated planning and scheduling for goal-based autonomous spacecraft. *IEEE Intelligent Systems*, 13:50–55, September 1998.

[28] J.A. Shin and E. Davis. Processes and continuous change in a sat-based planner. *Artificial Intelligence*, 166, 2005.

[29] A. Tate. Roots of SPAR—shared planning and activity representation. *The Knowledge Engineering Review*, 13, 1998.

[30] A. Tate. <I-N-C-A>: A shared model for mixed-initiative synthesis tasks. In *Proc. IJCAI Workshop on Mixed-Initiative Intelligent Systems*, 2003.

[31] A. Walczak, L. Braubach, A. Pokahr, and W. Lamersdorf. Augmenting BDI Agents with Deliberative Planning Techniques. In *in The Fifth International Workshop on Programming Multiagent Systems (PROMAS*, 2006.

[32] G. Wickler. Using planning domain features to facilitate knowledge engineering. In *Proc. KEPS Workshop, ICAPS*, 2011.

[33] G. Wickler, A. Tate, and J. Hansberger. Supporting collaborative operations within a coalition personnel recovery center. In *Proc. 4th Knowledge Systems for Coalition Operations (KSCO)*, pages 14–19, 2007.

[34] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[35] S. Yoon and S.Kambhampati. Towards model-lite planning: A proposal for learning & planning with incomplete domain models. In *Proc. Workshop on AI Planning and Learning, ICAPS*, 2007.

[36] H.H. Zhuo, Q. Yang, D. H. Hu, and L. Li. Learning complex action models with quantifiers and logical implications. *Artificial Intelligence*, 174(18):1540 – 1569, 2010.

Track Record of Proposers

The University of Huddersfield

T.L.McCluskey (PI): Prof Lee McCluskey obtained a PhD in computer science while working as a lecturer at the City University in 1989. He holds a BSc and MSc in Mathematics from the Universities of Newcastle and Warwick respectively. During the last 25 years he has been research active in areas such as *APS*, Machine Learning, Knowledge Engineering, Formal Methods and Domain Modelling. Currently he is Professor of Software Technology at the University of Huddersfield, where he is Research Director in the School of Computing and Engineering at Huddersfield. The School holds over 100 PhD students, has just secured a c.£8 million research grant for an EPSRC Centre in Innovative Manufacturing in Avanced Metrology, and will play a major role in the University's new c.£12 million Enterprise and Innovation Centre to be opened in May 2012. His relevant expertise for the project falls into several categories:

Knowledge Engineering, Domain Modelling and Verification and Validation: In the early 1990's McCluskey led a series of projects funded initially by the CAA, then by the National Air Traffic Services, under the heading of *FAROAS* (*formalisation and animation of requirements for oceanic aircraft separation*), successfully delivering a domain model called the *CPS*. Written in many-sorted first-order logic, the *CPS* embodied separation criteria for aircraft management in the North Atlantic, and was secured in a tools environment which enabled its verification, validation, animation and maintenance.

Knowledge Engineering for Planning (KEPS): McCluskey's basic research [4] resulted in the five partner EPSRC responsive mode project (GR/M67421) *Planform: An Open Environment for Building Planners* for which he was overall leader. This led to the development and integration of planning technology, notably the development at Huddersfield of KEPS software called GIPO [6]. In 2005 at Monterey, USA, GIPO won the *best tools award* at the First International Competition on Knowledge Engineering for AI Planning and Scheduling (ICKEPS), and has had a leading role in shaping subsequent research in the area. Currently McCluskey works closely with Roman Bartak of Charles University, Prague, in organising annual events in KEPS at the annual ICAPS conferences.

Machine Learning: Prof McCluskey's PhD was in the area of machine learning of heuristics for domain independent planners. Combining his machine learning work with engineering of domain models led to the EPSRC responsive mode project (GR/K73152) *IMPRESS: Improving the Quality of Formal Requirements Specifications Using Machine Learning Techniques (1996-98)* which developed meta-tool technology to investigate the application of machine learning to the validation of domain models. The project successfully developed tools based on *theory revision* which automatically revised the *CPS* model to fit with training data [5]. Recently he has returned to work on applying machine learning to *APS*, concentrating on using training examples to learn planning domain models. This research has been embodied two learning tools: *Opmaker* for learning from one plan script example and *LOCM* [2] for learning from many example plans. This, and his earlier work on theory revision, will directly support research in WP5 of this proposal.

Autonomic Systems in Transport: Currently Prof McCluskey is leading COST Action 1103 called *Towards Autonomic Road Traffic Support Systems (2011-15)* with 30 partners throughout Europe. This aims to integrate disparate research in intelligent traffic management. Its research focus of self-managing, self-maintaining, self-protecting, and self-adapting systems will intersect and create synergies with the proposed research project.

Leadership in the AI Plannng Community: Prof McCluskey was a proposal co-author, and a Network Executive member, of the 60 node *Planet II, European Network of Excellence in AI Planning (2001-2003).* He led the development of Planet's roadmap on KEPS, and co-ordinated Planet's technical unit in this area. He was conference co-chair of the ICAPS 2006, and will be program co-chair of ICAPS 2012 with Brian Williams, Professor of Aeronautics and Astronautics at MIT. He has helped organise and contributed to many ICAPS workshops relevant to this proposal, including in knowledge engineering, verification and validation, machine learning, and applying planning to real world problems. Currently he is organising the UK PlanSIG (the UK's annual workshop on *APS*) at The University of Huddersfield to be held in December 2011.

The University of Edinburgh

Austin Tate (PI): Prof Austin Tate holds the Chair in Knowledge-Based Systems at the University of Edinburgh and is the Director of the Artificial Intelligence Applications Institute at the University. He helped form AIAI in 1984 and since that time has led its efforts to transfer the technologies and methods of artificial intelligence and knowledge systems into commercial, governmental and academic applications throughout the world. He holds degrees in Computer Studies (B.A. Lancaster, 1972) and Machine Intelligence (Ph.D. Edinburgh, 1975). He is a professionally Chartered Engineer.

Prof. Tate is a Fellow of the Royal Society of Edinburgh (Scotland's National Academy) and Fellow of a number or organisations: the Association for the Advancement of AI, European AI, the British Computer Society, the British Interplanetary Society and the International Workflow Management Coalition. He is a Senior Visiting Research Scientist at the Institute of Human Machine Cognition in Florida.

Prof. Tate is an international authority on Knowledge-Based Planning and Activity Management Systems [8, 3, 7] and is involved with industrial and governmental organisations deploying AI technology in the UK, Europe, Japan and the USA. Work has involved Command, Planning and Control for activities such as Non-combatant Evacuation Operations, Air Campaign Planning (including work with the Pentagon), US Army Small Unit Operations, Emergency Response and Disaster Relief. A number of projects in the UK and internationally have involved Search and Rescue Coordination, Personnel Recovery and Multi-national Coalition or Joint Forces Planning and Execution Aids.

Prof. Tate's team is funded by governments and businesses across the world. His research is supported by the US Defense Advanced Research Projects Agency (DARPA), the US Air Force Research Laboratory (Rome, NY), and UK Defence Science Technology Labs (DSTL) amongst other organisations. He has been engaged on some of the leading US Defence Advanced Research Projects Agency (DARPA) funded programs such as Planning Initiative, Agent-Based Computing and Semantic Web programs. He is Edinburgh Principal Investigator for the £7 million 6 year Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration funded in the UK (EPSRC GR/N15764/01). He is a key scientist on the European Union funded OpenKnowledge project, involving some of the top European research groups involved in peer-to-peer agent systems for emergency response. He also led the international Coalition Agent eXperiment (CoAX) project involving some 30 organizations in 4 countries over a 3 year period. He is Chief Technical Officer of I-C2 Systems Ltd. - a company seeking to develop advanced aids for emergency response.

Prof. Tate is on the Senior Advisory Board for the highly-rated IEEE Intelligent Systems journal and is a member of the editorial board of a number of other journals. The grants held by Prof Tate recently include: OpenVCE: Army Research Lab (ARL, US), \$400,000, 2009–2010; FireGrid: Building Research Establishment (BRE) via grant from Department of Trade and Industry (DTI), £2.4 million, £161,835 to AIAI, 2006–2009; OpenKnowledge: 2.3 million Euros, 2006–2008; Slam Games via ITI TechMedia (Scotland), £50,000 consultancy, 2005–2006; Scottish Enterprise IM-PACs POC+ £41,379, 2005–2006; Co-OPR2, DARPA, \$400,000, 2005–2007; Co-OPR, DARPA/SAIC \$350,000, 2004–2005; Scottish Enterprise IM-PACs POC £169,985, 2004–2005; FASTC2AP, DARPA/Global InfoTek \$100,000; CoSAR-TS, DARPA/AFRL \$280,000; CoAKTinG, EPSRC/e-Science, £500,000; Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), EPSRC grant number GR/N15764/01, total value: £7,000,000, Edinburgh value: £1,300,000; I-X and Coalition Agents eXperiment (CoAX), DARPA, £2.6 million enterprise and process modelling project, \$800,000; etc.

Gerhard Wickler: In 1996 Gerhard Wickler moved to Edinburgh to start his PhD studies in the area of AI Planning, which he successfully completed in 1999. He went on to hold research positions in Italy, Belgium, and Germany, working in several areas of AI. Since 2004 he has been senior researcher at the Artificial Intelligence Application Institute (AIAI) within the School of Informatics at the University of Edinburgh, where he teaches the AI Planning course.

Dr. Wickler regularly publishes in AI-related conferences and journals, reporting on his research in AI Planning and Intelligent Agents applied to emergency response [10, 9]. He is an active reviewer for a number of conferences and journals. He has been a member of the programme committees for various workshops and conferences, including the Intelligent Systems track at ISCRAM. He is currently principal investigator and grant holder of an EOARD-funded research project in the area of AI planning and plan execution. In May 2010, he was elected onto the board of directors of the ISCRAM Association and has received the ISCRAM Distinguished Service Award.

References

[1] James Allen, James Hendler, and Austin Tate, editors. *Readings in Planning*. Morgan Kaufman, 1990.

[2] S.N. Cresswell, T.L. McCluskey, and Margaret M. West. Acquiring planning domain models using LOCM. *Knowledge Engineering Review (To Appear)*, 2011.

[3] Ken Currie and Austin Tate. O-Plan: The open planning architeture. *Artificial Intelligence*, 52:49–86, 1991.

[4] T. L. McCluskey and J. M. Porteous. Engineering and Compiling Planning Domain Models to Promote Validity and Efficiency. *Artificial Intelligence*, 95:1–65, 1997.

[5] T. L. McCluskey and M. M. West. The automated refinement of a requirements domain theory. *Journal of Automated Software Enginnering, Special Issue on Inductive Programming*, 8(2):195 – 218, 2001.

[6] R. M. Simpsom, D. E. Kitchin, and T. L. McCluskey. Planning Domain Definition Using GIPO. *The Knowledge Engineering Review*, 22(1), 2007.

[7] A. Tate. The helpful environment: Geographically dispersed intelligent agents that collaborate. *IEEE Intelligent Systems*, 27(3):57–61, 2006.

[8] Austin Tate. Generating project networks. In *Proc. 5th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 888–893. Morgan Kaufmann, 1977. Reprinted in [1, pages 291–296].

[9] G. Wickler. Using planning domain features to facilitate knowledge engineering. In *Proc. KEPS Workshop, ICAPS*, 2011.

[10] Gerhard Wickler, Stephen Potter, Austin Tate, Michal Pěchouček, and Eduard Semsch. Planning and choosing: Augmenting HTN-based agents with mental attitudes. In *Proc. International Conference on Intelligent Agent Technology*, 2007.