Health, demographic change and wellbeing

Personalising health and care: Advancing active and healthy ageing

H2020-PHC-19-2014

Research and Innovation Action

GA 643644

Deliverable 5.6
Activity monitor: Algorithms for user activity monitor

Deliverable due date: 05-2017	**Actual submission date: 31.05.2017**
Start date of project: February 1, 2015	**Duration: 42 months**
Lead beneficiary for this deliverable: UNITN	**Revision: 1.0**
Authors: Paolo Bevilacqua, Marco Frego, Thomas Given-Wilson, Axel Legay, Sean Sedwards.	
Internal reviewer: Daniele Fontanelli, Luigi Palopoli	

The research leading to these results has received funding from the European Union's H2020 Research and Innovation Programme - Societal Challenge 1 (DG CONNECT/H) under grant agreement n°643644		
Dissemination Level		
R	Restricted	
P	Public	**X**

Contents

1 Executive Summary

This deliverable contains a description about the technology for monitoring activity plans. The work refers to D5.2 (User reactive planning) and builds on D2.6 (Activity plans representation). It is delivered in parallel with D5.4 (Activity planning).

We propose that the Activity Monitor will be implemented on the *FriWalk* and make decisions based on the most recent available global state of the activity and its history. The Activity Monitor on a *FriWalk* always makes its decisions based on the most reliable information available.

Each time an activity plan is provided by the Activity Planner, we will use a mixture of probabilistic reasoning and quantitative metrics to monitor the performance of an activity. Some metrics will provide feedback to the participants (e.g., slow down), some will trigger re-planning of the activity (e.g., obstructions), some will trigger a cancellation (e.g., an event is cancelled by the user because the hard constraints cannot be met, multiple uncooperative behaviour) and some will just be logged for use by the Activity Evaluator.

This report completes the work started in P1 with the production of a preliminary document regarding the Activity Monitoring (D5.5). The innovative contributions with respect to the preliminary version are: 1. We divide the problem of activity monitoring into two different sub-problems, i.e. a high—level monitoring regarding the whole activity and the associated global constraints, and a motion monitoring considering the local constraints posed by the environment during the motion of the FriWalk along a path (e.g. avoid collision with surrounding people). 2. The activity monitor considers the probability distributions of the different parameters associated to each action (and generated by simulations during the planning phase) to assert the probabilistic fulfilment of the high—level constraints (in the first version, simulations were performed directly by the monitor, and the constraints were asserted through the application of techniques of statistical model checking). 3. Section 6 (Semantic representation of activity plans) has been updated to reflect the latest updates regarding the modelling and generation of activity plans (as illustrated in the deliverables D2.6 and D5.4).

2 Introduction

The decisions about an activity will be principally made by the Activity Monitor on individual *FriWalk*s, but based on a global view of the activity obtained from the infrastructure.

As described in Deliverable D5.4, the planning of an activity is characterised by two logical steps, i.e. the synthesis of a high-level sequence of tasks and its refinement in an executable plan. This splitting reflects naturally the shape of the Activity Monitor, which consists in two different modules: The Activity Execution Monitor and the Motion Execution Monitor. The first is related to the high-level planner and checks that the current realization satisfies the hard constraints with acceptable confidence. The second is part of the Reactive Planner and takes care of the non-predictable real status of the environment (e.g. the presence of pedestrians is contemplated but their actual position is unknown until runtime), captured by the sensing system. Whenever the path is declared unfeasible (in probability) a local re-planning of the path is enforced.

An important extension of the Motion Execution Monitor is to handle group activities, e.g. a fleet of walkers that share the same high level plan. For the walkers that are part of that group, the Monitor has to accept the proximity of the other participants and should not invoke a dynamic re-planning. The correct policy is to slow down and follow the leading agents. Such agents are usually placed in front of the group or are selected on the base of their field of view in order to prevent occlusions and detect possible obstacles. This group cohesion is modelled considering the principal clothoid spline of the Motion Planner with a sufficiently wide offset. We call this structure a clothoid tunnel, which has been discussed in the Deliverable D5.2 related to the Reactive Planner. The policy is thus to remain in the centroid of the group keeping the fleet of walkers in a certain configuration, e.g. in rows.

In the case of group activity, the global view will be updated frequently while a *FriWalk* maintains contact with the cloud. Under circumstances of reliable communication, the cloud and all *FriWalk*s will work with an up-to-date view of the current status and make implicitly coordinated decisions (since they use the same software). Under circumstances of poor communication, each *FriWalk* will make decisions based on its most recent global view, updated with accurate local information from its own sensors.

There is related work in the fields of robotics and multiplayer games, but the particular challenges described above are somewhat unique and distinct from robotics. Humans are not robots and should not be forced or coerced to follow a rigid path except, perhaps, under extraordinary circumstances. This would defeat the objective of ad hoc social interaction and personal preference. Moreover, the environment is not fixed (by virtue of the existence and movement of other pedestrians and unforeseen obstacles), hence the notion of an a priori "optimal plan" is to be intended in weak sense: the optimality is subject to the information available at the moment the plan is designed. In ACANTO we undertake a dynamic group motion planning and monitoring problem, where the notion of optimality is defined by respecting hard constraints and optimising the quantitative metrics of an activity.

We will use a mixture of probabilistic reasoning and quantitative metrics over traces to monitor the performance of an activity. Some metrics will provide feedback to the participants (e.g., slow down), some will trigger re-planning of the activity (e.g., obstructions), some will trigger a cancellation (e.g., multiple uncooperative behaviour) and some will just be used to help improve the activity for the future. The activity monitor will principally work at the low level of sensor data (mass, velocity, position, etc.), based on a specification compiled from an activity plan defined in a variant of the PDDL language, as outlined in D2.5.

3 Context and challenges

This section presents the context and challenges for the ACANTO project with respect to the monitoring of activities. An activity in ACANTO has the intention to provide therapeutic benefit and enjoyment for each user of the system. Therapy includes engaging in exercise and thus burning calories. Enjoyment includes visiting places and seeing sights of interest. Social interaction with friends can provide both therapy and enjoyment, so activities are primarily intended for a group of participants. Both individual and group-related metrics are therefore of concern to the Activity Monitor.

We assume that the Activity Planner will create a suitable Activity Plan to meet the needs and constraints of all the intended participants, based on input from the Activity Generator. This includes ensuring that for each participant (*FriWalk* user), a feasible path exists that satisfies the user's constraints and requirements. That is, a path exists that avoids the user's undesirable areas and is sufficiently challenging, without being too challenging, to meet the user's therapeutic requirements. Meeting all the users' constraints and requirements suggests that the group of participants are likely to have similar profiles. Even if this is so, activities may need to include enforced pauses to allow slower participants to catch up. The chosen route of the activity must have sufficient capacity (free space along the way) for the number of participants.

Once a plan has been decided by the Activity Planner and initiated, its execution needs to be constantly monitored. The purpose of monitoring is to ensure that: (*i*) the users' hard constraints are met (i.e., always … , never …); (*ii*) the users' soft constraints are met most of the time (i.e., try to …); (*iii*) the users' therapeutic goals are met; (*iv*) the social and enjoyment goals of the activity are met (interaction between participants and with the environment) and, finally; (*v*) to log the activity for offline evaluation by the Activity Evaluator.

Implicit in the achievement of (*i*)–(*v*) above is the ability to respond to changing environmental conditions and uncooperative behaviour of the participants. The changing environmental conditions here includes unforeseen obstructions that may prevent the activity from following its intended course, rather than the instantaneous obstructions handled by the Reactive Planner. Uncooperative behaviour refers to participants who choose (or are forced) not to follow the activity plan.

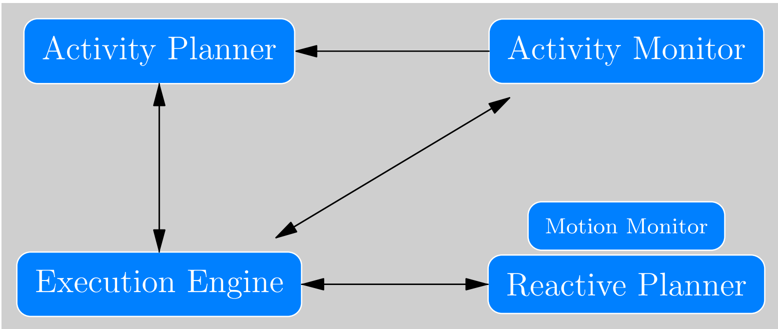


Figure 1

Figure 1 illustrates the basic relationship between the Activity Planner, the Reactive Planner (with the Motion Monitor), the Execution Engine and the Activity Monitor. The Activity Planner constructs an activity plan from a palette of alternatives, considering all the preferences, hard constraints and soft constraints of all the participants. Each Reactive Planner monitors its user's physical location with respect to the nominal plan, other moving and static objects in the environment, and the constraints it is provided (e.g., desirable proximity to others). Users may not necessarily follow the suggestions, but at all times the sensors calculate the user's actual position with respect to surrounding objects and upload the information to the cloud. This global information is curated and made available to each *FriWalk*, for the use of their Reactive Planner and Activity Monitor. The Activity Monitor monitors the user's progress with respect to the high level goals of the activity (e.g., adherence to intended route, group coherence, calories burned, exercise) and reacts accordingly to any deviation.

Unforeseen obstructions that cause the activity to be re-planned will be handled locally, however the information about the obstruction will be uploaded to the cloud for distribution to the other *FriWalk*s. Depending on the nature of the obstruction and blockages, the Activity Monitor decides if the activity remains feasible (e.g., if a feasible re-routing is possible), otherwise it signals the information to both the Planner and the Execution Engine.

In D2.6 (Activity plans representation) we identified the PDDL class of languages as suitable to represent an activity plan at a syntactic level. Variants of PDDL contain all the quantitative, temporal, probabilistic and nondeterministic elements that we need, decomposing the activity into a sequence of tasks. At an abstract semantic level, an activity is the traversal of a graph annotated with metric information. An activity plan will be devised using the graph resulting from an abstraction of the environment. Thus, the Activity Monitor will consider the traversal of a graph, whose nodes are the Points of Interest defined in PDDL.

In Section 5 we identify the notion of a user's hard and soft constraints, respectively those which must be adhered to and those which it merely desirable (perhaps probabilistically) to adhere to. In addition to general constraints, a user's profile may also include activity-specific constraints, such as the minimum and maximum amounts of exercise required / allowable or the desired number of calories to burn. The activity itself will have global requirements, such as the places to visit and the desired amount of group cohesion (how well the group stays together).

The success of an activity will be judged by how well it achieved its global objectives and by its *worst* performance with an individual participant. Considering an average or sum of individual performances could mask the fact that the activity under-performed with some participants. The Activity Monitor will thus need to keep track of both local and global metrics, i.e., metrics that apply to individuals and to the group as a whole. In general, such metrics will be temporal (include timing constraints), history dependent (calculated over a path, not just a state) and monitored continuously. During the course of an activity, the Activity monitor will use these metrics to: (*i*) influence the behaviour of participants without affecting the activity plan (e.g., cause a participant to slow down); (*ii*) modify the activity (e.g., re-routing on the graph to avoid an unforeseen obstacle); (*iii*) stop the activity (e.g., due to uncooperative behaviour by multiple participants) and, finally; (*iv*) to log the activity for post-processing by the Activity Evaluator.

4 State of the art

In this section we briefly review the literature closely related to the challenges of the Activity Monitor and with regard to the technology already chosen for the management of other aspects of the activity.

In D2.6 (Activity plans representation) we made the case to view our planning problem as a sequence of discrete control actions that change the configuration of the system. The motion planning problem thus becomes a task planning problem, with the motion being emergent of completing the tasks. This discretisation simplifies the consideration of motion in continuous space and provides a simple metric for analysis and monitoring.

We identified the PDDL class of languages, standard in the field of robotic planning, as being suitable to represent an activity plan at a syntactic level. PDDL generally allows tasks to be defined with pre- and post-conditions. Specific variants of PDDL allow the expression of complex temporal and quantitative constraints to complete an overall mission [3] [4] [5] [6]. Other variants permit the inclusion of soft constraints via probabilities [7] and the inclusion of nondeterminism to model partial knowledge [8] [9]. Of particular relevance to group activities are extensions to include multiple agents [10]. While there are no off-the-shelf solutions to our group motion planning and activity monitoring problems, there are a number of pre-existing open source parsers of PDDL languages [14] [15] [16] which we may incorporate.

Since ACANTO activities are intended for a group of participants, we briefly review some of the previous work that considers group motion planning and monitoring.

Collaborative planning that involves both robots and humans has been considered before [17]. The authors develop some notions of partial plans, shared plans, and how to achieve successful collaboration without conflicts along the way. The approach is a general discussion of ways that such plans can be represented and reasoned about. While interesting in a foundational sense, and also in the sense that both robotic and human agents are considered, this does not really apply well to our domain. The main limitations are that the notion of plans does not align with those of ACANTO, and the authors explore foundational concepts rather than the implementation details required for a working system.

Some similar ideas are considered in [18] where the authors also consider collaborative planning between humans and machines. However, the focus of their work is on architectures for supporting such planning, rather than the specific capabilities of particular machines. Thus, although such architectures are interesting in a broad sense, they do not align with the specific problems posed by the ACANTO project.

In [19] the author makes arguments for augmented distributed artificial intelligence (DAI) being able to improve solutions to multi-agent collaborative and / or competitive systems. They present examples and arguments from game theory and physics as well as DAI background to support their approach. This paper is mostly a position paper on what could be done and makes good arguments for cross-disciplinary approaches. However, since there are not significant results and development of the concepts in a meaningful sense for our applications, this remains interesting but not directly related.

In [20] the authors present and *architecture for behaviour-based agents* (ABBA) that can be used for planning and collaboration between robots. The ABBA exploits communication between agents to allow for collaborative response to requirements, and this exhibits collaborative planning behaviour. However, this does not support the kinds of behaviours we wish to consider for this project, since the robots focus on small command tasks and do not have a larger plan that would correspond to our activity plans. Although we could potentially shoe-horn activity plans in, the underlying assumptions about behaviour (robot as opposed to human), communication (reliable as opposed to unreliable), and activity (robots here do not activate at all times, here we cannot turn humans "off"), this does not fit well with our scenario.

Centralised planning approaches are considered in [21] where the authors show that existing efficient centralised planning approaches can fail to find an optimal solution for path planning for multiple robots. They then explore some different approaches to planning that can consider different constraints and requirements of the robots involved. They show that even taking these into account they can improve the overall planning results by randomising some aspects of path searching. Although somewhat interesting, the domain of the research is quite different. The research focuses on robots, while in ACANTO our agents may not be cooperative and are human. Further, to do the planning as in [21] requires stricter communication and centralisation than we assume in the ACANTO project. Thus, although some of the key ideas may be interesting for solving the overall issues of handling many constraints at once, the domain is quite different.

There is a body of research that may appear related in *swarm* planning for robotic agents [22] [23] [24] [25] [26] [27]. The key concepts are for various agents to collaborate in finding paths or routes within an environment that allows them to navigate. Further, such systems are typically meant to operate with agents having limited information about the environment, limited communication capabilities, limited computing capabilities, and to be ``fault tolerant''. Superficially this seems to match the goals of ACANTO quite well, since we also consider (in this deliverable) problems related to path (that can be as an abstraction for an activity) planning for multiple agents with similar limitations. However, the differences become quite significant when further detail is considered. The various agents in swarm systems do not have the proximity requirements that groups do in the ACANTO system, instead swarms spread out in some phases to improve their path finding. Often swarm path finding is related to finding a path to some objective whose location is unknown, while in ACANTO the activity plan generally identifies a path on a larger scale, with small detail left to the reactive planner. Relating to this, the swarm agents tend to become landmarks or locations as part of some swarm approaches, whereas in ACANTO we do not consider leaving agents alone as markers of paths. Lastly, the fault tolerance of swarm systems is typically in allowing failure of some (or many) of the agents as long as a larger goal is achieved. In ACANTO we cannot allow "failure" for any agent for many conditions, so instead we must operate in a much more restrictive manner towards agents' needs. There are other differences as well, but for these reasons we largely cannot adapt ideas or concepts from such works on swarms of agents.

Shifting further into collaboration in planning between agents, in [32] the authors consider how to improve path planning algorithms for multi-agent teams. The main development over prior work [33] [34] [35] is in allowing agents to modify the plans of other agents during path planning. This is in addition to respecting various constraints and requirements for each agent. Although some of the key problems are similar, and the modification of other agents' plans may be useful to consider in ACANTO, the approach does not easily transfer. The path planning is quite different in that the focus of [32] is collision avoidance and path optimisation, while in ACANTO paths would be made to coincide as much as possible. Also, their approach considers stronger coordination requirements than we assume in the ACANTO project, and more centralised planning. Although some of the concepts may adopted to some degree, the core algorithms do not easily transfer.

In [36] the authors consider how to collaboratively find paths for road vehicle coordination. This has similarities to the ACANTO project in that planning is done by agents in a collaborative manner, and may involve actions that come from humans (drivers) rather than strictly robots (self-driving vehicles). They adapt some centralised planning approaches and show how they can be modified to work to some degree in a decentralised scenario. They focus upon rapid results as the paper concerns collision avoidance. The approach presented is useful to observe the adaptations from centralised to decentralised planning, although the domain for collaborative monitoring and planning in ACANTO is less time constrained as collision avoidance is handled by the reactive planner.

5 Case studies

We briefly recall three case studies identified by WP1 and used in D2.5. The resulting hard constraints, soft constraints and preferences are summarised in Table 1, for reference in other sections of this document.

Isabel

Isabel is an 82 years old woman who has lived alone for the past two years. She lives in a flat by herself in Newcastle. She no longer goes out very often and has become very physically inactive, even if her doctor suggested to stop with this unhealthy behaviour. Her daughter brings her groceries once a week. While Isabel used to enjoy going for walks, she no longer has anyone to go walking with. She recalls the times she spent walking with fondness and wishes that she had someone to go walking with. One day, she receives an invitation by mail to try out the new *FriTab* and *FriWalk* system. After receiving the system when a researcher visits her, she tells the system about her background and interests. She tells the system that she used to enjoy walking. Later that day, the *FriTab* suggests that she meet a lady in the next street, Martha, who likes to visit the local shopping mall and has the same platform. The system has noticed that Isabel does not have many friends and believes that if she had a friend who also enjoyed going for walks, she might go there again. Isabel is hesitant at first but then agrees to meet up and try out the *FriWalk*. The *FriTab* tells Isabel to meet Martha the following Wednesday at 10am to enjoy a morning together at the shopping mall. Once she arrives to the shopping mall, the *FriWalk* shows the directions to get in touch with Martha at the prescribed time. Since Martha has a similar *FriWalk*, the two ladies meet with any problem. Isabel and Martha go for a walk in the mall and decide to buy some groceries on their own. The *FriWalk* suggests the route and monitor the execution of the activity, in order to report to her medic the physical activity that has been carried out. During the walk, the *FriTab* realises that Martha feels a little bit tired and suggests to interrupt the planned activity. The *FriTab* suggests Isabel and Martha to have lunch in a local cafe. The *FriWalk* devices guides them gently to the desired cafe where they have a pleasant lunch and agree to meet up again in the following days.

Michael

Michael is a 72 years old man who lives alone in Felling, Gateshead. For the past few years, he has found mobility very difficult and he is waiting for a hip operation. Consequently, he doesn't get out much. He used to enjoy visiting museums and now fulfils his passion for natural history by watching documentaries on TV. He would like to be able to get out to visit the museums in Newcastle.

A researcher visits Michael one day and shows him the *FriTab*. Michael explains to the researcher that he has mobility problems so wouldn't be able to get out much. But the researcher explains the *FriWalk* to him which is owned by several shops, galleries and museums in the area. He also explains that people on the *FriTab* network may be able to help him get transported to locations and events. So Michael enters his details into the system and tells it that he has mobility problems. The information on Michael's profile are also updated by his doctor, who also enters some constraints concerning the activities Michael can safely carry out.

The next day, the *FriTab* shows Michael that a tour is being organised at the Hancock Museum. It invites him to attend and tells him that another person attending would be willing to pick him up. The system knows that Michael enjoys natural history and that he also has mobility problems. It knows that the museum has several *FriWalk* devices that can help Michael. It also knows that one other attendee has a car and is willing to transport friends. Michael is hesitant but agrees to give it a try. So he tells the system that he will attend. The *FriTab* tells him that Jane will pick him up before the event in her car. Michael tells her his address.

At the arranged time, Jane picks Michael up and they drive to the museum. When he arrives at the museum, he is given a *FriWalk* which helps him to walk with the rest of the tour group. After the tour is over, the *FriWalk* even suggests a guided tour of its own that Michael can do alone without violating the medical prescriptions. However, Michael is tired but decides to come back and try the guided tour another day. The *FriTab* forwards the activity log file to the network for user profile updates.

Dorothy

Dorothy is a 69 years old woman who lives alone in Blaydon. Dorothy uses a walker to get around because she finds that it gives her confidence after her fall one year ago. Moreover, it helps her in keeping a constant physical activity for a correct rehabilitation, according to the medical prescriptions. Dorothy loves shopping and likes it when her friend occasionally takes her shopping at the MetroCentre. While she likes the MetroCentre, she is nervous about going there alone and worries that she would get lost. But she would like to go there more often.

Dorothy is shown the *FriTab* and told that it clips onto *FriWalk* devices which are available at the MetroCentre. She decides to try out the *FriTab* system. Several days later, the system suggests that Dorothy visit the MetroCentre to enjoy some shopping. The system has noticed that Dorothy has stayed indoors for several days and believes that she would benefit from getting out. Dorothy thinks that it would be a good idea and asks the *FriTab* for more information. The *FriTab* suggests that she get the 2:15 bus from the nearby bus stop which will take her to the MetroCentre. It tells her that it will give her directions to the MetroCentre and will help her find her way around inside.

She gets the bus and travels to the MetroCentre. When she gets there she swaps her walker for a *FriWalk* and clips in her *FriTab*. The *FriTab* shows her that several shops have sales and gives her directions. Furthermore, it also advices her of the presence of her friends Rita and Marion, both equipped with a *FriWalk*. Dorothy meet them in front of the Central Café and then take a walk in some shops. The *FriTab* suggest them to go to the theatre inside the mall to see a romantic movie. After the movie, Rita and Marion decided to go home, while the *FriTab* suggests Dorothy to visit the mall first floor to take a look to some very affordable items at the shoe shop. Meanwhile, Dorothy accomplishes her daily schedule of physical activities. After a while, when she starts feeling tired, she presses a button on the *FriTab* and it directs her back to where her walker is. She unclips her *FriTab* and it tells her where to get the bus home.

Use Case	Hard Constraints	Soft Constraints	Preferences
Isabel	1. Always remain ≤ x m from the bathroom	I. Level of tiredness always < y	A. Walking B. Meeting someone who enjoys walking C. Visiting the grocery store
Michael	2. Avoid stairs 3. Maximum walked distance < x m 4. Speed always < y	II. Avoid crowded rooms	D. Interest in natural history E. Visiting museums
Dorothy	5. Never remain alone 6. Walk ≥ x meters 7. Walk ≥ y minutes (medical prescriptions)	III. Waiting at ≤ x minutes at the bus stop	F. visiting shops with sales G. visiting clothing shops H. watching romantic movies

Table 1 Summary of use cases

6 Semantic representation of activity plans

This section describes the concepts behind our semantic representation of activity plans, highlighting the consequences of our choices for the Activity Monitor.

In D2.5 (Activity plans representation (preliminary)) we proposed a graph-based semantics to underlie activity plans. This approach uses familiar concepts from GPS satellite navigation and has already been successfully applied in a related assistive technology context [11] under the DALi project [1].

Nodes are annotated by their semantic location in the environment (e.g., supermarket, bakery, toilet) and with their physical coordinates. Likewise, edges are annotated (weighted) with the distance between their adjacent nodes. To account for user preferences (e.g., not straying too far from a toilet – hard constraint 1 in Table 1 – or not getting too close to a noisy arcade – soft constraint II in Table 1) and avoiding general crowding (observed by cameras in the environment), the edges are made directional and "lengthened" (their weight is increased) according to crowding and desirability of the destination node [11]. Impassable temporary obstructions are handled by removing sections (subgraphs) of the graph.

In D2.6 we identified the PDDL class of languages as a suitable means to represent activity plans at a syntactic level. These languages transform a motion planning problem into one of task planning, decomposing the problem into a series of goals and making it easier to deal with motion in continuous space. Variants of this language have incorporated notions of rewards (e.g., to model calories burned), nondeterminism (e.g., to model partial knowledge) and probabilities (e.g., to represent soft constraints), which are necessary for our application and align well with our statistical model checking technology.

In D2.6 we showed a simple direct correspondence between a global plan (the a priori optimal path through the graph abstraction of the environment) and its description in PDDL. An activity plan must of course account for multiple users and therefore multiple trajectories. The concrete trajectories of participants will be concurrent in time and space, so must be disjoint. This creates a question as to how the global plan of a group activity should be represented.

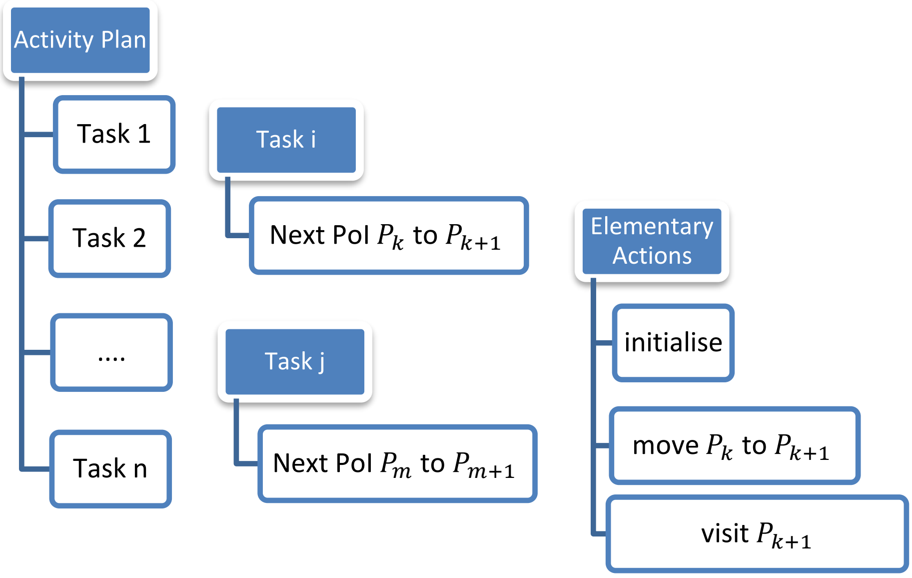
An activity plan could conceivably comprise a path in the graph, chosen to satisfy all the constraints and objectives of the activity and its participants. The abstract plan is translated into a single path made of a sequence of clothoid tunnels that encompass the whole group of participants. A participant's Reactive Planner will then try to stay within some maximum distance from the centroid of the group while remaining inside the tunnel.

Within a group activity there can in general be no a priori notion of an optimal path, since concepts such as socialisation are effectively nondeterministic. That is, how the participants interact within the group cannot be known or specified in advance, hence it is not possible to optimise. Our use of PDDL allows us to treat the activity plan as a transition system and thus apply probabilistic reasoning and optimal controller synthesis to activity planning and monitoring. Given a transition system with nondeterminism, it is possible to calculate the maximum and minimum of metrics over executions of the system (e.g. min/max estimation of time or distances for completing an action). If the nondeterminism can be refined to a probabilistic distribution, it is possible to calculate or estimate the expectation of the same metrics.

The Activity Planner may thus create an activity plan that is provably capable of achieving the goals of the activity in a probabilistic sense. The Activity Monitor will report the actual performance of the activity, which can then be compared with its expected performance bounds by the Activity Evaluator. Whenever the success probability of achieving a goal decreases below a certain threshold value (but the goal can reasonably still be achieved) a warning is shown to the user on the FriTab. For instance, invite the user to move a little faster, interrupt the visit to the current Point of Interest, etc. If the user is not collaborative, and the goal becomes unfeasible (e.g. probability of success below a minimum value), the Activity Planner is invoked to update the activity plan on the basis of the soft and hard constraints (e.g. remove some Points of Interest from the plan).

7 Activity monitoring with runtime verification

We recall briefly the structure of the high level activity plan, produced by the Activity Planner. This plan takes care of the sequence of tasks required to satisfy the needs of the user, e.g. finds the best path and times to visit some Points of Interest, respecting opening times, etc. Each task is subdivided into elementary actions, which are of low level, e.g. move from P_0 to P_1 on a particular clothoid.



Each elementary action has its own properties that are evaluated by the Monitoring Service. The most important are the duration and the distance, but there can be more. Those properties are used to check if the global plan is still feasible or not. For instance, if a particular action takes too long to be completed, it is possible that the next steps become unfeasible, e.g. a shop closes, we miss the bus, the user is too tired, etc. There are a number of such constraints that although feasible from the theoretical point of view of the high level plan, can become unfeasible in practise. The reasons of those failures can be addressed in two main categories: user decisions and needs, external causes as unforeseen obstacles and problems. Some failures produced by an user's decision may be if the user enjoys the Point of Interest and prolongs the visit more than the estimated time, encounters a friend and stops to chat. In the second category (external problems) threre are, for example, obstacles on the path or crowded zones that cause a long waiting time.

Therefore, the Monitoring Component is divided into two parts: The Activity Execution Monitor and the Motion Execution Monitor. The first component checks if the current status of the execution respects the constraints so that the plan can be completely accomplished, the second component is of lower level and is used to analyse the single elementary actions that build the activity plan. The external problems are herein considered, for example obstacles on the path, that require a deviation and therefore an increase in the execution time of the elementary action.

7.1 Activity Execution Monitor

All the activities executed by the FriWalk are composed by a sequence of tasks, where each task consists in a sequence of elementary actions to be executed (see D5.4). During the execution of an activity, the Activity Execution Monitor service is required to periodically check the execution status, and to verify that the probability of violating the hard constraints associated to the activity remains under a certain threshold value.

In particular, given a hard constraint (e.g. the total duration of the activity is lower than 3 hours), and the current status of the execution (e.g. the user is still performing the j^{th} task at the current time t), the probability of violating the constraint corresponds to the probability that the sum of the times required to complete the current and all the subsequent tasks is greater than the remaining time. As explained in deliverable D5.4 (Activity Planning), we assume to know the distributions of probabilities for the physical parameters regarding the different actions performed during the execution of an activity (e.g. duration), that are collected from the KnowledgeBase, and from simulations performed directly on maps of the environment, considering the scenario identified by the current real time information obtained from the sensing system. Thus, the probability density function for the duration of a sequence of actions is computed as the convolution of the density functions associated to each action.

Whenever the Activity Execution Monitor is invoked, it computes the probability of violation for each of the hard constraints related to the activity. Whenever the monitor discovers a hard constraint that could be violated with a probability greater than some given threshold value (e.g. 10%), it marks the current activity as unfeasible, and invokes the Activity Planner to generate a new plan, according to the mutated scenario.

7.1.1 Support for groups

The Social Activities within ACANTO are characterized by their "social" aspect, and thus are in general shared and executed by more users. However, for all the users belonging to the same group, the sequence of actions defining the Activity is the same, while the set of hard and soft constraints and preferences will be different for each user. Since the Activity Execution Monitor is required to ensure the fulfilment of all the hard constraints (at least in a probabilistic sense), in the case of a group of users, the hard constraints associated to an activity should be defined as the conjunction of all the hard constraints associated to each of the users. Thus, for example, if a user requires the overall duration to be lower than 3 hrs., while another user requires an overall duration lower than 2 hrs., the hard constraint of the second user prevails.

7.2 Motion Execution Monitor

The motion plan is based on a high level plan produced by the Activity Planner. As previously described, this a priori plan takes into account the available knowledge of area to be travelled, with the various constraints and Points of Interest. It considers the physical parameters of the user and the preferences. This high level plan is split into elementary actions that are passed to the Motion Execution Monitor, which is a specific component of the Reactive Planner. This monitor is of low level and takes care of the various information obtained by the sensing system of the FriWalk, e.g. sensors. The gathered information is then merged and used to check if the elementary task can be safely executed or not. A typical case is the presence of a person along the walker's path that cannot be predicted at high level. For example, the Activity Planner knows that the current area is not crowded and thus is a good choice for the a priori plan, however a particular person on the path cannot be exactly predicted. The Motion Execution Monitor recognises from the sensing system the presence of such an unforeseen obstacle if it is in a relatively short horizon in front of the walker. A collision analysis is then performed, it is possible that the obstacle is crossing the walker's path but at a safe distance or at a safe time, in such a case no further action is needed. Otherwise an action of the Reactive planner is invoked. This can produce a simple slow down of the walker to let the obstacle pass, a complete stop of the walker or a call of a dynamic (local) re-planning. The easiest situation is to slow down the system without changing the path, a more involved case requires the design of a new local path that provides a dynamic obstacle avoidance, leaving for a while the current (optimal) high level path in favour of a suboptimal, but mandatory, deviation. The Reactive Planner takes care of such a local deviation and returns a path that reconnects to the high level plan after the obstacle has been avoided. In facts, the hypotheses are that the high level plan is optimal and the occasional local modifications of the speed or deviations are only suboptimal local perturbations of the original plan.

More in detail, the Motion Execution Monitor measures the probability of success of the task under analysis: the task is declared as not feasible if its success probability is below a certain threshold. In such a case the Reactive Planner is required to find a solution: the cheapest solution is to temporarily reduce the walker's speed. However, this is not always sufficient, i.e. the obstacle stands still on the walker's path and must be avoided changing the current path or the obstacle is moving against the walker causing a collision. The Motion Execution Monitor waits for a feasible solution of the Reactive Planner, and if there is one, it is analysed as any other elementary task and hence processed, otherwise, if the dynamic re-planning fails, a new high level plan is requested. The high level planner has thus to be informed of a new forbidden area and must produce a new global plan from the current position of the walker to its destination.

We consider two types of obstacles, static and dynamic ones. The presence of an unforeseen static obstacle will require a local dynamic re-planning. This works well in practise if the size of the obstacle is not too big with respect to the size of the walker. If it is not the case the Reactive Planner will declare a failure and a high level re-planning is required.

8 Example application

In this section we present a simple application of the techniques presented in section 7.

8.1 Activity Execution Monitor

To illustrate a simple use case scenario for the Activity Execution Monitor we simulate a "toy" activity, composed by a sequence of elementary actions, as shown in Figure 2.

Figure 2: Sequence of elementary actions composing a simple activity

To each elementary action composing the activity is associated a probability density function associated to the parameter representing the maximum amount of time required to complete it.
Moreover, a global constraint c_1 is associated to the activity, i.e. "the overall duration of the activity must be lower than $t_{\max}$", where $t_{\max} = 1200$ s.
The probability of respecting the constraint c_1 assuming that the action T_i have been started at time t_i, and is still executing at the current time t_{cur}, can be computed as:

$$p_{c_1} = P(T_{\text{end}} \leq t_{\max}) = P(T_i + T_{i+1} + T_{\dots} + T_n \leq t_{\max} - t_i | T_i > t_{\text{cur}} - t_i)$$

where T_{end} is a random variable representing the overall duration of the activity, while T_j is a random variable representing the duration of the action j.
At the beginning of the execution, when the monitor is invoked for the first time, the values are:

$$i = 1, t_i = 0, t_{\text{cur}} = 0$$

and the probability of respecting the constraint

$$p_{c_1} = 0.98$$

is above the threshold (= 0.9), thus, as expected, the activity is considered feasible.
Now assume that after 5 minutes, the user is still performing the first action of the activity:

$$i = 1, t_i = 0, t_{\text{cur}} = 300$$

and the probability of respecting the constraint

$$p_{c_1} = 0.64$$

now is below the threshold (= 0.9), thus the Activity Monitor considers the current activity as unfeasible, and invokes the Activity Planner to update the current plan and to render it feasible again (e.g. by removing some intermediate points of interest from the activity).

8.2 Motion Execution Monitor

A simple use case scenario for the low—level motion monitoring is shown in Figure 3. The walker is

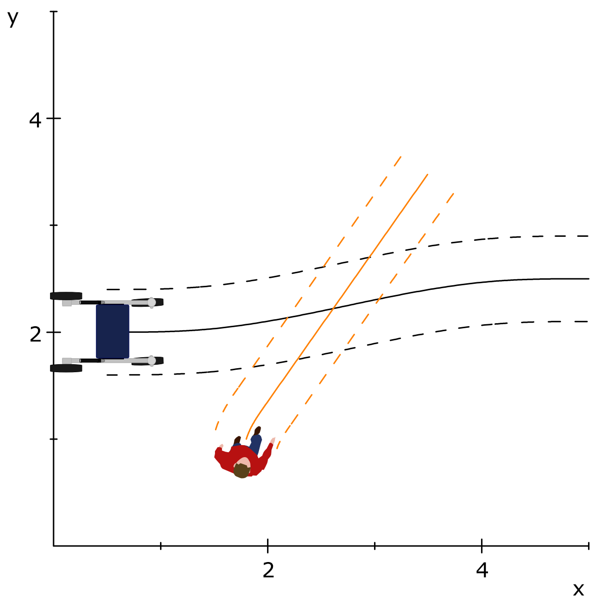


Figure 3: simple example of a pedestrian crossing the path of the FriWalk

moving along a planned path, while a pedestrian is approaching from the right. During the motion of the FriWalk, the Motion Execution Monitor is constantly invoked to check the feasibility of the current path. Whenever some people enter the field of view of the FriWalk, the monitor estimates possible trajectories based on their current position and heading. Then, the monitor determines the probability of collision taking into account also the estimated velocity of both the walker and the dynamic obstacles, and computing for each time instant the probability that a certain portion of the path will be occupied (see Figure 4).

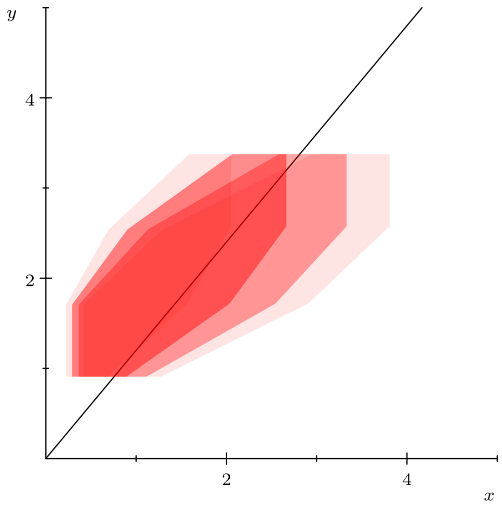


Figure 4: plot showing the probability that a certain portion of the path will be occupied over time (in red), and the trajectory of the Walker (in black).

Whenever the probability that the trajectory of the walker (the black line in the plot in Figure 4) will collide with some of the dynamic obstacles in the environment becomes greater than a certain threshold value, the Motion Execution Monitor will invoke the Reactive Planner to locally update the current trajectory as appropriate.

9 Links with other work packages

The technological aspects of this deliverable have close links with deliverables of WP2 and with the other deliverables in WP5. This deliverable specifically refers to D5.2 (User reactive planning), builds on D2.6 (Activity plans representation) and is delivered in parallel with D5.4 (Activity planning). The “social” requirements of the Activity Monitor (e.g., to monitor the preferences, and hard and soft constraints of participants) have been led by the case studies and user requirements identified by WP1.

The notion of a group is key to an activity and thus key to the Activity Monitor. In D5.2 we describe how the Reactive Planner suggests an optimal direction to each participant of an activity by “monitoring” the presence of static and dynamic obstacles. In contrast, the Activity Monitor monitors the feasibility of the “high-level” activity, considering specific activity-related metrics. In both cases performance is judged with respect to user preferences and an a priori optimal “global plan” for the whole group.

In D2.6 (Activity plans representation) we proposed the PDDL class of languages to represent an activity at a syntactic level and a graph based abstraction to represent the activity at a semantic level. These choices have strongly influenced how the Activity Monitor functions. In particular, PDDL decomposes motion planning into a sequence of control actions with pre- and post-conditions.

10 Conclusion

We have provided an investigation and demonstration of the challenges and algorithms for monitoring activity plans. Individuals have personal (selfish) requirements and constraints, while the desirable characteristic of a group activity are based on cooperation and communication.

Groups pose technical challenges in terms of both planning and monitoring. In contrast to solo activities, groups must be coordinated. To maximise efficiency, it is desirable to distribute the planning and monitoring tasks among the *FriWalk*s, while offloading as much as feasible to the cloud infrastructure. Reliable communication cannot always be guaranteed, hence the *FriWalk*s must be fail-safe and autonomous. In addition to some information not being available due to poor communication, the provided information may also contain (sensor) errors, so the overall system must be robust to measurement "noise" and partial information.

To address these issues, we have proposed that the Activity Monitor will be implemented on the *FriWalk* and make decisions based on a local copy of the current global state of the activity and its history. Local sensor information will be sent to the cloud, where it will be processed and distributed to other *FriWalk*s. The Activity Monitor on *FriWalks* thus makes its decisions based on the most reliable information it has available. A similar history is also maintained by the cloud, noting that the various histories stored across the system may not be identical. The principal role of the cloud is to "curate" the global data (e.g., resolve ambiguities and log it for the Activity Evaluator), however it will also have the ability to make decisions about the activity. E.g., it may wish to send a signal to the *FriWalk*s to terminate the activity.

Each time an activity plan is provided by the Activity Planner, it will be translated into a low level specification suitable for the Activity Monitor. We will then use a mixture of probabilistic reasoning and quantitative metrics to monitor the performance of an activity. Some metrics will provide feedback to the participants (e.g., slow down), some will trigger re-planning of the activity (e.g., obstructions), some will trigger a cancellation (e.g., multiple uncooperative behaviour) and some will just be logged for the Activity Evaluator, to provide feedback to carers and help improve the activity for the future.

In D2.6 (Activity plans representation) we identified the PDDL class of languages as a suitable means to represent activity plans at a syntactic level. These languages transform a motion planning problem into one of task planning, decomposing the problem into a series of goals and making it easier to deal with motion in continuous space. Variants of this language have incorporated notions of rewards, nondeterminism and probabilities, which are necessary for our application and align well with our chosen monitoring technology.

From the representation of the environment as a graph data structure we encode various metrics as weights assigned to the edges. For the purposes of planning, paths in the environment that optimise parameters of an activity can be found by finding the shortest weighted path in the abstract graph. The same weights can also be used to monitor the progress of the user with respect to an activity, and this gives great insight about actual paths and the global progress and feasibility of the plan.

Bibliography

[1] DALi Project, ICT-2011-288917: FP7/2007-2013, http://www.ict-dali.eu.

[2] B. Boyer, K. Corre, A. Legay and S. Sedwards, "PLASMA-lab: A flexible, distributable statistical model checking library," in *Quantitative Evaluation of Systems (QEST)*, 2013.

[3] M. Fox and D. Long, "PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains," *Journal of Artificial Intelligence,* vol. 20, pp. 61-124, 2003.

[4] "NDDL Reference manual," [Online]. Available: https://github.com/nasa/europa/wiki/NDDL-Reference. [Accessed 28 7 2016].

[5] D. E. Smith, J. Frank and W. Cushing, "The ANML language," in *ICAPS-08 Workshop on Knowledge Engineering for Planning and Scheduling (KEPS)*, 2008.

[6] A. Gerevini and D. Long, "Plan constraints and preferences in PDDL3: The Language of the Fifth International Planning Competition," Department of Electronics for Automation, Bresica University, Italy, 2005.

[7] H. L. Younes and M. L. Littman, "PPDDL1.0: An Extension to PDDL for Expressing Planning Domains with Probabilistic Effects," in *Proceedings of the 14th International Conference on Automated Planning and Scheduling*, 2004.

[8] "NuPDDL: nondeterminism and more in PDDL," [Online]. Available: http://mbp.fbk.eu/NuPDDL.html. [Accessed 28 7 2016].

[9] S. Sanner, "Relational Dynamic Influence Diagram Language (RDDL): Language Description," 2010. [Online]. Available: http://users.cecs.anu.edu.au/~ssanner/IPPC_2011/. [Accessed 28 7 2016].

[10] D. L. Kovacs, "A Multi-Agent Extension of PDDL3.1," in *Proceedings of the 3rd Workshop on the International Planning Competition*, 2012.

[11] A. Colombo, D. Fontanelli, A. Legay, L. Palopoli and S. Sedwards, "Efficient customisable dynamic motion planning for assistive robots in complex human environments," *Journal of ambient intelligence and smart environments,* vol. 7, no. 5, pp. 617-634, 2015.

[12] M. Guo and D. V. Dimarogonas, "Reconfiguration in motion planning of single-and multi-agent systems under infeasible local LTL specifications," in *Proc. IEEE Conference on Decision and Control*, 2013.

[13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research,* vol. 30, no. 7, pp. 846-894, 2011.

[14] "Java parser for the Planning Domain Definition Language," 2013. [Online]. Available: https://github.com/gerryai/pddl-parser. [Accessed 28 7 2016].

[15] "PDDL4J library," 2014. [Online]. Available: https://github.com/pellierd/pddl4j. [Accessed 28 7 2016].

[16] "pddl-lib," 2015. [Online]. Available: https://github.com/hfoffani/pddl-lib. [Accessed 28 7 2016].

[17] B. J. Grosz and S. Kraus, "Collaborative plans for complex group action," *Artificial Intelligence,* vol. 86, no. 2, pp. 269-357, 1996.

[18] J. Allen and G. Ferguson, "Human-machine collaborative planning," in *Proc. of the Third International NASA Workshop on Planning and Scheduling for Space*, 2002.

[19] S. Kraus, "Negotiation and cooperation in multi-agent environments," *Artificial intelligence,* vol. 94, no. 1, pp. 79-97, 1997.

[20] D. Jung and A. Zelinsky, "An architecture for distributed cooperative planning in a behaviour-based multi-robot system," *Robotics and Autonomous Systems,* vol. 26, no. 2, pp. 149-174, 1999.

[21] M. Bennewitz, W. Burgard and S. Thrun, “Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots,” *Robotics and autonomous systems,* vol. 41, no. 2, pp. 89-99, 2002.

[22] F. David, “Map-based navigation in mobile robots: I,” *A review of localization strategies: Cognitive Systems Research,* vol. 4, pp. 243-282, 2003.

[23] M. Mamei and F. Zambonelli, “Physical deployment of digital pheromones through RFID technology,” in *Proc. IEEE Swarm Intelligence Symposium (SIS)*, 2005.

[24] K. J. O'Hara and T. R. Balch, “Pervasive sensorless networks for cooperative multi-robot tasks,” in *Distributed Autonomous Robotic Systems 6*, Springer, 2007, pp. 305-314.

[25] S. Nouyan, A. Campo and M. Dorigo, “Path formation in a robot swarm,” *Swarm Intelligence,* vol. 2, no. 1, pp. 1-23, 2008.

[26] Y. Zhang, D.-w. Gong and J.-h. Zhang, “Robot path planning in uncertain environment using multi-objective particle swarm optimization,” *Neurocomputing,* vol. 103, pp. 172-185, 2013.

[27] H. Mo and L. Xu, “Research of biogeography particle swarm optimization for robot path planning,” *Neurocomputing,* vol. 148, pp. 91-99, 2015.

[28] Z. Beck, L. Teacy, A. Rogers and N. R. Jennings, “Online planning for collaborative search and rescue by heterogeneous robot teams,” in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 2016.

[29] M. Čáp, P. Novák, M. Seleck, J. Faigl and J. Vokffnek, “Asynchronous decentralized prioritized planning for coordination in multi-robot system,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[30] M. Čáp, P. Novák, A. Kleiner and M. Seleck, “Prioritized planning algorithms for trajectory coordination of multiple mobile robots,” *IEEE Transactions on Automation Science and Engineering,* vol. 12, no. 3, 2015.

[31] A. Kulkarni, M. Can and B. Hartmann, “Collaboratively crowdsourcing workflows with turkomatic,” in *Proc. of the ACM conference on Computer Supported Cooperative Work*, 2012.

[32] V. R. Desaraju and J. P. How, “Decentralized path planning for multi-agent teams with complex constraints,” *Autonomous Robots,* vol. 32, no. 4, pp. 385-403, 2012.

[33] E. Frazzoli, M. A. Dahleh and E. Feron, “Real-time motion planning for agile autonomous vehicles,” *Journal of Guidance, Control, and Dynamics,* vol. 25, no. 1, pp. 116-129, 2002.

[34] Y. Kuwata, J. Teo, S. Karaman, G. Fiore, E. Frazzoli and J. P. How, “Motion planning in complex environments using closed-loop prediction,” in *Proc. AIAA Guidance, Navigation, and Control Conf. and Exhibit*, 2008.

[35] S. Karaman and E. Frazzoli, “Incremental sampling-based algorithms for optimal motion planning,” *Robotics Science and Systems VI,* vol. 104, 2010.

[36] M. Vavrinec and M. Schaefer, “Prioritized planning for road vehicles coordination,” 2015.

[37] R. Finkel and J. Bentley, “Quad Trees: A data structure for retrieval on composite keys,” *Acta Informatica,* p. 1–9, 1974.

[38] M. Guo, K. H. Johansson and D. V. Dimarogonas, “Revising motion planning under linear temporal logic specifications in partially known workspaces,” in *Proc. IEEE International Conference on Robotics and Automation*, Karlsruhe, 2013.

[39] J. Dias, J. Rodrigues, N. Kumar and K. Saleem, “Cooperation strategies for vehicular delay-tolerant networks,” *IEEE Communications Magazine,* vol. 53, no. 12, pp. 88-94, 2015.

[40] C. J. Muise, V. Belle and S. A. McIlraith, “Computing Contingent Plans via Fully Observable Non-Deterministic Planning,” in *AAAI*, 2322-2329, 2014.