Report from Dagstuhl Seminar 17031

Planning and Robotics

Edited by

Malik Ghallab[1], **Nick Hawes**[2], **Daniele Magazzeni**[3],
Brian C. Williams[4], **and Andrea Orlandini**[5]

1 **LAAS – Toulouse, FR,** `malik@laas.fr`
2 **University of Birmingham, GB,** `n.a.hawes@cs.bham.ac.uk`
3 **King's College London, GB,** `daniele.magazzeni@kcl.ac.uk`
4 **MIT – Cambridge, US,** `williams@csail.mit.edu`
5 **CNR – Rome, IT,** `andrea.orlandini@istc.cnr.it`

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 17031 on "Planning and Robotics". The seminar was concerned with the synergy between the research areas of *Automated Planning & Scheduling* and *Robotics*. The motivation for this seminar was to bring together researchers from the two communities and people from the Industry in order to foster a broader interest in the integration of planning and deliberation approaches to sensory-motor functions in robotics. The first part of the seminar was dedicated to eight sessions composed on several topics in which attendees had the opportunity to present position statements. Then, the second part was composed by six panel sessions where attendees had the opportunity to further discuss the position statements and issues raised in previous sessions. The main outcomes were a greater common understanding of planning and robotics issues and challenges, and a greater appreciation of crossover between different perspectives, i.e., spanning from low level control to high-level cognitive approaches for autonomous robots. Different application domains were also discussed in which the deployment of planning and robotics methodologies and technologies constitute an added value.

Seminar January 15–20, 2017 – http://www.dagstuhl.de/17031
1998 ACM Subject Classification I.2 Artificial Intelligence, I.2.9 Robotics
Keywords and phrases adjustable autonomy, artificial intelligence, automated planning and scheduling, goal reasoning, human-robot interaction, plan execution, robotics, robust autonomy
Digital Object Identifier 10.4230/DagRep.7.1.32

1 Executive Summary

Andrea Orlandini
Malik Ghallab
Nick Hawes
Daniele Magazzeni
Brian C. Williams

Automated Planning and Scheduling (P&S) and Robotics were strongly connected in the early days of A.I., but became mostly disconnected later on. Indeed, Robotics is one of the most appealing and natural application area for the P&S research community, however such a natural interest seems to not be reflected by advances beyond the state-of-the-art in P&S

Planning and Robotics, *Dagstuhl Reports*, Vol. 7, Issue 1, pp. 32–73
Editors: Malik Ghallab, Nick Hawes, Daniele Magazzeni, Brian C. Williams, and Andrea Orlandini
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

research in Robotics applications. In light of the accelerated progress and the growth of economic importance of advanced robotics technology, it is essential for the P&S community to respond to the challenges that these applications pose and contribute to the advance of intelligent robotics.

In this perspective, a Planning and Robotics (PlanRob) initiative within the P&S research community has been recently started with a twofold aim. On the one hand, this initiative would constitute a fresh impulse for the P&S community to develop its interests and efforts towards the Robotics research area. On the other hand, it aims at attracting representatives from the Robotics community to discuss their challenges related to planning for autonomous robots (deliberative, reactive, continuous planning and execution etc.) as well as their expectations from the P&S community. The PlanRob initiative was initiated as a workshop series (http://pst.istc.cnr.it/planrob/) started at the International Conference on Automated Planning and Scheduling (ICAPS) in 2013. The PlanRob workshop editions gathered very good feedback from both the P&S and Robotics communities. And this resulted also in the organisation of a specific Robotics Track at ICAPS since 2014.

The aim of this Dagstuhl Seminar was to reinforce such initiative and increase the synergy between these two research communities. Then, most of the attendees contributed with position statements (whose abstracts are available in this report) to present their major challenges and approaches for addressing them. In general, this involved sharing views, thoughts and contributions across the following main topics:

- **Long-term autonomy / Open world planning**, providing an overview on issues related to continuous planning for robots with partial information or even incomplete models;
- **Knowledge Representation and Reasoning in Planning**, with presentations on cognitive features and robot planning;
- **Challenges in Industrial, Logistics & Consumer Robotics**, providing relevant insights related to deployment of robots in real world scenarios;
- **Human-Robot Planning**, with a wide overview on planning solutions for dealing with interactions between humans and robots;
- **Planning and Execution**, discussing issues and challenges related to robust planning and execution for robot control;
- **Task & Motion Planning / Hybrid planners**, with presentations on integrated solutions for robot control at different levels;
- **Reliable and Safe Planning for Robotics**, providing an overview of ISO standards for robots and, more in general, investigating the exploitation of formal methods to guarantee reliability in robotic applications;
- **Technological Issues in Robot planning/Multi-robot Planning**, with statements on technological issues in (multi-) robot solutions.

Each session was animated by (i) an opponent, whose role was to be critical about the position statements and (ii) a moderator, to organise the discussion. Therefore, opponents and moderators have provided a short summary of the session ideas and discussion in dedicated Synthesis Sessions to further foster the discussion.

In addition, two panel sessions have been organised on (i) **Evaluation, Benchmarking and Competitions**, discussing the experience in RoboCup@Home and the organisation of the new Planning and Execution competition (that will be held in 2017), and (ii) **Outreach & Training**, discussing about the possible organisation of summer schools and the opening of new scientific networking initiatives (e.g., a COST action).

During the seminar, discussions focused on different issues, challenges, possible solutions and new promising trends over a very wide variety of relevant topics: knowledge representation, modelling issues, the need of incomplete models; cognitive features such as, for instance, learning and goal reasoning; human-aware solutions for flexible human-robot interaction; adaptive solutions for human-robot collaboration; robust execution capable of effectively dealing with failure; integration issues in robotic architecture that, e.g., exploit different kind of models and then perform hybrid reasoning; application of formal methods to provide verification and validation functionalities to guarantee reliable robotic systems; etc. Indeed, addressing the integration of P&S and Robotics for development of intelligent robots entails covering a heterogeneous spectrum of problems, often requiring complex solutions that require a vast set of knowledge and technologies.

During the seminar, there was a very high level of engagement and interaction between the participants, enabling a lively and productive week. The main outcome of the seminar was to share a common understanding of issues and solutions with thorough discussions. And the workshop ended with an open discussion on possible follow ups and possible actions to create further opportunities for fostering synergies and interactions between the two communities.

2 Table of Contents

Overview of Talks

3 Overview of Talks

3.1 Joint Human-Robot Activity is a context and a challenge for pertinent investigation in Automated Planning

Rachid Alami (LAAS – Toulouse, FR)

Let us consider what should be the planning abilities for a robot that has to share Task and Space with a Human partner. Planning and more generally on-line deliberation is clearly a necessary ability since it allows the robot to reason on action and situation consequences, to anticipate or to act pro-actively.

This is a task oriented problem. The question can be expressed as "How to perform a task, in presence or in interaction with humans, in the best possible way i.e. taking into account safety and efficiency but also acceptability of robot behaviour by the humans and legibility of robot intentions".

The robot has to build and manage "Shared Plans" involving Humans and itself. Besides the criteri mentioned above, the models should integrate the key notion of predicting and reasoning about human mental state as well as human preferences. Based on this, the problem is not only to build a plan for a robot which collaborates with humans but to build a "sufficiently good" plan that answers satisfactorily, a various levels of abstraction, the questions: what, who, where, when, how? Our aim is to discuss the issues mentioned above and illustrate them based on preliminary results that not only give some concrete examples of human-aware task and motion planning but also how they can fit in a coherent architecture for a cognitive and interactive robot.

3.2 A Blueprint for the Evolution of Perspectives: Planning Technology as a Basis for the Mass Customization of Robots

Iman Awaad (Hochschule Bonn-Rhein-Sieg – St. Augustin, DE)

As the field of robotics expands beyond a critical mass with a view to advancing the public interest, a change in paradigm is needed that transfers the complexity of customizing some functions from those with deep technical competence to the users – a process of mass customization.

Allowing for customization of functionality will enable users to adjust the functions for which they acquired the robot to their own needs and biases, thus enabling it to serve as an extension to their own capacity, rather than of what the manufacturer might perceive to be a standard set of customers' capacities and needs. This ability to customize the functionality would inherently necessitate a transparent interaction that explains what the agent is doing and why, and enables the user to modify behavior by specifying preferences, contexts, and rules (what to do, what not to do and when). The user's own explanations may well play a role in specifying such knowledge. Given that planning technology is responsible for the decision making process that determines the behavior of the robot (plan-based robot control), it is perfectly placed to play a central role in this customization process by enabling end-users to directly customize and even create the planning domains that are used by the robots via other ubiquitous tools (such as tablets and mobile communication devices).

This is just one of many aspects that will need investigation in the process of scaling up robotic ubiquity, alongside issues such as legal/regulatory implications, embedded cultural bias and social acceptance, ethical ramifications, etc. (Looking at the two simple examples of diffusion of drones and autonomous vehicles already exemplifies many of these factors). One extra point is that now all "parameters of the robot's autonomy" are defined by the manufacturers, some of these parameters would need to be transferred to the customer/user. This would apply to aspects that relate to personal/cultural/practical/social preferences. (Even McDonalds "localizes", and offers veggie burgers in Hindu regions. In this case, we can't talk just of regional localization but individual customization, because each household has its own individuality and potential preferences for how the autonomy of a robot is manifested.) Perhaps, by keeping in mind this goal of enabling the customization (whether by learning, or by parametrization, and so on..), we may also find that we as developers have created toolsets and modalities that simplify the process of adding and changing the functionality of robots. More needs to be done within the community to speed up the development process and remove the extensive barriers to entry that currently exist. The sharing of best practices, lessons learned, solutions (which should be developed with re-use in mind), and even raw data sets would be a start in this direction. The creation of a central repository for the various application domains that makes available specifications of planning domains, tasks, actions, preferences, agendas, and context, in whatever representations they were formulated in is a worthwhile endeavor. Cooperation is important for interoperability but also for transboundary regional policy-making (e.g. Uber).

Finally, we need to be aware of (and exploit) technological innovation that is developing rapidly in parallel, and that may or may not accelerate or help shape the trajectory of autonomous Position paper for Dagstuhl Seminar 17031 Planning and Robotics (PlanRob) robots. For example, in the same way that the community has benefitted from technologies that were initially developed for the mobile communications market, we should capitalize on the technology (and standards) that have been developed for the Internet of Things and other innovations that are yet to appear. Similarly, in the same way that the development of selfdriving cars resulted from the cooperation between the community and the automotive industry, a similar cooperation with architects and home furnishing companies could go a long way in injecting the much sought-after structure in home environments and providing (both static and possibly dynamic) knowledge of these objects and environments to the planning and acting processes.

3.3 Towards Autonomous Robots via Technology Integration

Roman Bartak (Charles University – Prague, CZ)

We share the challenge of developing autonomous robots that can do anything that their hardware allows them to do. The ultimate testbed is a robot that can do any task that a human can do when remotely controlling the robot. The idea is learning how people are performing the task with a robot and then "programming" the robot to be able to solve the same (and similar) tasks hopefully in a more general setting. To fulfill this vision one needs expertise in many areas including control theory, computer vision, localization, path finding, activity planning, knowledge representation, machine learning, etc. Despite progress in all these areas separately, there are still big gaps between them that prevent efficient exploitation of research results to build advanced integrated systems such as autonomous robots.

Robotics today is very separated from Artificial Intelligence (and activity planning). We can identify several gaps there such as symbolic vs numeric reasoning and model-free vs modelbased methods. We believe that different approaches are better for different settings and hence it is more appropriate to find a way how to integrate them rather than preferring one approach over the other one to handle all the problems. Activity planning is a symbolic modelbased approach while robotics is based more on numeric model-free techniques. Many problems arise from the clash between these different worlds. How can we obtain the symbolic model necessary to apply planning techniques for a particular robotic hardware? Which modelling framework is appropriate to provide necessary expressivity and efficiency? How to formulate the planning goal based on the current state of the system? When (re-) planning should be initiated? How does the plan convert to executable instructions? How does the sense-act approach fit the plan-execute approach? Etc. In our research, we develop model-centric techniques with the focus on planning domain models that are efficient for problem solving. We use flying drones as a robotic platform because they are "kinetically" simple (opposite for example to robotic hands) – the drone can only fly and observe. Still, the drones can solve interesting practical tasks such as mapping, inspection, search, tracking, delivery etc. Our focus is on software for controlling the drone and for information processing rather than on hardware, which is a "standard platform". This is based on idea that current hardware is advanced enough to perform complex tasks but the weak part is software that controls it. Hence we believe that AI will play a more significant role in robotics in upcoming years.

The first question is what are the symbolic activities to be used in activity planning for robots. We are trying to identify activities as somehow homogenous behaviors using machine learning techniques (such as clustering) applied to sensor (and control) data obtained from a drone when being manually controlled. Currently, we do not use the camera as a sensor, but it would be very interesting to exploit computer vision techniques as a source of extra sensor inputs (very rich inputs in this case). The next step is, for such activities, finding some formal description that can be parameterized (for example flying forward for a specific distance) and finding a controller for executing such activities. This way we are trying to bridge the continuous (numeric) world of robots with the symbolic world of planning. Having the activities, the next step is finding a way of efficient planning with them. PDDL planning is based on the "flat" structure of activities with no extra control knowledge. Despite a huge progress in domain-independent planning, PDDL planners are still hardly applied to practical problems due to efficiency problems. There exist modelling frameworks such as hierarchical task networks and control rules to guide the planners, but it is cumbersome to obtain such models. We are going in the direction of recipe-based planning models where the causal structures of activities can be learnt by observing how the robot solves a specific task while being controlled by a human (activities need to be detected first from sensor data). Hierarchical structures seem desirable there to get better flexibility via having reusable tasks that decompose to simpler sub-tasks. Getting experience from linguistics, namely formal grammars that can describe hierarchical structures, might be beneficial there thanks to exiting support tools for formal grammars, for example, allowing one to do formal reasoning with the models such as verification. This is an active research topic where technology developed for one area (natural language processing) can be exploited in a very different area (activity planning). Though we are addressing automated techniques to obtain and use the activity models, we also see a big gap in authoring tools for developing control software for robotic platforms. Frameworks such as ROS simplified transfer of tools between robotic platforms, but using ROS still requires non-trivial knowledge and low-level programming skills that makes it hard to program "standard" robotic platforms for specific tasks. We believe that the above-described approach of using symbolic activities (directly executable by a robotic platform) that are connected via recipes for performing specific tasks can simplify development of robotic software. Visual programming languages and systems such as Ozobot are good motivation there. They can be used to manually describe recipes (plans) to solve specific tasks as well as to visualize automatically-learned recipes. The challenge is how to go beyond the models for classical sequential execution of activities to more flexible reactive models that are still easy to understand. The major reason for having some formalism for activity models is the need to verify such models before they can be used in industrial setting. In summary, we see symbolic models as a way to simplify development of robotic software. These models need to be tightly connected to control software that is based more on numeric techniques. Hence integration of various technologies is necessary there.

3.4 Robot Planning for the mastery of human-scale everyday manipulation tasks

Michael Beetz (Universität Bremen, DE)

Robot planning can be considered as the reasoning about the future execution of robot programs (plans) in order to optimize their performance in terms of achieving their goals and efficiency (McDermott). The holy grail of robot (action) planning ever since the Shakey project has been to equip robotic agents with human-level (manipulation) action capabilities. Unfortunately, the progress along this dimension has been modest at best. I believe that much of the lack of progress is caused by the way the research field of task planning abstracts (robot) actions (see PDDL). It makes the assumption that reasoning about abstract preconditions and effects is sufficient for planning complex manipulation tasks. If we interpret this assumption from a probabilistic point of view, we can restate it by asserting that the probability of achieving the desired effects of actions is conditionally independent of how the robot executes the actions given that the preconditions of the actions are satisfied. This means that our robot action planning systems would not change their belief about whether an action is executed successfully depending on whether the robot plans to grasp an object with one hand or two, which grasp type it applies, and so on. Or, if a fetch action is executed by two-year old or an experienced waiter. In contrast our experience with realizing human-scale manipulation activities for robotic agents shows that most of the intelligent problem-solving capabilities of robots are needed in order to decide how to execute the actions to make them succeed, that is to achieve the desired effects of an action and avoid the undesired ones.

I believe that in order to materialize the impact that robot planning technology can have for robotic agents that are to accomplish human-scale manipulation activities, we have to extend our representation and reasoning machanisms to include the concepts of motor cognition. Motor cognition is a discipline in cognitive psychology of action which is concerned with the learning, reasoning, and planning of how to parameterize and synchronize motions in order to accomplish actions. I foresee a new generation of powerful robot planning systems that do not only reason symbolically about their actions but also subsymbolically with their "eyes and hands". Today's disruptive technologies, in particular modern game technology, physics simulation, data analytics, and deep learning give us the opportunities to pursue this direction.

3.5 Plug&Play Autonomous Robots

Ronen I. Brafman (Ben Gurion University – Beer Sheva, IL)

Robotics today is reminiscent in many ways of personal computing in the early 80s. Some key industrial applications, various toy applications, and even more difficult to use than DOS. One of the keys to more powerful computing, and to more useful robotics, is the ability to easily integrate new software and new hardware without having to configure them manually. The difficulty in robotics is even greater as new capabilities can interact in complex ways not only internally, but also externally. Beyond this, we need simple ways of getting robots to do what we wish, and writing dedicated scripts each time is not a good solution. One of the key software-engineering challenges for robotics is to facilitate a world in which it is easy and safe to integrate new capabilities into a robotic platform. We believe AI, and AI planning in particular, provides some of the key ideas for addressing this challenge.

We are trying to address this need by developing formal, machine readable and actionable "robot-capability description language" – essentially, a rich action description language that replaces semi-formal software engineering formal techniques, and is closely related to efforts to standardize the specification of web-services. Essentially, we argue that action description languages should be elevated to the status of function specifications. We are aware of the existence of formal methods for programming robots that provide powerful tools for writing code with behavior guarantees. Yet, we fear that these will be confined to the small community of researchers working on them, as there is a large, and likely to be growing community of users that are continuously contributing useful robotic code, albeit one written using standard programming techniques and with standard tools. Providing a formal specification of the properties of this code seems much easier and more realistic than rewriting this code from scratch, and amenable, to some extent, to automation.

If every functional module has an associated formal description of its normal behavior, it is easy to provide added value services that

1. Monitor its performance and alerts of any abnormal situation,
2. Improve the model based on actual experience,
3. Verifies controllers that combine existing modules, and provides information about their probably effects,
4. Combines existing modules automatically,

– and probably additional added value services that would be developed in the future.

To this effect, we have developed a rich XML-based specification language that contains four classes of functions: achieve, maintain, observe, and detect, tools for generating monitoring code, and tools for generating automated interfaces between this code and the ROSPlan system.

In our work we continuously discover new desirable features, and we anticipate that this will continue to be the case, and hope to join efforts with other in providing the "right" specification language and tools that exploit it.

3.6 Planning with ROS

Michael Cashmore (King's College London, GB)

We are working towards integrating Hybrid Systems Planners with real-world systems. This involves a number of challenging and interesting questions. Given real-valued and non-linear functions within the planning process, what new kinds of models can be explored?

In each robotic system there is a decision as to what should be included in the planner's model, and what should be handled by a specialised component. This question, and the way in which external planning tools are connected together presents as many interesting problems as the modelling decisions themselves.

All too often the planner is seen by non-experts as a black box. It is expected to produce a behaviour that is already scripted. To use a general planner in a robotic system in a more interesting way, a large amount of integration takes place around it. Consider the following requirements sketch for a planning robot:

- The models used by the system are generated automatically:
  - This includes components for state estimation, state prediction, and abstraction.
  - These build a model of the current environment in the language of the planner.
  - The long-term goals of the robot, are either provided by hand or driven by the robot's motivations.
  - A model of the robot's capabilities are generated from a formal description of the hardware.
- The planning takes place at multiple levels of abstraction, starting with long-horizon strategic plans, which gradually are refined into short-horizon task plans.
- Plans generated by the planner are executed robustly with some prior preprocessing:
  - They are pre-processed into a structure that explicitly contains plan failure conditions and causal links (which are not often included in planner output).
  - They are also translated into a structure that has some formal guarantees on controllability; or are combined with execution rules. The resultant plan could better be described as a controller.
- Even using the most advanced planning techniques for uncertainty, the robotic system reliably deviates from the planner's model. Action or plan failure that occurs as a result is detected, and also repaired in a way that does not unwittingly affect other ongoing plans and processes.
- The robot interacts with humans and so:
  - A component is included for plan legibility, so that a user can understand that the robot lives between and outside of scripted and broken behaviour.
  - The user is able to modify the long term goals and behaviour of the robot.
  - The user is able to interact within the confines of planned behaviour; assisting the robot, being assisted, and communicating.

All of the above could arguably be included within the black box of the planning system. I am interested in exploring which components are essential in any "planning" robotic system, which are optional, and which can be replaced by equivalent components.

When engaging in the task of integrating components, the result is often the minimal functioning system. I am also interested in providing a generic architecture for linking those essential components that will facilitate the easy use of existing libraries in new systems, opening up the black box of planning to general (ROS) users.

3.7 Teach Once Logistics Perspective

Martin Davies (Guidance Automation Ltd – Leicester, GB)

Any repetitive task is ripe for automation. Since the invention of the mechanized weaving loom in the late 1700's, the mechanization and subsequent automation of manufacturing practices has revolved around three steps:

Firstly, the identification of a suitable process for completing a stage of the manufacturing process. Requiring skilled humans, throughput is low and cost is high.

Secondly, process optimization reduces variation and errors. The introduction of jigs, fixtures, an SOP (standard operating procedure) etc. de-skills the task and reduces product variance.

Finally, once the process is tested and rigid, the now dominant cost factor, the human element is removed. Automation is brought in, increasing profitability and securing market dominance.

The concrete example of this is an automotive production line. High numbers of identical products are manufactured for a number of years with minimal variance. The line consists of a number of cells, with each cell designed for a specific task. When the product line is refreshed, the factory is shutdown, the cells are reconfigured and robots within cells are re-taught via manual operation. They will be re-taught in less than a day and will follow that teaching for a number of years before being reconfigured. Teach once. Repeat infinite.

Robotics now is about replacing humans directly in dynamic environments. Guidance Automation automates and provides scheduling software for forklift automated guided vehicles (AGV's) in logistics environments. We are at the "third" stage in the manufacturing process, but the application of the technology is overly complex and unwieldy.

Consider now the installation of a fleet of forklift AGV's. We have to map the environment, and potentially install some form of navigational aids. We then have to align the map that the AGV's will navigate to CAD, enabling us to have AGV's that can navigate the environment in a frame of reference common to the existing warehouse management system (WHM).

Now we have to link all stock locations in the warehouse to physical locations in order to schedule the AGV's. This requires mapping and labelling all shelves, which currently may only be in a human readable format.

Then there is a large amount of scripting and teaching to be done, determination of locations to pick and drop pallets. Environmental variance is high, shelves may be at different heights, scripts cannot be cut and pasted. System operation must be guaranteed without the requirement to teach every pick location.

Finally, we also have to handle the inability to see all objects. Paper hanging down from pallets, obscuring shelves and markings. The process is rigid for deskilled human operators, but too flexible for robotic undertaking.

My question is how do we apply the teachings of the manufacturing industry to the logistics environment. Should our focus be on robot evolution or is the logistics problem too unconstrained to solve efficiently? How can we as robotics experts influence the logistics process enabling us to make it teach once.

3.8 Learning Spatial Models for Navigation

Susan L. Epstein (City University of New York, US)

Deliberative robot navigation architectures often model the world as a detailed metric map. Given a target destination, the robot constructs an optimal plan within the map and then executes it. Realistically, however, doors open and close, and people (or other robots) move rapidly about. In such dynamic worlds, a map identifies only static obstructions. Thus, plan-based navigation requires plan repair and often re-planning.

Our approach, SemaFORR, is intended for autonomous indoor navigation, where maps are unreliable or unavailable, and landmarks may be absent, obscured, or obliterated: complex office buildings, warehouses, and search-and-rescue settings. Rather than respond only to percepts and known obstructions, we have chosen to learn spatial affordances, spatial abstractions that facilitate movement and represent a robot's experience of the world. Our thesis is that spatial affordances learned from local sensing during travel can both support effective, autonomous robot navigation and provide a lingua franca for dialogue with a human traveling companion. SemaFORR's spatial affordances include unobstructed areas, useful transit points, route segments, doors, and passageways. Together they form a spatial model that represents the robot's world but is not a metric map. SemaFORR has rapidly learned spatial models that support efficient travel in a variety of simulated two-dimensional worlds. That approach was purely reactive, however, without recourse to a map or a planner.

SemaFORR can learn a spatial model either from its percepts as it navigates or in simulation on a map of its environment. Current work includes the construction of ROS-based SemaFORR modules parameterized for a variety of real-world robot platforms. Work is also underway to adapt SemaFORR for movement through crowd models based on well-documented human behaviors. We will extend work on movement toward and through crowds to real-world environments, where we can test a variety of human-robot interactions.

Thus we envision SemaFORR as a collaborator in navigation in two ways:

- As a companion to SLAM: Current development includes classical planning in a traditional metric map, novel planners in the spatial model, and techniques to integrate them. We expect that a plan derived in SemaFORR's spatial model will prove more flexible than those of traditional map-based planners.
- As a companion to a human traveller: Recent work in cognitive neuroscience (including the 2014 Nobel Prize in Physiology or Medicine) has detected place cells, grid cells, and direction mechanisms in mammalian brains that have strong analogies to our spatial affordances. Thus we believe that SemaFORR is a strong foundation for dialogue with a human traveling companion about decisions and the nature of the environment. The

user-friendly qualities of SemaFORR's spatial model and the simplicity of its reasoning structure provide a natural common ground within which to discuss which way to travel and why.

3.9 Flexible Execution of Human-Robot Collaborative Plans: a cognitive control

Alberto Finzi (University of Naples, IT)

In social and service robotics, complex collaborative plans should be executed while interacting with humans in a natural and fluent manner. Indeed, a robotic system is often provided with structured tasks to be accomplished; on the other hand, this execution should be continuously adapted to the human activities, commands, and interventions. In these scenarios, the human interaction is unpredictable and very complex (multimodal, verbal and non-verbal, either explicit or implicit, etc.), therefore, several mechanisms should be supported, such as human state/activity/intention recognition, joint attention, attention manipulation, referencing, turn-taking, action coordination, dialogue management.

Different frameworks have been proposed in the robotics literature to conciliate natural human-robot interaction and the execution of complex cooperative plans. The dominant approach relies on the planning and execution paradigm and deploys replanning to adapt task execution to the behaviors of the agents involved in the interaction. This paradigm is effective in mixed-initiative planning and execution, however, the associated continuous planning/replanning process usually impairs the naturalness and effectiveness of the interaction with the humans and the environment.

We propose to tackle these issues from a different perspective exploiting the concept of cognitive control introduced in cognitive psychology and neuroscience to describe the executive mechanisms/functions needed to support flexible, adaptive responses and complex goal-directed cognitive processes and behaviors. Inspired by this literature, we propose to deploy a supervisory attentional system paradigm [Norman Shallice 1986]. In this framework, executive attention plays a crucial rule. Indeed, the supervisory attentional system coordinates and monitors hierarchically organized behavioral schemata exploiting attentional regulations to facilitate the execution of desired processes, while inhibiting the inappropriate ones. This paradigm seems particularly relevant not only for flexible plan execution, but also for human-robot interaction, because it directly provides attentional mechanisms (attention manipulation, joint attention, action facilitation, habituation, etc.) considered as pivotal for implicit, non-verbal human-human communication [Tomasello 2008].

Following this approach, we propose and discuss an interactive framework that combines human-aware planning, flexible and interactive plan execution, human monitoring, multimodal interaction, and task teaching. In this setting, a cooperative plan is considered as an attentional guidance for an attentional executive system influenced by the human actions and the environmental changes. Finally, we discuss how the proposed framework can support not only flexible and interactive execution of structured tasks, but also incremental task adaptation through teaching by demonstration.

3.10 Combined Task and Motion Planning is Classical Planning

Hector Geffner (UPF – Barcelona, ES)

Robot planning is a broad area. I focus here on what's called "combined task and motion planning". Some approaches split the problem into two, task and motion planning, that are ad- dressed by two types of planning algorithms. Such a decomposition however tends to be ineffective as the two components are not independent. More recent approaches have aimed at exploiting the efficiency of modern classical planners, either by taking the spatial constraints into account as part of a symbolic, goal-directed replanning process [7], or by using geometrical information in the computation of the classical planner heuristic [3]. My position is that combined task and motion planning (CTMP) is classical planning, and that it may pay to address the problem in this way. What is classical planning? It's planning from a known initial state using deterministic actions with known effects for achieving a goal state. It is assumed that the state space is discrete and finite, and given in compact form as the values of a set of variables whose values are changed by the actions. The first obstacle that needs to be overcome in order to formulate and solve CTMP as classical planning is that the space of robot and object configurations is not finite or discrete. Yet, it's common for such configuration spaces to be discretized by means of probabilistic sampling schemes [5]. The second challenge is the limitation of existing classical planners for modeling CTMP problems even when discretized. It's not clear indeed how to express for example that "spatial collisions" are to be avoided in STRIPS-like languages without ending up with huge encodings. The third challenge is that, even if one develops a suitable planning language for modeling discretized CTMP as a classical planning problem, there may be no effective planners for dealing with such a language, nor efficient ways for translating it into one that can be handled by modern planners like LAMA. Yet these are all limitations of current classical planners, not of classical planning that is supposed to deal with sequential decision problems involving deterministic actions and a fully known initial state. Moreover, these limitations have little to do with robotics. For example, the Atari video-games, and many of the games of the General-Video AI game competition are classical planning problems that cannot be addressed by the standard classical planners. Indeed there is no PDDL encodings for such problems but just a simulator. In the last few years, we have developed expressive planning languages [2] and classical algorithms that can effectively plan with such languages and with simulators [6, 4]. More recently, we have shown how these ideas can be applied to CMTP where problems involving tens of objects and a PR2 robot can be fully compiled and solved as classical planning problems [1].

References

1 J. Ferrer, G. Frances, and H. Geffner. Submitted. Videos at http://bit.ly/2fnXeAd, 2016.
2 G. Frances and H. Geffner. *Modeling and computation in planning.* In Proc. ICAPS, 2015.
3 C. Garrett, T. Lozano-Perez, and L. Kaelbling. *FFRob: An efficient heuristic for task and motion planning.* In Algorithmic Foundations of Robotics XI, pages 179–195. Springer, 2015.
4 T. Geffner and H. Geffner. *Width-based planning for general video-game playing.* In Proc. AIIDE-2015, 2015.
5 S. M. LaValle. *Planning algorithms.* Cambridge, 2006.
6 N. Lipovetzky, M. Ramirez, and H. Geffner. *Classical planning with simulators: Results on atari.* In Proc. IJCAI, 2015.

7 S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel. *Combined task and motion planning through an extensible planner-independent interface layer*. In Proc. ICRA, pages 639–646, 2014.

3.11 Planning for Long-Term Robot Autonomy

Nick Hawes (University of Birmingham, GB)

An open problem within the use of planning technologies on robots is the problem of planning for long--term autonomy. There are at least two challenges within this problem. The first is that within a long--term autonomous robot, planning may never formally start and end. Instead the robot should maintain a plan which achieves goals for some future time horizon, where that time horizon is only part of a longer--term schedule of goal--driven behaviour. For example the robot may have a known list of goals which it should achieve that day, or that week, and must also respond to goals provided to it in an on--demand fashion. It also must manage it's limited resources (notably battery and time) to ensure that it is able to achieve all its goals in its future, not just the ones in the time horizon. Finally it must also be able to deal with the inevitable failures and unexpected consequences of operating in the real world. This challenge brings together planning and scheduling along with prior work on oversubscription planning, continual planning and goal--driven autonomy. The second challenge within planning for long--term autonomy is being able to automatically generate planning domains, environment models etc. in such a way that they capture the experience of the robot (in plan execution, and of the environment more broadly) over the long time periods it operates for. This will allow for planning models which better match reality, resulting in better performing robots and fewer failures at execution time.

3.12 Plan-based robot control

Joachim Hertzberg (Universität Osnabrück, DE)

Robotics and AI planning are both in a healthy state, as we all know. A deep integration of the two is open in many respects. In my view, the depth of integration would increase with the number and caliber of processes of robot control, on the one side, and planning, on the other, that run in closed loop – and with closed loop, I mean that they both take input from the respective other one and generate output for the respective other one (which this one takes as input). To make such a closed-loop integration possible, requires a deep understanding of what happens on both ends of the loop, and requires deep integration in terms of representation formalisms, representation granularity, control granularity, and, of course, interfaces. I will name three issues, which are intertwined, where I see room as well as the need for improvement.

Execution monitoring

The classic. As long as a course of action runs perfectly as planned and as envisaged by the planning domain model, all is good. As soon as it deviates from this nominal course, we have little to say about it. This starts with the problem of recognizing in the first place when deviation starts. We may represent time lines about state variables to include timing behavior of actions – this allows to detect delays, but not every delay is a fatal deviation. Even if we determine that something has gone wrong with executing some action or plan, then what was the cause of the fault? AI planning is strong in modeling abstractly nominal courses of action; plan-based robot control needs deep models of the environment and its dynamics that allow non-nominal developments to be understood, too. To make it efficient, both should probably work in the same representation framework.

Semantic perception

Interpretation of the data flow from a configuration of robot sensors is, in the utmost of cases and methods, understood as a process of bottom-up aggregation and abstraction from sensor data "upward" to symbols, and eventually into pieces or sentences in a representation language. That is good, but it is just one part of the story. To have its knowledge influence the action of the robot efficiently, the inverse process is needed: the priming by context, as determined by reasoning about the knowledge about the current situation, of the act of perceiving. This starts from directing the sensors and the sensor data processing resources to salient spots or events in the environment, continues over discarding much if not most of the raw sensor data deemed uninteresting, and goes into interpreting the salient data in the current context according to the current needs. Sensor data – be they single still images or full ROS bags – don't hit us like rainfall. As part of robot control, they are, or should be, actively acquired, to a large extent.

Dealing with huge, flawed, and deficient bodies of knowledge

Knowledge in many robot domains is huge (think of all that needs to be known about an office building for a courier robot), facts are subject to change independent of robot actions (think of all that goes on in an office building outside of the robot's control), and the robot can impossibly know all that is the case in its world, even though it may once become relevant for its action (again, think of all that is the case in an office building). Yet, it has to get along with what it knows, as good as it knows it. AI has it since a long time that reasoning is defeasible. Robotics tells you: this is the norm rather than an exotic exception in a robot's knowledge base; whatever reasoning is used has to function with huge knowledge bases that contain large numbers of flaws and gaps. What is a formalism and a calculus to cope with that? What additional robot tasks are needed for making its precious knowledge base sustainable in spite of these flaws and gaps? As far as I know, no one knows.

3.13 Flexible Planning for HRI

Laura M. Hiatt (Naval Research Lab – Washington, DC, US) and Mark Roberts (Naval Research – Washington, US)

As robots become more pervasive in our daily lives, it becomes more important to be able to interact with them, and task them, naturally and spontaneously. One major hindrance to achieving this goal is a lack of flexibility in how robots can execute tasks and interact with the world. For example, if a robot that is carrying a tool to a teammate is asked by another human to help them hold open a door, the robot should conceptually be able to help. In most current robotic planning systems, however, unless such a scenario was specifically foreseen and engineered, this level of flexibility is not possible: the robot would either have to prematurely end the tool task, or deliver the tool and then return to help with the door. This both hinders overall robot performance and, we argue, decreases the quality of interactions between robots and human partners.

We are addressing this problem by beginning to investigate how different robotic tasks can be concurrently executed in an ad hoc fashion, even if they utilize overlapping resources on the robot (such as the same arm). One of the key questions of this concurrency is how to ensure the correctness of the combined execution of the tasks. In our approach, we address correctness by enabling tasks to specify constraints that other executing tasks must observe when executing concurrently. Returning to the earlier example, the robot carrying the tool to the teammate may specify that its arm can move around or be used in another task as long as the tool stays 3 inches away from any object. Both goal reasoning and planning algorithms must then be extended to support these constraints, allowing reasoning about domains not only where multiple tasks can execute at once, but also where the tasks can physically affect one another.

It is our belief that giving robots this additional level of flexibility will both increase their overall functionality, as well as increase their ability to naturally team with human partners.

3.14 Safety Reconsidered – planning for safe human-robot collaboration

Michael W. Hofbaur (Joanneum Research – Klagenfurt/Wörthersee, AT)

Real world applications of human-robot collaboration require conformity to relevant standards. For robotic manipulators, for example, it is obligatory to operate the machines according to the guidelines defined in ISO 10218 “Robots and robotic devices – Safety requirements for industrial robots” and ISO 15055 “Robots and robotic devices – Collaborative robots”. These standards introduce a framework for safety that is conceptually different from the safety concept that is typically used in computer science, AI and planning, in particular.

For example, ISO 15066 defines specific operational modes that restrict the functionality of a robot to physically safe manipulation operations. These limitations often lead to impractical robot applications (e.g. too slow, inadequate force / torque capabilities for real-world applications, etc.). Using environmental perception and task- / situation-aware high level control using advanced planning and scheduling could significantly improve the robot's capabilities and enable new applications of human-robot collaboration. However, this will also require the planning component to obey certain practical requirements, such as coding-standards and architectural-considerations and guaranteed dependability levels.

Our impulse talk will therefore consider safety concepts from both perspectives and sketches possible future directions for planning and scheduling in high-level control systems of collaborative robots that enable safe autonomous behaviors of these machines.

3.15 Conditional Planning for Human-Robot Interaction

Luca Iocchi (Sapienza University of Rome, IT)

Service robots interacting with people in home or public environments are required to execute many different tasks, including various forms of interaction with the users. These actions typically depends on the needs or requests of the users and generating all the possible combinations in advance and manually is a too demanding task. Moreover, when interacting with naive users, the robot has to be robust to many unexpected situations. Finally, the assumption that a robot is always able to have perfect knowledge about the situation is too unrealistic and plans must be robust to imperfect and noisy perception.

In this scenarios, automated planning procedures are very useful to generate many interactions with a compact representation of the domain, resulting in less effort for the designer of the system and better performance of the overall task.

However, the most common standard planning techniques present the following issues:

1. classical planning assumes perfect knowledge and perfect action execution;
2. replanning after failures assumes perfect sensing (to detect failures and to determine the new initial state for replanning);
3. Markov Decision Processes (MDP) require perfect observability of the state;
4. Partially Observable Markov Decision Processes (POMDP) allow for modelling perception uncertainty, but efficiency of the algorithms does not scale with the complexity of the problem and determining correct probability values is a difficult task.

Conditional planning instead has nice features in this scenario: (i) it is based on explicit sensing, thus perception is limited to the execution of such sensing actions (rather than passively used to determine any state); (ii) only partial knowledge about the initial state is required; (iii) conditional planners are efficient. In other words, conditional planning allows for execution of minimal sensing procedures that minimizes the risk of wrong execution of the plans due to wrong perceptions.

However, conditional planning still suffers from the following problems: (i) actions are assumed to be deterministic (except for different sensing outcomes) and perfect; (ii) plans do not contain loops, so it is not possible to model repetitions of parts of the plan (which is useful in HRI applications). We propose to solve these problems by adding an additional layer in the plan generation procedure that aims at improving the robustness of a plan generated by a conditional planner through execution rules. This additional layer takes the conditional plan generated by a planner and a set of declarative rules and generates a more complex and robust plan.

The execution rules allow to: (1) define execution variables (different from planning variables) associated to action executions; (2) check their values at run-time in order to execute local recovery procedures when the values of such execution variables affect the success of the execution of actions; (3) define conditions that allow repetitions of parts of the plan.

This idea has been implemented and experimented with different formalisms, including the transformation of an MDP policy to a conditional plan [Iocchi et al. ICAPS 2016] and the use of the conditional planner Contingent-FF integrated in ROSPlan (http://kclplanning.github.io/ROSPlan).

While the robust plan is represented using the Petri Net Plans (PNP) formalism (http://pnp.dis.uniroma1.it) and executed by the PNP engine.

The method has been tested with a real robot interacting with many users in public environments, within the COACHES project (https://coaches.greyc.fr/)

Although, the system is still sensible to wrong perceptions (i.e., it is still possible that wrong perceptions determine wrong executions of the plans), this risk is minimized, since perceptions are performed a minimum number of times, that is only when it is strictly required to proceed with the plan.

Discussion will include how to further improve the system, by adding more principled solutions of the above mentioned problems. Moreover, generation and execution of robust plans is an orthogonal feature that is useful for every robot planning application domain.

3.16 Rethinking Computational Investments in Planning and Execution

Gal A. Kaminka (Bar-Ilan University – Ramat Gan, IL)

Planning when to plan, and when not to. I work on multi-robot plan execution; in particular, on executing plans for teams of robots. I have been working on plan execution systems for teams of robots, for more than 15 years. A cornerstone of executing team plans is to recognize the points in which robots will make a decision, e.g., by voting or other social-choice mechanism. By recognizing these points, the systems I have designed are able to guarantee good teamwork of the robot, in the sense of carrying out their tasks in an agreed-upon manner.

There is really no feasible way to plan the voting process in advance, in the sense of planning out which robot will vote for what option. This is inherently an execution-time process. However, planning to hold a vote should be possible, just as it should ideally be possible to plan whether to hold a vote of a specific type (e.g., plurality vs. Borda vs. dictatorial). The lesson is that some executions you cannot plan, but you can plan to execute.

More generally …Back in 1997, I was taking a graduate course in artificial intelligence planning, taught by Craig Knoblock and Yolanda Gil. As a final project, we were asked to write a paper that would tackle an open question, provide the literature survey and recommend directions for further research. My paper addressed the computational effort in planning and execution. I contrasted the approaches of the planning community (graphplan and satplan were the newest planners), and the robotics community (subsumption was being pushed out in favor of behavior-based control). The two communities were seemingly at odds, scientifically. The mainstream planning community focused, as it does today, on building planners that extensively relied on simulating the effects of actions. To do this, they needed models of how the world behaves. The mainstream robotics community had just completed its embrace of reactivity, subsumption and behavior-based robotics, which emphasized throwing away models (and therefore planning), or at the very least limiting the role of planning significantly. Just a few years before, Matt Ginsburg wrote that "Universal Planning is an almost universally bad idea" in an issue of AI Magazine, and Agre and Chapman, having successfully built Pengo without a planner, were leaving both communities with the impression that there is a justified gap between robotic acting in dynamic environments, and planning for static environments. Sure, there were also researchers working on integrated planning and execution, but they were mostly working on softbots or learning policies via reinforcement learning.

I took the position that the planning community and the robotics community were in fact in complete agreement: they were both advocating the use of incredibly dumb executors. The planning community were expecting an executor which will blindly execute a series of actions, given as grounded operators. The most that would be expected from such an executor would be to check whether preconditions and effects hold as predicted. The robotics community, having given up hope on planning, was instead building very dumb mechanisms as well. Execution of policies, from this respect, is not very different: follow the policy and myopically respond as dictated. The various behavior selection and fusion mechanisms which are often discussed in this community are as myopic as the sequential selection of grounded operators for execution. In short, both communities were assuming essentially all computation is carried out in planning time. By comparison, decision-making during execution is computationally trivial, because it is myopic.

I would like to see us shifting the computational burden, to do more computation during execution (possibly, invested in projections of future state, but not only). Some HTN planners, and BDI agent architectures, come somewhat close to this, in the sense that they both allow on-line refinements for plan recipes. But their refinement is an all-or-nothing deal: recipes given in advance are either refined or are not; they are not usually modified during execution. The philosophical shift in focus of the autonomous agents community, from the agent as a planner, to the agent as a plan selector is not enough in this regard. It emphasizes the importance of integrating planning and execution, and it highlights the very real challenges involved in everything beyond generating plans. But plans are still thought of as rigid objects, generated by planners, to be handed off for execution after some filtering and selection.

I would like to have a planner that knows when to plan, and when to leave off planning to a later point in time; a planner that plans for later planning. I believe the way to achieve this goes through rethinking of the planning process, as a process that spans execution. No more interleaving of calls to a planner with calls to step execution. Rather, a single computational process that plans when it can, and automatically stops filling in details when it cannot.

3.17 Cognitive Robotics on the Factory Floor

Erez Karpas (Technion – Haifa, IL)

Industry 4.0 is one of the most common buzzwords heard today. The term is a reference to the industrial revolutions of the past: The (first) industrial revolution, in the 18th century, came about with the advent of the first steam- powered machines. The second industrial revolution, in the 19th and early 20th century, involved using electricity to power assembly lines in performing mass production. The third industrial revolution, from the 1970s, involved computer-integrated manufacturing (as well as computer-aided design), and is the origin of industrial robotics. While the term industry 4.0 has many different interpretations, it usually refers to integrating many sensors in the factory, and using analytics to derive some actionable insight from the data (this is often called Internet-of-Things and Big Data). In my talk, I will argue that cognitive robots could be of great benefit on the factory floor, and should be counted as an integral part of the fourth industrial revolution.

The first major benefit of cognitive robotics is in reducing the cost of setting up a factory. Traditional industrial robots typically require extensive, low-level, programming by highly specialized experts | a very expensive process in both cost and factory downtime. On the other hand, cognitive robots could be programmed by giving them a goal, such as “assemble 1000 electric razors of type X”, and will plan all the low-level details by themselves. Second, customized manufacturing is a very important trend now. For ex- ample, Motorola now allows customers to customize the phones they order on the web, choosing between millions of possible configurations. Cognitive robots could manufacture each device according to order, further taking into account deadlines, shipping schedules, etc.

Finally, although ideally robots will be able to do everything humans can, this will not happen any time in the near future. Thus, human-robot teamwork is an important part of any cognitive manufacturing robot. The ability to communicate with, and work alongside, a human requires high-level reasoning, and is an interesting challenge to the planning and robotics community. Although putting cognitive robots on the factory floow is a significant challenge. However, I believe it is easier to succeed on the factory floor than, for example, in a home environment for service robots, because the environment is much more structured and controllable, and because there are far fewer ethical concerns. Furthermore, a manufacturing setting provides clear and measurable economic benefit, which can allow us to claim that it is worth investing in planning and robotics research.

3.18 Multi-Robot Planning with Spatial and Temporal Constraints

Sven Koenig (USC – Los Angeles, US)

There are several gaps between symbolic AI planning research (as typically presented at ICAPS) and robotics that need to be addressed to make AI planning research more useful for robotics:

- Robots operate in dynamic worlds, which makes on-line planning necessary. They also need to make many decisions quickly during execution, not only because stopping wastes time but also because the world continues to evolve. Thus, AI planning research needs to focus more on realtime planning.
- Robots operate in spatial environments. Thus, AI planning research needs to focus more on spatial planning and the integration of spatial and temporal planning.
- Robots operate around people. Thus, AI planning research needs to take into account that the behavior of robots needs to be predictable (for example, that similar tasks should result in similar behaviors). This criterion needs to be incorporated into the objective function of AI planners. (This issue is also important in the context of replanning for

teams of humans, such as for emergency teams, where it is often important to keep the modifications of the previous plan small in case of contingencies in order to avoid a large coordination overhead. Again, similar situations should result in similar behaviors.)
- Robots cannot execute plans perfectly since planning uses models of the world and models never represent reality perfectly. Thus, plan execution will frequently deviate from the plan, which makes plan-execution monitoring and replanning necessary. AI planning research often assumes that this issue can be handled well with online replanning, which always re-solves the planning problem from the current state in case plan execution deviates from the plan. However, planning is often too slow for online replanning to be a viable strategy for real-time planning. Thus, AI planning research needs to develop integrated planning and plan-execution architectures that use slow replanning only very selectively. They could, for example, use hierarchical replanning strategies that use fast plan-adaptation whenever possible and slow replanning only as a last resort.
- Finally, multi-robot systems are more fault-tolerant and allow for more parallelism than single-robot systems. Thus, AI planning research needs to focus more on cooperative multi-agent planning, both in centralized and – very importantly – decentralized settings. Some AI planning research studies multi-agent planning but focuses on privacy, which is less important for robotics than other applications.

The research of my research group addresses these issues in the context of multi-robot path finding, where multi-robot teams have to assign target locations among themselves and then plan collision-free paths to them. Examples include automated warehouse systems, autonomous aircraft towing vehicles, office robots and game characters in video games. For example, hundreds of robots already navigate autonomously in Amazon fulfillment centers to move inventory pods all the way from their storage locations to the packing stations. Path planning for these robots is NP-hard, yet planning must find high-quality collision-free paths for them in real-time.

There is a long way to go to bridge the gap between AI planning research and robotics. We advocate robotics-friendly planning domains for IPC competitions as one possible way to engage AI planning researchers. For example, one suggestion is to use a planning domain that models the Harvard TERMES robots as part of the multi-agent planning competition. The Harvard TERMES project investigated how multiple robots can cooperate to build userspecified three-dimensional structures much larger than themselves. Planning is required, even for single robots, to build structures effectively since they need to build ramps to reach high places but ramps consist of many blocks and are time-consuming to build. Thus, robots need to plan carefully when and where to build ramps and, once built, how to utilize them best. Planning for single robots is already difficult due to the large number of blocks and long plans. Planning for multiple robots is even more difficult since, as for multi-robot path finding, it needs to reason about how to achieve a high degree of parallelism without robots obstructing each other even though many robots operate together in tight spaces.

3.19 Explainable Robotics

Lars Kunze (University of Birmingham, GB)

Planning and decision making in the real world requires autonomous robots to draw on different sources of knowledge. This includes prior knowledge about the domain, knowledge acquired through longterm experience, and knowledge derived from more recent observations through robotic sensors. Hence, planning and decision making as well as the resulting behavior of robot systems, can be quite complex. For end-users, and sometimes even for robot developers, it might not be clear why a system behaves in a particular way. Therefore, I argue that autonomous robots should be equipped with principled ways that allow them to explain their own behavior and their own decisions. For example, to answer the question "why did you stop in front of a green traffic light?" an autonomous car could generate an explanation such as "I stopped because I saw a person approaching at very high speed". By providing explanations and/or justifications for decisions users can follow and comprehend the internal processes of robot systems. Such transparency can lead to an increased user acceptance and eventually to trust in autonomous robots in general. Moreover, while analyzing the behavior of a robot system developers could benefit from mechanisms that can explain 'why' an action was performed.

Realizing systems that can provide explanations about themselves and their own behavior poses several challenges. First of all, these systems require explicit (and interpretable) representations about the world, about themselves, and their planning and decision making processes. Secondly, robots need to be equipped with inference mechanisms to reason about different possible explanations. Finally, to provide explanations to users and developers novel interfaces for Human-Robot Interaction (HRI) are required. In previous work, we have developed the Semantic Robot Description Language (SRDL). SRDL is based on the Web Ontology Language (OWL) and provides a principled way to describe robots, their components, and their capabilities semantically. It allows robots to explain what tasks (and actions) they are able to perform and it allows them to infer why they are not able to perform certain actions. Hence, I believe that SRDL is a good starting point for enabling robots to reason about their own planning and decision making processes. For reasoning about different possible explanations we are currently investigating Answer Set Programming (ASP). ASP has been successfully integrated with ontologies and uses explicit representations to reason about different possible worlds (or Answer Sets). Hence, I think that ASP is a reasonable candidate for generating sets of different explanations for planning decisions.

Finally, I believe that the design and the development of novel HRI interfaces that make interpretable models accessible to users and developers is an open problem which should be addressed by researchers from various disciplines including (Cognitive) Psychology, Computer Science (HCI), and Robotics.

To summarize, I believe that autonomous robots (and other AI systems) should be equipped with the capability to 'explain' their own behavior and their own decisions. I suspect that such explanations will lead to a better understanding of robot systems in general. Thereby, user acceptance will be increased and trust in robot systems will be build up. Additionally, developers will benefit from transparent planning and decision processes when analyzing the behavior and the performance of complex robot systems.

3.20 Probabilistic Planning for Mobile Robots with Formal Guarantees

Bruno Lacerda (University of Birmingham, GB)

In recent years, the field of mobile service robots has witnessed important developments in terms of the ability for robust deployments in real life environments. Thus, we are quickly approaching an era where robot systems are regularly deployed among humans as an extra tool to improve productivity, for example, in office environments. Robots in such scenarios can perform a range of tasks such as fetch-and-carry, or security checks. In order to perform such deployments, one needs the ability to quickly generate robust and efficient plans for large scale systems. On top of that, another aspect I see as a crucial element in any safe, robust and efficient design/deployment loop is the ability to provide formal guarantees of performance for such plans. For example, guarantee that a robot will never navigate into dangerous regions of the environment; provide a value for the probability of a task being successfully completed; or give an expectation of the time that the execution of a task will take.

I have been researching the use of probabilistic model checking techniques for the generation of high-level plans for mobile robots with probabilistic formal guarantees, and using such plans on real life mobile service robot deployments. In parallel with the research on probabilistic model checking, and using many similar techniques, there have been efforts from the artificial intelligence and planning community on sequential decision making under uncertainty. This research generally does not provide formal guarantees. On the other hand, it focuses on fast generation of plans for large problems, using approximation techniques, something which probabilistic model checking approaches struggle with. Broadly speaking, even though probabilistic model checking and sequential decision making use many of the same underlying models, historically, the point of view of each field has been slightly different. A key point of my research agenda is to bring ideas from these two fields together, applying them to the deployment of safe and robust robot deployments in the real world.

Finally, I believe that extending probabilistic model checking techniques to multi-robot systems can yield very significant contributions to the field. In particular, there are many works applying sequential decision making techniques to multi-agent coordination. Building on those, ad also extending single-robot verification techniques in order to provide team level guarantees at different levels of abstraction is currently my main research goal. These techniques will combine approaches from sequential decision making and probabilistic verification in order to generate policies for task allocation and multi-robot coordination, with attached probabilistic guarantees, such as “regardless of the state of the team, there will always be one robot able to get to reception within 5 min, with probability 0.95”. The main challenges in order to achieve this goal are to correctly apply different techniques in a well founded way, such that the overall framework can scale while still being able to provide meaningful guarantees over both individual robots, and overall team performance.

3.21 Planning for Persistent Autonomy: Where are we struggling?

Daniele Magazzeni (King's College London, GB)

AI Planning is about determining actions before doing them, anticipating the things that will need to be done and preparing for them. Planners use domain-independent heuristics to guide the search in huge state spaces.

Recently, AI Planning has been successfully applied to handle complex systems. PDDL+ is the formalism used to describe hybrid systems, and allows the modelling of the differential equations governing the continuous behaviour of the system. This talk provides an overview of how PDDL+ can be used to model robotics and autonomous systems; presents a new PDDL+ planner based on SMT and the ROSPlan framework for planning with ROS; highlights some open challenges on the integration between planning and robotics.

3.22 Temporal Planning for Execution

Lenka Mudrova (University of Birmingham, GB)

Many researchers dream about a mobile service autonomous robot who will roam our work environments and homes and assist us with our every day tasks. If we pretend that robotics has solved all "low-level" problems to make such a robot possible (hence the robot has certain capabilities, such as moving around, grasping objects, opening doors, observing and recognising, understanding speech, etc.) then we face at least the following questions.

1. *How* the robot can *plan* its behaviour in order to be able to perform required tasks given by a human?
2. *When* the robot should act in order to satisfy human's time constraints?
3. How to *execute* obtained plan in more robust sense, i.e., the robot is not stopping and re-thinking everything all the time...
4. How to *react* to situations that can go wrong in the execution? How to react when experiencing *unknown unknowns*?

In my currect research, Ive focused on giving some answers to Questions 1 and 2, developing an approach based on merging of partial order plans with durative actions, that can quickly and effectively generate a plan for a set of independent tasks. This plan exploits some of the synergies and demands of the plans for each single task, such as common locations where certain actions should be executed. This approach also handles situations when a task is required to be satisfied within a time window, and the partial order of the plan is a strong benefit for execution, when the final plan can be joined online in reaction to the current observations.

In order to make a progress with Questions 3 and 4, I think the community needs to move away from the current evaluation type "it runs" to benchmarking. As benchmarking in the real world is hard due to influence of many uncontrollable events, I propose to focus on developing benchmarking domains in simulations (using standard robot simulators) where different aspect affecting execution can be plugged in and repeat it many times under same conditions. Hence, more discusstion is needed about what are the aspects to be monitor during execution and modelled in such a benchmark domain.

3.23 Symbiotic Human Robot Planning

Daniele Nardi (Sapienza University of Rome, IT)

While operating in domestic environments, robots will necessarily face difficulties not envisioned by their developers. Moreover, the tasks to be performed by a robot will often have to be specialized and/or adapted to the needs of specific users and specific environments. Hence, the conventional approach to planning, based on a fixed action specification does not seem a suitable modeling tool.

Learning how to operate by interacting with the user seems a key enabling feature to support the introduction of robots in everyday environments. Symbiotic autonomy is a recently introduced viewpoint where the user should help the robot to improve its performance and to perform tasks otherwise not achievable. This novel perspective to the design of intelligent robots leads to a number of interesting research questions that are related to planning. First, the robot should plan including speech acts and, more specifically, requests for help from the user. Second, the robot can learn from the user plans to accomplish complex task. Third, the robot can learn from action/plan failures by requesting explanations to the user.

Our aim is to explore the above research question and illustrate some initial contributions following this approach. In particular, we present a novel approach for learning, through the interaction with the user, complex task descriptions that are defined as a combination of primitive actions. The proposed approach makes a significant step forward by allowing task descriptions parametric with respect to domain specific semantic categories. Moreover, by mapping the task representation into a task representation language, we are able to express complex execution paradigms and to revise the learned tasks in a high-level fashion. The approach is implemented in multiple practical test cases with a service robot.

3.24 Towards an Integrated Approach to Planning and Execution

Tim Niemüller (RWTH Aachen, DE) and Gerhard Lakemeyer (RWTH Aachen, DE)

Challenges

Building a robotics system is inherently an integration challenge. A diverse set of software components must be combined and the interaction with the physical world places high demands on robustness and fault tolerance. Still, task-level planning and reasoning for autonomous mobile robots – that claims to help in solving in some of these challenges by automatic and flexible behavior design – is still the exception rather than the norm. A part of the problem is that the scope of the planning community often ends once the plan has been generated. In the robotics community, on the other hand, most researchers are concerned with other components such as perception, navigation, or manipulation. The modeling and integration overhead for planning systems often appears considerable even for small problems, and unable to scale to larger ones.

Therefore, from our perspective one of the fundamental challenges for the closer cooperation and mutual benefit of both such communities is an integrated approach to planning and execution. That starts with good craftsmanship to design and implement the appropriate software interfaces. But it also contains research questions, such as what would a unified language for planning and execution look like, and what would be its model and semantics? What would be an accurate and expressive representation of plans that allows to combine, for example, classical and temporal plans for a common executive and to choose the appropriate planner depending on the sub-problem to solve? What does execution monitoring for generated plans mean in the presence of uncertainty and contingencies?

Some Pieces of the Puzzle

To tackle these questions, efforts are required in both communities. While the planning community's first and foremost goal is to efficiently generate plans, for the robotics community task-level behavior is often only means to test and demonstrate other components. Bringing these worlds together requires dedicated work on the interface part, the integration of efficient planning systems and execution and execution monitoring of generated plans.

An observation is that execution of plans (handled by an executive of some sort) is often only an afterthought, if at all. It often requires interfacing with planners that produce output in a mostly non-unified format (contrary to the somewhat unified input language based on PDDL), making it harder to replace a specific planning system and also leading to many ad-hoc solutions if planning is used on a robot. We think that a unified language that combines the definition of a planning domain and problem, and also of the execution and monitoring of the resulting plans is desirable. We have made first steps with GologCP [5] which builds on ideas for automatic Golog/PDDL translation [3] and continual planning [2] to create an integrated framework to planning and execution. The results show a significant performance improvement and sound modeling. However, a language more similar to PDDL or other planning languages might be desirable, for example, building on PRS ops [1].

A generalized representation of plans might be desirable. There have been systems already using simple temporal networks (STN) on the planning [4] and the execution [6] side. It can represent simple or more complex plans and thus scale with the capabilities of the planning system. We are currently investigating the possibility to use a classical planner to solve sub-problems in a multi-robot context, and then use additional information to transform the resulting sequence into an STN to allow for parallel execution of parts of the plan.

It seems useful if the execution of plans itself was more carefully modeled including the interactions between planning and execution. This might also include extensions for plan repair if re-planning is too costly, starting execution already once a (likely) prefix of the plan has been determined, or to include assertions as in continual planning, which are conditionally expanded sub-plans that allow to cope, for example, with incomplete knowledge (postponing it to be a run-time decision, requiring a closer integration).

While we do not suggest giving up the separation between planning and execution generally, we think it is still worth investigating the possible interactions of the two processes, and benefits or drawbacks on making these more explicit, fine-grained, and more expressive.

Evaluation Scenario

The integration of planning and execution demands different evaluation scenarios as are used, for example, in the International Planning Competition. Autonomous mobile robotics scenarios naturally require such a closer integration as the environments are typically dynamic, short reaction times are necessary, and there are many uncertainties. In cooperation with Karpas, Vaquero, and Timmons we therefore proposed a Planning Competition for Logistics

Robots In Simulation [7]. It provides a suitable scenario at a comprehensible size. It is to be understood as a first step and we explicitly deem other scenarios relevant and useful. The chose scenario is based on the RoboCup Logistics League [8], an established real-world robotics league that focuses on production in modern manufacturing environments downscaled and build with readily available hardware. It has a natural focus on planning and reasoning systems for multirobot coordination and cooperation. Performing the simulation in simulation (at least initially) allows to easily adjust complexity, size, and duration.

References

1 Alami, R.; Chatila, R.; Fleury, S.; Ghallab, M.; and Ingrand, F. 1998. An architecture for autonomy. The International Journal of Robotics Research 17(4).

2 Brenner, M., and Nebel, B. 2009. Continual planning and acting in dynamic multiagent environments. Autonomous Agents and Multi-Agent Systems 19(3).

3 Claßen, J.; Röger, G.; Lakemeyer, G.; and Nebel, B. 2012. PLATAS – integrating planning and the action language Golog. KI – Künstliche Intelligenz 26(1).

4 Halsey, K.; Long, D.; and Fox, M. 2004. CRIKEY – A Temporal Planner Looking at the Integration of Scheduling and Planning. In Workshop on Integrating Planning into Scheduling at 13th International Conference on Automated Planning and Scheduling (ICAPS).

5 Hofmann, T.; Niemueller, T.; Claßen, J.; and Lakemeyer, G. 2016. Continual Planning in Golog. In Proceedings of the 30th Conference on Artificial Intelligence (AAAI).

6 Levine, S. J., and Williams, B. C. 2014. Concurrent plan recognition and execution for human-robot teams. In Int. Conf. on Automated Planning and Scheduling (ICAPS).

7 Niemueller, T.; Karpas, E.; Vaquero, T.; and Timmons, E. 2016. Planning Competition for Logistics Robots in Simulation. In WS on Planning and Robotics (PlanRob) at Int. Conf. on Automated Planning and Scheduling (ICAPS).

8 Niemueller, T.; Lakemeyer, G.; and Ferrein, A. 2015. The RoboCup Logistics League as a Benchmark for Planning in Robotics. In WS on Planning and Robotics (PlanRob) at Int. Conf. on Aut. Planning and Scheduling (ICAPS).

3.25 How much reliable are plan-based controllers for autonomous robots?

Andrea Orlandini (CNR – Rome, IT)

Decisional autonomy is considered among one of the key system abilities for robotics applications. This entails robotics platforms to be endowed with a wide set of automated reasoning capabilities to be implemented by means of suitable technologies. Among these, automated planning and scheduling (P&S) technology plays a crucial role.

In general, automated P&S systems are finding increased application in real-world mission critical systems that operate under high levels of unpredictability. Given a description of a desired goal, and a model of possible actions and their causal/temporal constraints, the planning problem consists of finding a plan, which is a sequence of actions, the execution of which is calculated to lead to the goal state under normal circumstances. Such technology can be used to generate plans to control a plant (for example a robot), driven by goals often issued by humans. Such technology is occasionally referred to as model--based autonomy. Then, a P&S system takes as input a domain model and a goal, and produces a plan of actions to be executed, which will achieve the goal. A P&S system typically also offers plan execution and monitoring engines.

To foster effective use of Automated P&S systems in (near future) robotics applications such as, for instance, service robots, it is of great importance to significantly increase the trust of end users in such technology.

On one hand, automated P&S systems often bring solutions which are neither "obvious" nor immediately acceptable for them. This is mainly because these tools directly reason on causal, temporal and resource constraints; moreover, they employ resolution processes designed to optimize the solution with respect to non trivial evaluation functions. On the other hand, due to the non--deterministic nature of planning problems, it is a challenge to construct correct and reliable P&S systems, including, for example, declarative domain models. That is, it is not straightforward to guarantee the correctness/reliability of P&S systems.

In this regard, Verification and validation (V&V) techniques may represent a complementary technology with respect to P&S, that contribute to develop richer software environments to synthesize a new generation of robust problem--solving applications. The aim of this talk is to discuss open issues related to V&V techniques in P&S considering multiple needs, i.e., considering V&V of domain models, V&V of plans, V&V of plan executions, V&V of planners, V&V of plan execution engines and V&V of plan execution monitors.

References

1 Bensalem S, Havelund K, Orlandini A (2014) Verification and validation meet planning and scheduling. International Journal on Software Tools for Technology Transfer 16(1):1–12.

2 A. Cesta, A. Finzi, S. Fratini, A. Orlandini, and E. Tronci, (2010) Validation and Verification Issues in a Timeline-Based Planning System, Knowledge Engineering Review, vol. 25, no. 3, pp. 299–318.

3 Orlandini A, Finzi A, Cesta A, Fratini S (2011) TGA-based controllers for flexible plan execution. In: KI 2011: Advances in Artificial Intelligence, Proceedings of the 34th Annual German Conference on AI, Springer, LNCS, vol 7006, pp. 233–245.

4 Cesta A, Finzi A, Fratini S, Orlandini A, Tronci E (2010) Analyzing flexible timeline-based plans. In: Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010), pp. 471–476.

3.26 Some practical issues for social consumer robots: an industrial perspective

Amit Kumar Pandey (Aldebaran Robotics – Paris, FR)

Personal social robot for everyone is the next big thing in the history of robotics. It is the time when robots are entering into our day to day life. And we, the human, together with such social robots, are converging towards creating an intelligent and embodied eco-system of living, where robots will coexist with us in harmony, for a smarter, healthier, safer and happier life. There are some key ingredients for such robots to be a successful personal and social robots. Such robots are expected to behave in socially accepted and socially expected manners. For this to achieve, Social Intelligence of robots will be the paramount.

For achieving Social Intelligence in robots, there is a great need towards developing coherent theoretical and functional framework, by identifying the basic ingredients of development of social skills. Some of those are social interaction, situation assessment, social learning, socially-aware manipulation & navigation. For each of these aspects, the robot has to plan and act accordingly to fulfil the needs, while taking into account that it is operating in a human-centered environment with potentially human around it. This creates a new era of planning problem, which goes beyond the mere safety aspect of planning towards socially-aware aspects. For example, the robot has to plan the interaction action, for it to appear social, the robot has to chose where to focus and what to "perceive" to be useful for the situation and interaction, the robot has to react to stimuli, the robot has to understand the meaning of the day to day tasks, so that it can plan to perform them differently in different situations, without the need of pre-programming for each and every situation it can encounter in daily life, the robot has to be able to incorporate high-level human-oriented and Social constraints in its manipulation and navigation plannings, and should be able to come up with shared plans if necessary, (and by involving human) to achieve a task in socially intelligent manner. Further, all these have to be achieved with the additional constraint of being realtime, intuitive and for real environment.

The talk will illustrate such issues, through some of the use cases for social robot grounded with some European Projects. It will try to provide a generalized definition of action from Human-Robot Interaction and Socially Intelligent Robot perspectives. This will be followed by feedback from real users and discuss some of the immediate multi-disciplinary R&D challenges and needs from industrial perspective, and some initial results towards solving them, including planning for interaction, perception, manipulation & navigation, and highlighted human-in-the-loop based learning aspects of robots for understanding task semantics, complex affordances, and being proactive. The talk will conclude with some open and grand challenges ahead, for us, the interdisciplinary community, to brainstorm and solve.

3.27 Multirobot coordination

Simon Parsons (King's College London, GB)

This short paper describes progress on the challenging problem of planning and executing missions with teams of robots. The overall goal of this work is as follows:

> Given a mission specification and access to a group of robots with a range of abilities, first select a team of robots, some subset of the full group, which can achieve the mission. Then construct a plan for achieving the mission, distributing tasks to individual team members. Finally, execute the mission, monitoring the progress of the plan, and adjusting it if and when that is necessary[1].

While we have made progress on several of the elements, we are still a considerable way from the goal.

---

[1] I think of this as the Mission Impossible problem since it was a major feature of the plot of pretty much every episode of the early seasons of the 1960s TV show. Later series dropped the "Dossier Scene" in which the team leader picked the members of his team according to their skills.

The area on which we have made the most progress in recent years is that of task allocation. Provided that the mission is specified as a set of sub-tasks, task allocation is enough to generate a plan. (A necessary extension to this work is being able to handle the decomposition of missions into tasks in the case in which a mission is not specified so conveniently.) In [2, 4, 3], we have looked at task allocation mechanisms in a range of scenarios. These range from the simple case in which all tasks are known at the start, all tasks require one robot, and all tasks can be carried out by every robot [2], through the case in which tasks are given to the team over time [4] to the case in which tasks can require more than one robot, and there are constraints between tasks [3]. The approaches that we used were all market-based, in which team members bid for tasks based on the cost to them of executing the tasks (in the scenarios we have looked at, cost is related to the distance of the robot from the task, fuel cost if you will). More work is necessary here, for example to allow robots to switch tasks when that is appropriate, and to handle truely heterogeneous capabilities.

The other area in which we have made progress, overlaps somewhat with what was just described. In this work, [6, 7] we looked at constructing a plan for a team in a distributed fashion, here using standard planning representations. The advantage of constructing a plan like this, is that team members can make use of local information (see [5] for an example of how exploiting local information can be advantageous). Once a plan is constructed, team members can then monitor their progress against it, and can decide whether it is necessary to contact their teammates to ensure that joint activities are correctly coordinated[2].

The major area in which there is still work to do (in addition to the areas already mentioned) is the one that corresponds to the first element of the description above – from a mission specification, and a set of potential members, identify the team members with the necessary skills to complete the task. This could be viewed as a planning task: pick a subset of the group, see if there is a feasible plan to complete the mission, if not then try another subset. However, a more attractive approach is one which views this as both a knowledge representation problem (how to represent a mission, and the capabilities of a team member) and a problem of reasoning at different levels of granularity, since it should not be necessary to fully plan out the completion of a mission to select a team that is capable of achieving it.

References

1 S. Parsons, S. Poltrock, H. Bowyer, and Y. Tang. Analysis of a recorded team coordination dialogue. In *Proceedings of the Second Annual Conference of the International Technology Alliance*, London, 2008.

2 Eric Schneider, Ofear Balas, A Tuna Özgelen, Elizabeth I Sklar, and Simon Parsons. Evaluating auction-based task allocation in multi-robot teams. In *Workshop on Autonomous Robots and Multirobot Systems (ARMS) at Autonomous Agents and MultiAgent Systems (AAMAS)*, Paris, France, May 2014.

3 Eric Schneider, Elizabeth I. Sklar, and Simon Parsons. Evaluating multi-robot teamwork in parameterised environments. In Lyuba Alboul, Dana Damian, and Jonathan M. Aitken, editors, *Towards Autonomous Robotic Systems: 17th Annual Conference, TAROS 2016*, pages 301–313. Springer International Publishing, 2016.

4 Eric Schneider, Elizabeth I Sklar, Simon Parsons, and A Tuna Özgelen. Auction-based task allocation for multi-robot teams in dynamic environments. In *Towards Autonomous Robotic*

---

[2] In human teams, a large amount of the communication seems to be status updates [1]. By updating only when necessary, we can save communication bandwidth.

Systems: 16th Annual Conference, TAROS 2015, pages 246–257. Springer International Publishing, 2015.
5 E. I. Sklar and M. Q. Azhar. Argumentation-based dialogue games for shared control in human-robot systems. *Journal of Human-Robot Interaction*, 4(3):120–148, 2015.
6 Y. Tang. *A Symbolic Exploration of the Joint State Space and the Underlying Argumentation-based Reasoning Processes for Multiagent Planning.* PhD thesis, Graduate Center, City University of New York, 2012.
7 Y. Tang, T. Norman, and S. Parsons. A model for integrating dialogue and the execution of joint plans. In *Proceedings of the 8th International Conference on Autonomous Agents and Multi-Agent Systems*, Budapest, Hungary, 2009.

3.28 Knowledge-level planning for Human-Robot Interaction

Ron Petrick (Heriot-Watt University – Edinburgh, GB)

At a basic level, the automated planning problem is one of context-dependent action selection: given an initial state, a domain description, and a set of goals, generate a sequence of actions whose execution will bring about the goal conditions. However, the problem of action selection is not unique to automated planning. One important field where this issue is also of primary concern is that of spoken dialogue systems, whose tools play a central role in addressing the problem of human-robot interaction. At the heart of the dialogue system is the interaction manager whose primary task is to carry out a form of action selection: based on the current state of an interaction, the interaction manager makes a high-level decision as to which spoken, non-verbal, and task-based actions the system should apply. An important aspect of research in this area has been the development of toolkits to support the construction of end-toend systems. Given the parallels between the planning and dialogue tasks, our recent work has explored the application of automated planning techniques to human-robot interaction (HRI) as an alternative to standard dialogue system toolkits (such as Trindikit, COLLAGEN, IrisTK, OpenDial, among others).

While the link between natural language processing and automated planning has a long tradition, going back to at least the 1980s, in recent years the two communities have focused on different problems and solutions, with planning for natural language problems largely overlooked in favour of more specialpurpose solutions. For instance, the interactive systems toolkits attempt to offer a one-stop solution for system building combining action selection, representation, and technical architectures. In contrast, the planning community has focused on defining domains in common representation languages like PDDL and comparing different domain-independent strategies within this context through events like the International Planning Competitions; the study of the representation languages themselves has also led to a better understanding of the trade-offs between different representations.

Our own work in this area has focused on applying domain-independent knowledge-level planning techniques to the problem of action selection in human-robot interaction. In particular, the beliefs of the planning agent (robot) about the world and other agents are represented, and sensing actions are used to model certain types of information-gathering speech acts. Task-based actions are also planned using the same general-purpose planning mechanisms.

However, the problem of human-robot interaction also offers some wider opportunities and lessons for the planning community. First, the presence of action selection at the core of interaction management offers the obvious possibility of applying other types of planning techniques. Second, the nature of the applications addressed by many HRI systems also highlights the importance of building real-world systems – an area that has gained wider traction in the planning community but one that is still somewhat outside the mainstream of most planning research. Finally, the process for evaluating robot-based dialogue systems, and in particular the role of human users, also presents new directions and challenges for planning.

3.29 Goal Reasoning for Robotics

Mark Roberts (Naval Research – Washington, US)

Goals are the hinge-pins of deliberative behavior. Whether explicit (e.g., as a symbolic structure), implicit (e.g., as a reward or error signal), provided by a designer, or learned over time, goals unify motivations and action. They can be used to prune the search for solutions, to label or query persistent storage, or to structure learning from experience. Researchers have recently proposed a synthesis of the research disciplines that examine deliberation about goals, calling it goal reasoning: the ability of an agent to determine, pursue, and modify its own goals in response to notable events. In this talk, we identify three challenges we have studied with respect to goal reasoning systems in robotics. We will present the goal lifecycle, a formal model of goal reasoning built on goal-task networks, and showcase its implementation in a system called ActorSim, which links several robotic and virtual autonomous systems.

To perform complex tasks, a team of robots requires both reactive and deliberative planning. For reactive control, a restricted variant of Linear Temporal Logic called General Reactivity(1) can be used to synthesize correct-by-construction controllers in polynomial time, but they often ignore time and resource constraints to maintain tractable synthesis. For deliberation, hierarchical goal reasoning can be used to reason about time and resources. However, the coordination of reactive control and deliberation remains a challenge, which we accomplish through a set of Coordination Variables. We integrate these two approaches in the Situated Decision Process (SDP), a predecessor of part of ActorSim. The SDP will allow an Operator to control a team of semi-autonomous vehicles performing information gathering tasks for Foreign Disaster Relief operations. We demonstrate that the SDP responds to a dynamic, open world while ensuring that vehicles eventually perform their commanded actions.

In complex and dynamic scenarios, autonomous vehicles often need to intelligently adapt their behavior to unexpected events. We extend the ActorSim to include information measures and expectations used by the vehicles to assess their performance. This system, called Goal Reasoning with Information Measures, is demonstrated using a disaster relief scenario in which a small team of vehicles is tasked with surveying a predefined set of geographical regions. Additionally, a preliminary study shows that the inclusion of resolution strategies increases the likelihood that it successfully finishes its goals.

Finally, robots are increasingly performing well on focused tasks in constrained worlds over increasing time horizons. We argue that goal reasoning is essential as autonomous systems, robotic or virtual, transition to operating in open worlds over time horizons of months or years while maintaining hundreds of goals that vary in duration and priority. But to achieve long-duration autonomy presents two new challenges. First, existing robots can have relatively short life spans, limiting progress. As a contingency, we plan to study long-duration autonomy in 3D game engines similar to the way in which the Robocup simulator served as an early testbed until robotics systems became more capable and reliable. Second, we must enable a robot to store, access, and learn over very long time horizons of weeks, months, or even years. This presents challenges not only in how to capture this information but also in how to maintain an ever growing knowledge base, retrieve relevant memories and experience, and update it with new knowledge. We argue that cognitive structures are needed to manage long-term memory structures, focus effort, and derive curricula. Further, we argue that hybrid model-based and reactive control architectures must be leveraged because each excels at complementary tasks in a robot.

3.30 Effective Hybrid Planners for robotics

Enrico Scala (Australian National University – Canberra, AU)

A fundamental building block towards an effective integration and exploitation of AI planning in robotics systems requires reasoning mechanisms over mixed representations combining logical and numeric constraints. This becomes apparent even in very simple problems of integration between task and motion planning, where geometric and causal reasoning have to be considered in an intertwined way [15, 5]. Decoupling them may result in very poor performances.

Planning for such hybrid systems is a very hard computational problem (even very restricted models are NP-hard to solve) since it requires an intertwined reasoning over discrete and continuous domain variables along with changes of the states that can also be both discrete and continuous. Work to better handle such problems has been done [4, 3, 14, 8, 7], but more work is still needed to scale up to realistic size problem.

In my research I am investigating different methods to deal with a discretised version of this class of problems: forward state space planning via heuristic search, compilation to satisfiability modulo theory, robust plan execution and plan repair ([9, 10, 11, 13, 12]). These works adapt and extend well known classical planning techniques to the hybrid case in different ways, all of them starting from the key observation that a powerful computational representation of the hybrid case (including processes) is that of sequential numeric planning with global constraints. Improving on (and exploiting) the reasoning about the exposed numeric structures of the problem becomes of crucial importance. By adapting and extending previous work done in classical planning, these works do actually attempt to solve classical (task) and numeric (motion) planning in an integrated way.

These approaches though suffer in some situations, and in particular when many and complex constraints need to be enforced. I see this seminar as a great opportunity to engage discussions with people working on constrained-based/timeline-based planning [1, 2] and/or motion planning, to study synergies and exchange of methods among these approaches.

References

1 Javier Barreiro, Matthew Boyce, Minh Do, Jeremy Frank, Michael Iatauro, Tatiana Kichkaylo, Paul Morris, James Ong, Emilio Remolina, Tristan Smith, et al. Europa: a platform for ai planning, scheduling, constraint programming, and optimization. *4th International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS)*, 2012.

2 John L. Bresina, Ari K. JÃ³nsson, Paul H. Morris, and Kanna Rajan. Mixed-initiative planning in mapgen: Capabilities and shortcomings. In *In Proceedings of the ICAPS-05 Workshop on Mixed-initiative Planning and Scheduling*, 2005.

3 Michael Cashmore, Maria Fox, Derek Long, and Daniele Magazzeni. A compilation of the full PDDL+ language into SMT. In *Planning for Hybrid Systems, Papers from the 2016 AAAI Workshop, Phoenix, Arizona, USA, February 13, 2016.*, 2016.

4 Giuseppe Della Penna, Daniele Magazzeni, Fabio Mercorio, and Benedetto Intrigila. Upmurphi: A tool for universal planning on PDDL+ problems. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling, ICAPS 2009, Thessaloniki, Greece, September 19–23, 2009*, 2009.

5 Christian Dornhege, Patrick Eyerich, Thomas Keller, Sebastian Trúg, Michael Brenner, and Bernhard Nebel. Semantic attachments for domain-independent planning systems. In *Proc. of International Conference on Automated Planning and Scheduling (ICAPS-09)*, 2009.

6 Maria Fox and Derek Long. Modelling mixed discrete-continuous domains for planning. *J. Artif. Intell. Res. (JAIR)*, 27:235–297, 2006.

7 Drew V. McDermott. Reasoning about autonomous processes in an estimated-regression planner. In *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS 2003), June 9-13, 2003, Trento, Italy*, pages 143–152, 2003.

8 Wiktor Piotrowski, Maria Fox, Derek Long, Daniele Magazzeni, and Fabio Mercorio. Heuristic planning for pddl+ domains. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), New York (NY), USA*, 2016.

9 Enrico Scala, Patrik Haslum, and Sylvie Thiébaux. Heuristics for numeric planning via subgoaling. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3228–3234, 2016.

10 Enrico Scala, Patrik Haslum, Sylvie Thiébaux, and Miquel Ramírez. Interval-based relaxation for general numeric planning. In *ECAI 2016 – 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, pages 655–663, 2016.

11 Enrico Scala, Miquel Ramírez, Patrik Haslum, and Sylvie Thiébaux. Numeric planning with disjunctive global constraints via SMT. In *Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, ICAPS 2016, London, UK, June 12-17, 2016.*, pages 276–284, 2016.

12 Enrico Scala and Pietro Torasso. Proactive and reactive reconfiguration for the robust execution of multi modality plans. In *ECAI 2014 – 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic*, pages 783–788, 2014.

13 Enrico Scala and Pietro Torasso. Deordering and numeric macro actions for plan repair. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1673–1681, 2015.

14 Ji-Ae Shin and Ernest Davis. Processes and continuous change in a sat-based planner. *Artificial Intelligence*, 166(1-2):194–253, 2005.

15 Siddharth Srivastava, Eugene Fang, Lorenzo Riano, Rohan Chitnis, Stuart Russell, and Pieter Abbeel. Combined task and motion planning through an extensible planner-independent interface layer. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 639–646. IEEE, 2014.

3.31 Planning for Open-ended Missions

Matthias Scheutz (Tufts University – Medford, US)

Many envisioned applications for future robots (e.g., robots for search and rescue domains, space environments, etc.) will take place in contexts that are "open", i.e., where aspects of the task and the mission are not known ahead of time and where unforeseen events can alter mission planning and execution. Most current robotic architectures, however, are only able to deal in a very limited way with open-world aspects. In particular, most planning algorithms assume that the domain model for the planner is given in its entirety, so that planning really amounts to search over the states and actions defined by the model. We argue that such an assumption is not warranted in open worlds and that task and action planners integrated into robotic architectures need to be able to intrinsically cope with unknown aspects of the domain model (e.g., goals that involved entities where the planner does not know all relevant aspects of the entity).

3.32 Planning with Incomplete models

Reid Simmons (NSF – Arlington, US)

Planners need models in order to predict the effects of actions. We all know, however, that any model of the real world is merely an incomplete representation. While some models may be higher fidelity than others, all are deficient in some respects, or another. Thus, planning can never accurately predict all the effects of actions in every context.

In robotics, feedback controllers are often used to address this deficiency – plans provide higherlevel of abstractions and the low-level controller behaviors are designed to achieve the intended effects of the plan. Clearly, however, there are drawbacks with this approach; for instance, suboptimal plans may be produced that cause the controllers to expend much more time/energy than would otherwise be necessary, and, in the worst case, the plans may fail altogether because the controllers are not applicable for the contexts in which the plans have put them.

The question, then, is how to deal with the inevitable fact of planning with incomplete models? I suggest three approaches that can help – planning with probabilistic models, learning to improve models, and planning with multiple models.

Planning with probabilistic models, such as MDPs or POMDPs, is now considered fairly standard in robotics. The idea is to "mask" the incompleteness of the models using a distribution of possible effects of actions, and to plan to achieve a metric such as maximizing (or exceeding some threshold of) probability of plan achievement, maximizing expected reward, or maximizing expected utility, taking risk into account. Such approaches produce plans that work well, on average, provided that the reward function, and transition and observation probabilities, are close to their true values.

This leads to learning to improve models. I start with the assumption that the models are reasonably accurate, but do not completely reflect reality. Here, experience-based learning can be used to improve transition and observation probabilities, and techniques such as learning from demonstration and inverse reinforcement learning can be used to improve the reward function. In addition, by modeling the uncertainty in the model parameters, active learning can be used to guide the agent to situations where it can efficiently learn the parameters it is most uncertain about.

Things get more complicated, though, in situations where models have hidden (latent) state. Effective methods for learning such models exist, such as EM or spectral methods, but they typically need large amounts of training data. We are exploring an alternate, data-efficient, approach that uses statistical tests to identify regions of the state space that appear to be drawn from different distributions than other parts of the state space. The approach incrementally searches ellipsoidal regions of the state space to find the contexts in which the observed distribution differs significantly from the model. For instance, it might discover that the robot typically slips more than expected when turning left at a high velocity in a given area of the building. While we, as people, may understand that is because the area is tiled, to the robot it is sufficient to have improved its navigational model in the given context, which enables it to create better plans (e.g., by slowing down in that area, or avoiding it altogether).

Finally, we acknowledge that different models often have different strengths and weaknesses. By choosing the most appropriate model for a particular planning task, one may produce better plans for that particular situation. To that end, we are exploring an approach that uses of a hierarchy of models. The idea is to plan first in a lower fidelity model, then check if the plan is valid in higher fidelity models. If not, it is assumed the higher fidelity model contains information relevant to a potential plan failure, and so the plan is patched using that model. In this way, the planner can use lower fidelity (and typically computationally less expensive) models, when appropriate, but still make use of higher fidelity models, when necessary.

This approach differs from standard abstraction and HTN planning in two major ways. First, unlike abstraction and HTN planning, any of the models in the hierarchy can produce a directly executable plan, thus, it is not necessary to plan in multiple models if the situation is "simple" enough. Second, rather than a linear, predefined ordering of models, as with abstraction planning, our approach supports a complete lattice of models. Different models in the lattice make different information explicit – for instance, one model may represent vehicle dynamics while another model may represent shape and material properties of the vehicle. Thus, choosing with which model to plan, in a given context, is a key issue with this approach, and one that we are actively researching.

While it is inevitable that all models of the real world are incomplete, to some extent, we can use that knowledge to our advantage in designing planners that explicitly handle sources of incompleteness, either probabilistically or through explicit model choice, and learners that can efficiently improve models through experience. The ultimate goal is to develop robotic systems that can reason about the incompleteness of their models and actively characterize and improve them, through interactions with the environment.

3.33 On the Shoulders of Giants: The Case for Modular Integration of Discrete Planners and Continuous Planners for Robotics

Siddharth Srivastava (UTRC – Berkeley, US)

The field of automated planning has its roots in planning for SHAKEY, a problem-solving robot created at SRI in the 70s. For the most part, the modern planning paradigm for autonomous robots assumes a similar architecture, making a clean separation between the abstract problem of task planning with discrete states and actions, and the problems of motion planning and control synthesis involving the computation of continuous trajectories in a high-dimensional configuration space. However, for the most part these fields have developed in mathematical isolation from each other. Naïve approaches for reconciling them (e.g., first computing a task plan and then "implementing" each action through a motion planner) result in solutions that are unexecutable because of the lossy abstractions required for constructing task-level models.

The situation in planning for robotics is thus not very different from the state of model checking just before the advent of SAT-modulo-theories (SMT) techniques. Many of the motivators for SMT research hold in our setting: we have efficient solvers (planners) at each level of abstraction; modeling entire tasks at the finest level of granularity is cumbersome when it is not impossible and typically results in computationally intractable problems. Indeed, from the point of view of making collective progress, developing new "task and motion" planners from scratch for various formulations (deterministic, non-deterministic or stochastic models of actuators with deterministic, non-deterministic or stochastic models of sensors for single or multiple agents), most of which are addressed independently in these fields, would be inefficient at best and reinvention in the worst case.

I believe that these observations provide evidence for new opportunities as well as challenges for innovative research in planning for robotics, geared towards the modular utilization of existing paradigms for discrete sequential decision making, motion planning, and control synthesis in a hierarchical planning paradigm. Opportunities stem from a perspective that allows us to learn from the development of SMT solvers. The challenges, and the corresponding domains of innovation, arise from the numerous aspects of planning for robotics that are not addressed in the existing theory for SMT solvers; a direct application of those techniques is unlikely to succeed. We need new research on rigorous methods for the synthesis and analysis of abstraction functions that translate planning problems between different levels of the hierarchy. Existing domain description languages require new constructs and semantics to succinctly and correctly express abstractions of sequences of low-level actions (existing representations lead to incorrect models and unexecutable solutions). Solutions to motion planning problems are continuous trajectories that need new symbolic representations suitable for high-level reasoning. Finally, uncertainty in sensing and actuation in each level of abstraction makes it harder to construct "lemmas" for use at the next higher level of abstraction.

3.34 Persistent, Instructable, Interruptible, Transparent Autonomy

Manuela Veloso (Carnegie Mellon University – Pittsburgh, US)

Besides the Roomba robots, and instances of service mobile robots in some hospital, public environments, there are not many autonomous robots that persist in real human environments. We have experienced the CoBot mobile robots that have persisted at Carnegie Mellon capable of autonomously performing navigation and service tasks for the last 5-6 years. (There are the Google-Uber-Baidu-etc self-driving cars, which are becoming a reality.) We are interested of further understanding how to better plan for such persistent "co-existing" robots.

The real world offers challenges to robot autonomy, at all levels, namely perceptual, cognitive, and actuation. Focusing on the real world where humans live, robots further concretely face navigation, timing, quality, and interaction challenges. Planning involves having models of the world, and models of actions. It is challenging, or most probably impossible, to get appropriate and correct models up front: the real world is dynamic, uncertain, personalized, and it includes other external, at best poorly modeled, actuators, such as other robots and people.

We are interested in the issues underlying robot autonomy, in particular planning, that needs to persist, exist, improve in the real world. Along this goal, we pursue research on instructing and correcting a robot, interrupting the execution of its plans, and making it be transparent about its actions and plans.

We have developed instruction graphs as a way for humans to provide verbal command-based instructions to become procedural plans. The robot is equipped with action and sensing primitives that the instruction graphs then organize in sequencing, conditionals, and looping constructs. Upon execution of the instructed graphs, humans can check that the plan is suitable or not, and correct it as needed. As the human knows the plan that is being executed, we conjecture that it may be easier for the human to correct the robot behavior than if the planning model were non-procedural actions. Such instruction graphs are independent of the robot platform. We have used instruction graphs for our CoBot and Baxter robots. We have also addressed the problem of acquiring a library of plans, as instruction graphs, for different tasks. The challenge is to recall a similar past learned plan and reuse it. We have also researched on generalizing different plans and proposing autocompletions of possible plans when a human is instructing or correcting a robot. We believe there is a lot left to do in terms of providing, generalizing, revising, and reusing plans.

We also research on the challenge of enabling a robot with the ability to replan when interrupted. When the robots generate plans to achieve their tasks, they then execute them determined with the sole goal of executing their plans. If CoBot is executing its plan to deliver a package to someone's office, and Manuela finds CoBot in the middle of the corridor, and she wants to tell "CoBot, thanks for the package, I got it!", she can't: CoBot will go all the way to the destination office to finish its delivery task. Tasks can of course be managed and interrupted remotely through an administrative interface, but not by naturally interrupting the robot. We have created an approach to interrupt a robot, but there is a lot left to do in terms of investigating the need to replan when and how, so that the robots can take input from humans about their plans and new task requests. In the general scenario of our research, in which robots encounter a wide variety of people, and not just the robot developers, such interruptible autonomy is a challenging research question, as the robot needs to be able to evaluate the requests and attend to them according to models of priorities, authority, level of accomplishment and type of tasks under execution and scheduled. Replanning becomes a question, which is not just a function of the failure of preconditions, but includes dynamic task optimization and model learning.

Autonomous robots that plan cannot be opaque to their human users. We research on methods to increase the transparency of the robot planners, such that the robot explains its choices, and reveals the actions selected, as well as possibly future actions. Interestingly, robot act according to plans that include future actions, so they can make their future actions known to humans, who could potentially better understand and possibly correct the intentions and plans of the robots. We have developed a verbalization algorithm which enables a robot to describe its experience in natural language. The robot can transform its planned and executed route into natural language. We also research on multiple techniques to enable a robot to be transparent to humans in their decisions, planning, and learning. We believe there is a lot left to do in terms of augmenting planning algorithms with the ability to increase their transparency towards humans. We research on expressive lights to improve the understanding of the robot's actions, on verbalization in different dimensions to generate varied descriptions of experience in natural language, and on augmenting video capturing of robot's plan execution with markings that aim at visually explaining the robot's performance.

In summary, we are interested in planning, as an integral part of a persistent autonomous mobile service robot, that needs to persistently interact with humans. We will discuss our research and open planning research directions in robot instructability, interruptibility, and transparency. Our underlying approach assumes that robots need to learn improved planning models over time with experience.

Participants

- Rachid Alami
  LAAS – Toulouse, FR
- Iman Awaad
  Hochschule Bonn-Rhein-Sieg – St. Augustin, DE
- Roman Bartak
  Charles University – Prague, CZ
- Michael Beetz
  Universität Bremen, DE
- Ronen I. Brafman
  Ben Gurion University – Beer Sheva, IL
- Michael Cashmore
  King's College London, GB
- Martin Davies
  Guidance Automation Ltd – Leicester, GB
- Minh Do
  NASA – Moffett Field, US
- Susan L. Epstein
  City University of New York, US
- Alberto Finzi
  University of Naples, IT
- Hector Geffner
  UPF – Barcelona, ES
- Malik Ghallab
  LAAS – Toulouse, FR
- Nick Hawes
  University of Birmingham, GB
- Malte Helmert
  Universität Basel, CH
- Andreas Hertle
  Universität Freiburg, DE
- Joachim Hertzberg
  Universität Osnabrück, DE
- Laura M. Hiatt
  Naval Research Lab – Washington, DC, US
- Michael W. Hofbaur
  Joanneum Research – Klagenfurt/Wörthersee, AT
- Jörg Hoffmann
  Universität des Saarlandes, DE
- Felix Ingrand
  LAAS – Toulouse, FR
- Luca Iocchi
  Sapienza University of Rome, IT
- Gal A. Kaminka
  Bar-Ilan University – Ramat Gan, IL
- Erez Karpas
  Technion – Haifa, IL
- Oliver E. Kim
  University of Birmingham, GB
- Sven Koenig
  USC – Los Angeles, US
- Lars Kunze
  University of Birmingham, GB
- Bruno Lacerda
  University of Birmingham, GB
- Gerhard Lakemeyer
  RWTH Aachen, DE
- Daniele Magazzeni
  King's College London, GB
- Lenka Mudrova
  University of Birmingham, GB
- Daniele Nardi
  Sapienza University of Rome, IT
- Tim Niemüller
  RWTH Aachen, DE
- Andrea Orlandini
  CNR – Rome, IT
- Amit Kumar Pandey
  Aldebaran Robotics – Paris, FR
- Simon Parsons
  King's College London, GB
- Ron Petrick
  Heriot-Watt University – Edinburgh, GB
- Mark Roberts
  Naval Research – Washington, US
- Enrico Scala
  Australian National University – Canberra, AU
- Matthias Scheutz
  Tufts University – Medford, US
- Reid Simmons
  NSF – Arlington, US
- Elizabeth Sklar
  King's College London, GB
- Stephen Smith
  Carnegie Mellon University – Pittsburgh, US
- Siddharth Srivastava
  UTRC – Berkeley, US
- Manuela Veloso
  Carnegie Mellon University – Pittsburgh, US
- Brian C. Williams
  MIT – Cambridge, US