May 24, 2016

PLANNING PERSONAL ASSISTANT

Mihai Danila

PLANNING PERSONAL ASSISTANT

ABSTRACT

A planning personal assistant system and method is disclosed for producing plans for day-to-day activities. The system enables a user to delegate the process of planning the execution of tasks to a computing device. The method is provided through a software application executed in any computing device that can engage in assistive communication with a user. The system receives a collection of tasks and produces one or more plans based on user location and constraints associated with each of the tasks. The plans are produced by optimizing a parameter such as total time spent performing all the tasks, distance traveled, number of trips, or combinations of the above. The user then selects an appropriate plan for execution. The disclosed planning assistant can produce plans optimized along several dimensions in ways that can exceed the optimization abilities of human users.

BACKGROUND

Personal assistants available today perform basic assistance tasks generally of an on-demand nature. These systems are generally able to answer simple questions on demand, place calls, create simple reminders, and perform other simple actions. More recently, a popular application has developed the ability to trigger reminders based on the user’s location. Nonetheless, users still plan their tasks with limited or no support from their assistant. This can mean lost opportunities in the form of unnecessarily delayed tasks, suboptimal execution of tasks, and also tasks that were forgotten and never executed. Thus, a method for generating plans from a collection of tasks is required.

DESCRIPTION

A planning personal assistant system and method is disclosed for execution of tasks with direct applicability to a wide range of activities. The system applies the technique of producing executable plans to the domain of task execution with applicability to day-to-day activity planning, work planning, and beyond. It provides users the ability to delegate the process of planning the execution of tasks to a computing device. Figure 1 illustrates a method for generating a plan from a collection of tasks. The method can be provided through a software application executed in any computing device that can engage in assistive communication with a user. In versions requiring optimization of plans involving multiple users, the method could be implemented using a server version interacting with various user devices provided with corresponding software application versions. The application could also include a customized interface for interacting with the user.

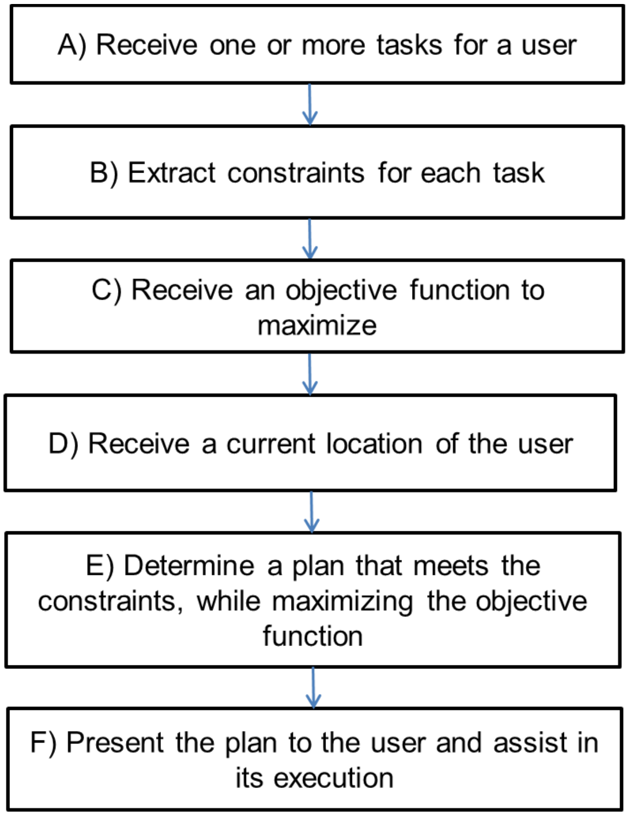
A) Receive one or more tasks for a user
B) Extract constraints for each task
C) Receive an objective function to maximize
D) Receive a current location of the user
E) Determine a plan that meets the constraints, while maximizing the objective function
F) Present the plan to the user and assist in its execution

FIG. 1: Method of a planning personal assistant

In step A the system receives a collection of tasks. The tasks could be obtained from the user's to-do list, or be inferred by the system. The user could prepare the to-do list in any suitable application, for example, a notepad, task reminder, etc. on an electronic device that is accessed by the system. The to-do list may be stored either locally in a memory of the electronic device or remotely in the Cloud or a server associated with the system. Tasks could also be inferred by the system from various sources, such as a phone text with an invitation to attend a meeting, a promotional e-mail message to purchase a product at a discount, or a voicemail message with a request to pick up the children from school.

In the next step B, the system procures constraints for each of the tasks. Any non-trivial planning problem implies that the tasks to be planned have some constraints associated with them. Constraints are an important input into the planning stage and could be of different types such as events that must occur at certain times or location constraints, as with buying items, which may be on sale only at particular locations. There could be attendee constraints, as is the case in meetings, or dependency constraints between tasks, as with the purchase of a perishable food item, which should not occur in the middle of a long travelling task even if the travel takes the user close to a store that sells that item. A typical constraint is the grouping of grocery tasks together rather than interspersed with other types of tasks. Tasks may have non-trivial combinations of constraints, for example, certain items must be purchased at specific stores, each having specific opening hours and the purchase can only be performed at times consistent with the user's own schedule. The system may also distinguish between strong constraints such as time constraints in a meeting from weak constraints such as grouping the purchase of grocery items. In some scenarios, the method may include collaborative planning in which the constraints may extend past a user's to-do list, and some tasks may support assignment to one of multiple other users.

In the next step C, the system receives a few predefined objective functions to optimize. The objective functions could be preset, or could be input by the user for specific task groups or instances of planning. Example functions to optimize include total time spent performing all the tasks, distance traveled, number of trips, or combinations of the above. Optimization need not be perfect, whereby a plan may be good enough to within some limit rather than being globally optimal. Instead of, or in addition to, receiving the objective functions, the system may elect to generate plans for a collection of typical objective functions and present all the resulting plans to the user for selection. The system may also annotate each plan with information useful for choosing between plans, such as time that it would take the user to execute the plan, number of discrete trips in the plan, maximum distance or trip duration, etc.

The system then receives the current location of the user in step D. The system may receive geolocation information of the user from various sources. For example, the system could receive global positioning system (GPS) coordinates from one or more electronic devices associated with the user. Additionally, or alternatively, the system could receive geolocation from location tagging in the user's recent social media posts or from user data, for example, predicted location based on receipts or travel information as indicated by e-mails or e-tickets.

Upon receiving the tasks, the constraints, and the location of the user, the system performs a planning step to produce a plan that respects all or most of the constraints in step E.

The planning problem could be solved using a variety of algorithms. For example, the problem could be modeled as a Constraint Satisfaction Problem (CSP) with discrete, continuous, and temporal constraints and solved by generic CSP algorithms, or can be modeled as a logic planning problem within first order temporal logic. The problem could in some implementations be modeled as a combinatorial or hybrid optimization problem, with a few aspects solvable algorithmically and other aspects solvable by a statistical technique such as neural networks, for example.

Having produced at least one plan, the system then displays the plan or choice of plans to the user for execution on a user's device in step F. If a choice of plans is provided, the user selects an appropriate plan for execution. The system may then further assist in executing the selected plan, for example, if the plan requires traveling to one or more locations, the system may display directions on a map annotated with the locations and times at which tasks in the plan can be executed. Tasks that are in the vicinity of the user's current location can be presented with prominence in the user interface, as an indicator that those tasks can be addressed first. The user may be allowed by the application to mark a task as either complete or deferred.

Other possible variations and specializations of the method could be as follows:

I. Re-planning

The assistant could allow the user to re-plan as required. For instance, if the plan requires a trip to a certain location for a single task, and the user removes the task from the task list, the assistant may update the plan to exclude a trip to that location altogether and re-optimize the plan. Furthermore, a portion of the remaining plan may be improved to make use of the slot of time now remaining unused. Additionally, if the user travels to a different location, the plan may change slightly to take into account the user's proximity to additional locations of interest for executing unlisted tasks. In some cases, the system may consider and present a simplified plan to the user to optimize computational cost. In some instances, the major planning component could run on a server and any plan updates could be run on a user's device. Furthermore, re-planning may require less computing power than full planning, especially when only small changes to the plan are expected.

II. Collaborative Planning

Multiple assistants may engage in collaborative planning. The collaborative planning problem is easily reduced to the original planning problem, where overlapping constraints are considered in a larger planning problem. For example, a meeting between a group of people may be planned, collaboratively, by their planning assistants, under the reasonable assumption that the assistants have access to their respective user's calendar. The system may even elicit a response from the multiple users, to a number of choices for the meeting time, asking the users to rate the choices, for example. The assistants could then share the results and present a tally to each user. As an alternative example, two or more members of a household may be in a position to borrow tasks from each other. Their assistants could, collaboratively produce an assignment of tasks that respects the schedules, locations, and future locations of both participants while maximizing some objective function.

III. Task Discovery

The system in some implementations could engage in task discovery, for example, if the user creates reminders for buying milk approximately once per week, the system could extrapolate a recurring task for buying milk and suggest it to the user.

The planning personal assistant system can be implemented for use in the Internet, an intranet, a user device, or another environment where the need to execute multiple tasks is a reality. The system can be implemented locally on a client device or implemented across a client device and server environment. The client device can be any electronic device such as a mobile device, smartphone, tablet, handheld electronic device, wearable device, laptop, etc.

The subject matter described in the disclosure can be implemented in software and/or hardware, for example, computers, circuits, or processors or can be implemented on a single device or across multiple devices, for example, a client device and a server device. Devices implementing the subject matter can be connected through a wired and/or wireless network. Such devices can receive inputs from a user, for example, using mouse, keyboard, or touchscreen and produce an output to a user, for example, through a display.