Combining Trip and Task Planning: How to get from A to *Passport*

Amin Abdalla and Andrew U. Frank

Vienna University of Technology
Institute for Geoinformation and Cartography
{abdalla,frank}@geoinfo.tuwien.ac.at

Abstract. Navigation-tools currently give us directions from location A to B. They help us with the physical process of moving from here to there. Tasks in general, are achieved by the subsequent determination and execution of sub-tasks until the goal is achieved. To help achieve the higher-ranking task, we commonly use so called "personal information management"-tools (PIM-tools). They offer possibilities to manage and organize information about errands that have personal or social implications. Such tasks are described in informal ways, todo-lists for example offer the storage of textual description of an errand, sometimes allowing geographic or temporal information to be added. The paper proposes a formalism that can produce instructions leading from A to the fulfilment of the "task". Thus connecting the high-level task, that represents intentions, with the physical level of navigation.

Keywords: PIM, task planning, routing, LBS, shortest path

1 Introduction

The problem of how humans find their way from A to B is an issue that puzzled researchers for many years and is investigated in various domains [24][26]. As a consequence we now see tools helping us to plan and execute navigation from one location to another. But these tools know little about the purpose of our trips. Although human wayfinding can, as Golledge and Gärling put it, be regarded as an *"...purposive, directed and motivated activity..."* [11] we have not seen attempts to integrate our intentions into GIS or navigation tools. The motivation behind our trips is an important factor that can help to improve the usability of such tools. Imagine asking your navigation device what the next task is and how to achieve it.

To handle and manage information about tasks or errands we usually use calendars or todo-lists, increasingly in digital form. These tools essentially store information about our intentions and motivations that imply sub-tasks such as navigation. The research field of managing and organizing personal information is referred to as "personal information management" (PIM) [16] or "task information management" [19]. While most of the studies are concerned about how to maintain digital information such as documents, pictures or webpages, efficiently; we like to focus on the task management part of it. That is how we retain information about our intentions and future activities and more specific, on how we plan to execute them.

In the following pages we will investigate the issue of integrating tasks into GIS respectively navigation-applications (or vice versa). By examining a specific problem we try to determine what information is needed to compute a suitable path through the search space (i.e.: the set of all possible states) of the problem.

Starting with an overview of relevant work, we will present and analyze the example of a "passport application". Finally we will propose a solution by narrowing it down to a shortest path problem, such that we can give instructions of how to get from a specific location and situation to the state that allows to apply for a passport.

2 Relevant Work

2.1 Personal Information Management

PIM activities can be viewed as "*an effort to establish, use and maintain a mapping between need and information*" [16]. *Need* in that sense depicts a necessity of a task (e.g.: a restaurant reservation before a meeting). To find out when we *need* to be at a meeting we consult a calendar. To find out what we *need* to buy we look at a todo-list. In this work we focus on calendars and todo-lists that often bear a very general and informal representation of tasks or errands. The only machine readable information currently supported are temporal intervals, due dates and to some extent locations. Temporal information is used to trigger alerts, or in some cases checked for overlaps of events. Spatial information is increasingly utilized for location based alerts on mobile devices. But current solutions are rather simplistic and as shown in [22,21,2] the integration of space and time bears more potential for supporting our daily life task planning.

2.2 Affordances and Places

The notion of *affordance* was shaped by Gibson [9], who investigated the perception of the environment. The core assumption is that objects or things are not primarily perceived by discrimination of their properties or qualities, as seen by orthodox psychology. He suggested that humans perceive their environment on the basis of its affordances, hence the possibilities of interaction. A horizontal surface, for example, affords support, what allows walking, as opposed to a steep slope that might afford slipping or falling. So the environment constrains the possibilities of what can be done. Jordan et al. [17] argued for an affordance based model of places, to improve the communication between the GIS and the user. They mention the work of Heft [14] who considers the role of functions or affordances of places in navigational processes. According to them an affordance based model of place is comprised of 6 aspects, listed in Table 1.

Table 1. The defining aspects of an affordance based place model are listed on the left side. The right column shows the implemented representations.

Defining Aspects	**Implementation**
physical features	location & object collection
actions	pick-up & locomotion
narrative	—
symbolic representations/Names	home,office,shop,etc..
socioeconomic and cultural factors	—
typologies/categorizations	container & street-node

In 2004 Raubal et al. [21] presented a comprehensive theory for location based services (LBS), in which they attempt to combine an *extended theory of affordances* with time geography [13]. It was achieved by embedding affordances into different realms: *physical*, *social-institutional*, and *mental* [20]. The integration of a social-institutional reality into the theory causes the physical affordance of taking a strangers purse, for example, to be suppressed by the context of institutional/social regulations (e.g.: law). Mental affordances are explained as the affordance to decide upon perceived physical or social-institutional affordances possible.

They set the framework for a LBS that can solve scheduling problems for a traveler with different time constraints and preferences. The article illustrates, besides other things, the need for an affordance based place description in order to be able to compute plans or schedules for an agent.

In our solution we adopt a simplified affordance model of place, dealing only with physical features (i.e.: collection of objects available), actions (i.e.: activities possible), categorizations and to some extent names (see Table 1).

2.3 Spatial behavior, decision making and problem solving

There are several fields that investigated spatial behavior and decision making extensively. A core issue in transport planning, for example, is the question of service demand. The dominant approaches for modeling it are (1) recording the spatial behavior or (2) the "*examination of the decision-making and choice processes that result in spatially manifested behavior*" [10]. These are referred to as behavioral and structural models. While structural models represent the aggregate movement activities of populations, behavioral approaches try to take the uniqueness of each individual into account. This lead to the investigation of wayfinding as well as cognitive mapping and its impact on spatial behavior. As research has shown, individuals build a cognitive map of their unique mental representation of the world [6]. This information can be facilitated to take decisions about movement in space [18]. This leads to a certain *behavior space* [?] in which our movements are located. Thus, our internal representation of the world has substantial impact on our movements and behavior.

Artificial intelligence is a field concerned with agent behavior, decision making and particular problem solving, but probably in a more formal way. There, a problem can be defined by five components [**?**]:

- states;
- initial state;
- transitions or actions;
- a goal test;
- path cost.

A solution then is an ordered action sequence that leads from the initial state to the goal state. A famous example for such a problem is the shortest path problem. It describes the question of how to get from one state to another with least effort, assuming a graph like structure. A well known algorithm to solve the problem is *Dijkstra's algorithm* [5]. We will show that by generalization we can narrow down the problem of getting a passport in the fastest possible way, to such a shortest path problem and solve it by the same means. The attempt is to translate the informal task and *behavior space* description into a formal problem description.

3 The Case Study

The scenario we chose to investigate is that of a person who plans to travel abroad and therefore needs a new passport since the old one is expired. In previous work [3] we investigated the planning process undertaken by the agent (see Figure 1). We define the task by two aspects: (1) spatio-temporal requirements, hence the physical presence of the person at the administrative office within the opening hours, and (2) a set of required objects, what we will call *equipment*.

In accordance with the assumptions mentioned in the previous section a *mental/cognitive map* [12,6] is utilized by the agent to map the tasks upon suitable places. For example: in order to acquire a picture the agent is aware of two shops that *afford* the task and are thus included in the decision making process.

The agent further puts the sub-tasks in order, by checking for dependencies. Since the task *photograph acquisition* has money as a precondition, it was derived that *money acquisition* needs to be conducted beforehand.

Finally the tasks are translated into a series of future actions (see bottom of Figure 1). In that part again a mental representation of the world is facilitated for the determination of travel times and routes.

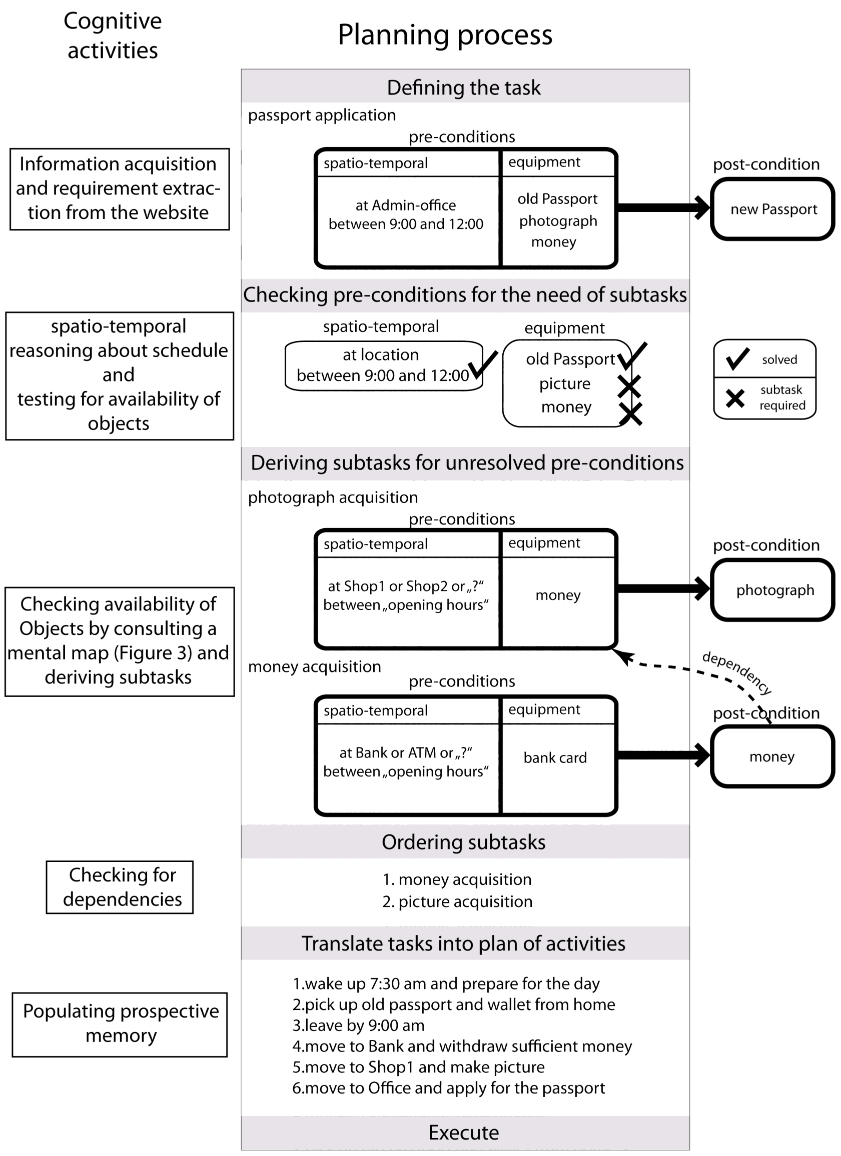
Cognitive activities
Planning process
Defining the task
passport application
pre-conditions
spatio-temporal
equipment
at Admin-office between 9:00 and 12:00
old Passport photograph money
post-condition
new Passport
Information acquisition and requirement extraction from the website
Checking pre-conditions for the need of subtasks
spatio-temporal
at location between 9:00 and 12:00
equipment
old Passport picture money
solved
subtask required
spatio-temporal reasoning about schedule and testing for availability of objects
Deriving subtasks for unresolved pre-conditions
photograph acquisition
pre-conditions
spatio-temporal
equipment
at Shop1 or Shop2 or „?“ between „opening hours“
money
post-condition
photograph
Checking availability of Objects by consulting a mental map (Figure 3) and deriving subtasks
money acquisition
pre-conditions
spatio-temporal
equipment
at Bank or ATM or „?“ between „opening hours“
bank card
dependency
post-condition
money
Ordering subtasks
1. money acquisition
2. picture acquisition
Checking for dependencies
Translate tasks into plan of activities
1.wake up 7:30 am and prepare for the day
2.pick up old passport and wallet from home
3.leave by 9:00 am
4.move to Bank and withdraw sufficient money
5.move to Shop1 and make picture
6.move to Office and apply for the passport
Populating prospective memory
Execute

Fig. 1. A simplified illustration of the planning process the person goes trough. On the left the person's cognitive activities are pointed out. In the middle the planning process is subdivided into certain steps. Outcomes for the defined tasks are found on the right side. [3]

3.1 Problem definition

Before elaborating on the proposed solution we provide a formal definition of the problem, according to the 5 components listed in section 2.3.

States: The problem definition uses a deterministic environment, hence each next state is determined by the transition (i.e.: the action the agent takes) that lead to it. The states in our search space are determined by a product value of the set of all locations L and the set E of all equipment states (i.e.: the objects carried). The state-set S is therefore defined as:

$$S \subseteq (L \times E)$$

S stands for all possible combinations of locations and equipment states, given the affordance constraints enforced by the environment. S can be seen as the formalization of a *behaviour space*.

Initial state: The initial state for our case study is *home* as location and an empty set of objects as equipment. We denote the start-state s as :

$s = (l, O)$ where $l \epsilon L$ and $O \subseteq E$
with
$l = home$
$O = \emptyset$

Transitions: Transitions or actions represent the rules of how a state can change from one to another. We consider two different kinds of actions (1) locomotion (e.g.: movement from one location to another) and (2) manipulation (e.g.: picking up an object). Each transition has a cost value c attached to it, in this example a value depicting the time an action takes in minutes. A transition is denoted as:

$t = ((l, O), (l, O), c)$ with $l \epsilon L$, $O \subseteq E$ and $c \epsilon \mathbb{R}$, c ≥ 0

Goal Test: The goal is achieved when a solution to the problem with the least cost is found. Thus an ordered sequence of transitions t that leads from the initial state s to the goal state g, with a minimum cost sum.

$g = (l, O)$ where $l \epsilon L$ and $O \subseteq E$
$l = office$
$O = \{oldpassport, photograph, money\}$

Path cost: In order to compute a final cost for a solution we need to be able to build a total sum over all the transitions in a path sequence, independent of the type (i.e.: manipulation or locomotion). For the example we defined the cost to be a temporal value, thus a certain amount of time spent on each action.

4 Solution

The problem definition above is already on a very generalized and abstracted level, but as we learn from figure 1 there are quite a number of cognitive activities involved to get from the general description of the task to a level of abstraction that allows to compute a plan. In the following we propose a formalism that helps building the state space and transitions defined in 3.1.1 and 3.1.3. We start with the definition of an algebraic model of the relevant environment, out of it we produce a state transition network representing the possible orders of object acquisition and finally combine the two into a single search space. Subsequently this allows us to run a well known problem solving algorithm (i.e.: Dijkstra's shortest path) to determine an optimal solution to the posed task.

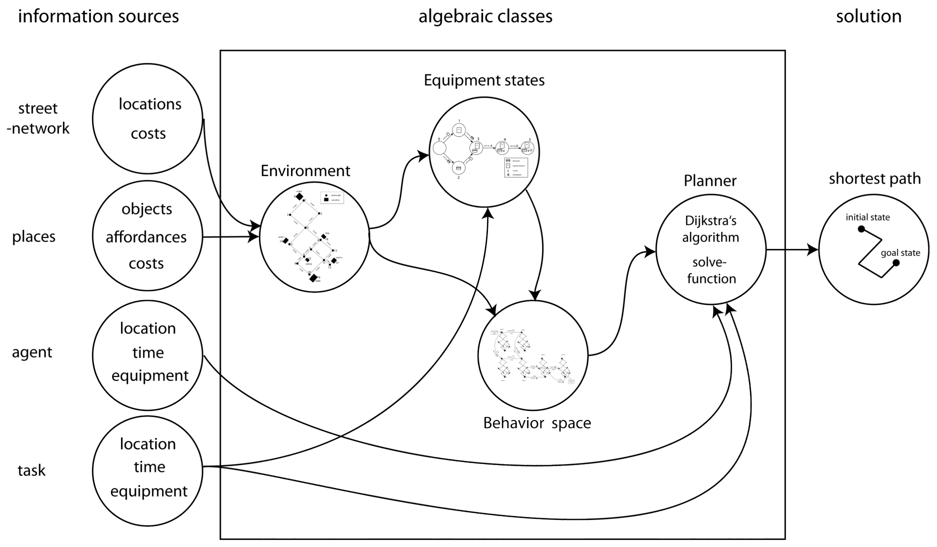

Fig. 2. A graphical illustration of the solution. The information sources on the left represent databases or data structures that form the basis for the algebraic classes.

The implementation was done in the functional programming language Haskell [25].

4.1 Representing the environment

In the first step we need a representation of the environment, that suffices the needs of an agent to solve the task. We therefore build an algebraic model of the environment sketched in figure 3. It was modeled as a collection of places that we can act upon, based on the affordances it offers and the relations between them. The underlying data structure utilized, is a graph that contains information about places, the objects contained, affordances and their locations. The implementation models some places as image schematic containers [15] and hence distinguish them from simple street junctions that are not necessarily perceived as such. A *container* offers a *pick-up* affordance for objects contained and a *locomotion* affordance to move to neighboring nodes, whereas street junctions offer locomotion only.

```
class Environment graph location object where

   affordance :: graph -> location -> [action]

   locations :: graph -> [location]

   adjacent :: graph -> location -> [location]

   travelcost :: graph -> location -> location -> TCost

   pickupcost :: graph -> Object -> location -> TCost

   locomotions :: graph -> [locomotion]

   nextLocation :: graph -> locomotion -> node

   queryObj :: graph -> Object -> [node]

   queryObjAt :: graph -> location -> requirements

   queryAllObj :: graph -> [Object]
```

The function *affordance* returns a list of activities afforded by a place at a specific location. The second function returns a list of all locations that are accessible to the agent. The *adjacent* function returns all locations that are immediately accessible to the agent by a single *locomotion*, or simply the neighboring nodes for a given node.

To acquire the cost that is assigned to the *locomotion* from one location to another the *travelcost* function is implemented. The *pickupcost* returns the cost that a pick-up action applied to a certain object at a specific location takes.

The *locomotions* function returns a list of all possible transitions/locomotions that can be conducted in the environment. To determine the next location of the agent given a *locomotion* the *nextLocation* function is defined.

The final three functions are concerned with the objects at different locations. While the first returns a list of locations where a specific object can be found, the second returns the preconditions a *pick-up object* affordance exhibits. The requirements come in the form of *equipment*.

The last function simply returns all the objects that can be *picked-up* in the environment.

By having the environment represented in such a way we can in the next step extract a representation of the equipment states, taking the preconditions of *pick-up* object affordances into account.

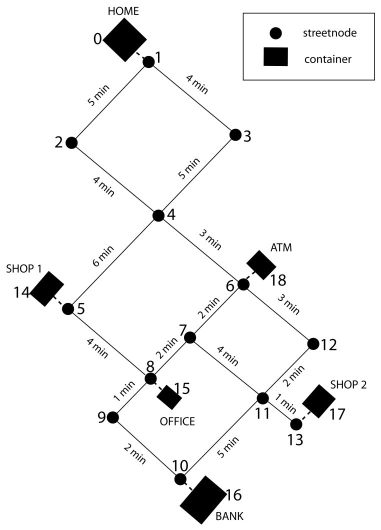

Fig. 3. The environment we consider consists of a street network and places that are linked to it. The places offer a pick-up affordance upon a set of objects. Each locomotion from one streetnode to another is uniquely defined by its start- and endnode and has a cost value attached to it.

4.2 Modeling the requirements

Now we can start building a finite and deterministic state-machine that models the possible equipment-states. Such a state machine in general is defined by a set of states S[1] , a transition function δ and an alphabet A. S is defined as:
$S \subseteq \mathcal{P}(O)$ with O being the set of required objects. Because we only consider directed edges, the subset S can be computed from $\mathcal{P}(O)$ by checking it for *consistency*. It means that an object collection cannot contain money without a bankcard for example, since the bankcard is a precondition to the *pick-up money* affordance. This helps to keep the example small and clear.
The transition function is defined as $\delta : O \times S \to S$, hence takes an object and a state and returns the next state.
We implemented the state machine as a class (see figure 4 for an illustration):

```
class ObjectAutomata state transition where
   makeStateSet :: Equipment -> [state]
   pickup :: Object -> state -> state
   makeTransitions :: [state] -> Equipment -> [transition]
   possibleTransition :: state -> [transition]
   nextState :: state -> transition -> state
```

[1] Note that the S here represents the equipment-state and thus differs to the S introduced in the problem definition

The first function creates the state set S, taking a set of required objects as an input. The pickup-function denotes the transition function δ by taking an object and a state as an input and returning the resulting state.
The subsequent two functions return all the possible transitions/actions (*make-Transitions*) and all the transitions possible from a specific state (*possibleTransition*).

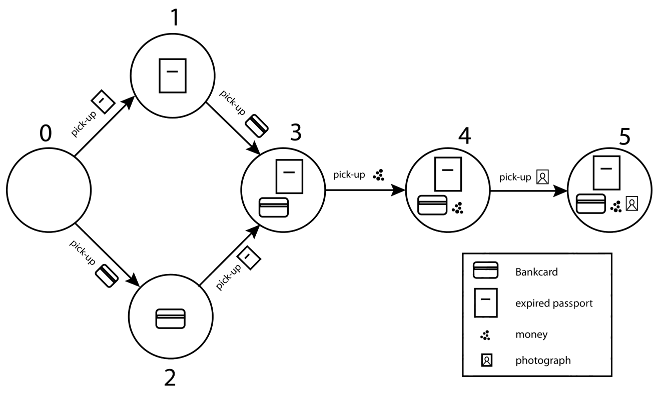

Fig. 4. By using only a single action, we can simplify the graph that depicts the state machine. Each transition is defined by its start- and endstate.

The final function takes a state description and a transition to return the next state. Having a representation of the object order, allows to proceed to the step of combination.

4.3 Combining environment and task requirements

To reach the state space described in section 3.1.1 we need to combine the environmental representation with that of the object requirements. Therefore a methodology suggested by Frank [7] is used. The work describes a mathematical formalism to merge two state-transition networks, based on category theory [4]. The combined state is defined by a pair of the individual location states and equipment states. The combined actions are simply the sum of both costs. To take environmental constraints into account, combination rules are introduced. These define the possible transition from one product-state to another. In our case these rules are determined by the locations that offer *pick-up* affordances for the required objects. Figure 5 gives an illustration of how the combined graph looks like. Again we represent the combined network as a class:

```
class BehaviorSpace combination state where
   states :: combination -> [state]
   transitions :: combination -> [(state,state,TCost)]
   nextTrans :: combination -> state -> [state]
   cost :: [(state,state,TCost)] -> state -> state -> TCost
```

The *combination* input of the first three functions is defined as a vector containing the minimum information needed to build the two different representations (environment and equipment states) described in 4.1 and 4.2. These are on the one hand a data structure or database that contains the information about the environment and on the other the set of objects required for the task.

The functions defined to describe the *BehaviorSpace* class are conceptually the same as some of the functions defined for the initial environment. Derive able is a set of states, a set of all possible transitions, a set of all the possible transitions based on a specified state and the cost for a given transition.

4.4 Problem Solving

In the final step we created a planning class, that mimics some of the cognitive activities shown in figure 1. The class is defined as follows:

```
class Planner behaviourspace state where
   stp :: behaviourspace -> state -> state -> [(Action,Cost)]
   solve :: behaviourspace -> Agent -> Task -> [(Action,Cost)]
```

The *stp* function computes the shortest path from the current state of an agent to the goal state defined in section 3.1.4, by running a *Dijkstra's algorithm*. To supplement the model we defined an agent and a task description that is used as an input for the problem solving function *solve*:

```
   data Agent = Ag Location Time Equipment
   data Task = T (Location,TimeInterval) Requirements
   agent = Ag home (08,00) []
   goal = T (office,((09,00),(12,00))) [passPhoto,money,oldPassport]
```

Note that the above definitions include time. We did not take temporal aspects into account, although we believe including it to some extent is straightforward. The final function *solve* takes, besides the agent and the task description, the computed *behavior space* and the required *equipment* as an input. The function then compares the agent's current state to the goal state defined in the task. It determines the set of missing objects and passes it to the *stp* function that builds the requirement state transition graph "(figure 4), extract the combination rules and combine it with the street network graph. The result is an optimal solution in form of a series of actions, locomotions and manipulations, along with the time it takes to conduct it.

5 Discussion and Outlook

The paper presents the attempt to merge information stored in a PIM-tool (i.e.: calendars or todo-lists) with a typical functionality found in GIS or navigation

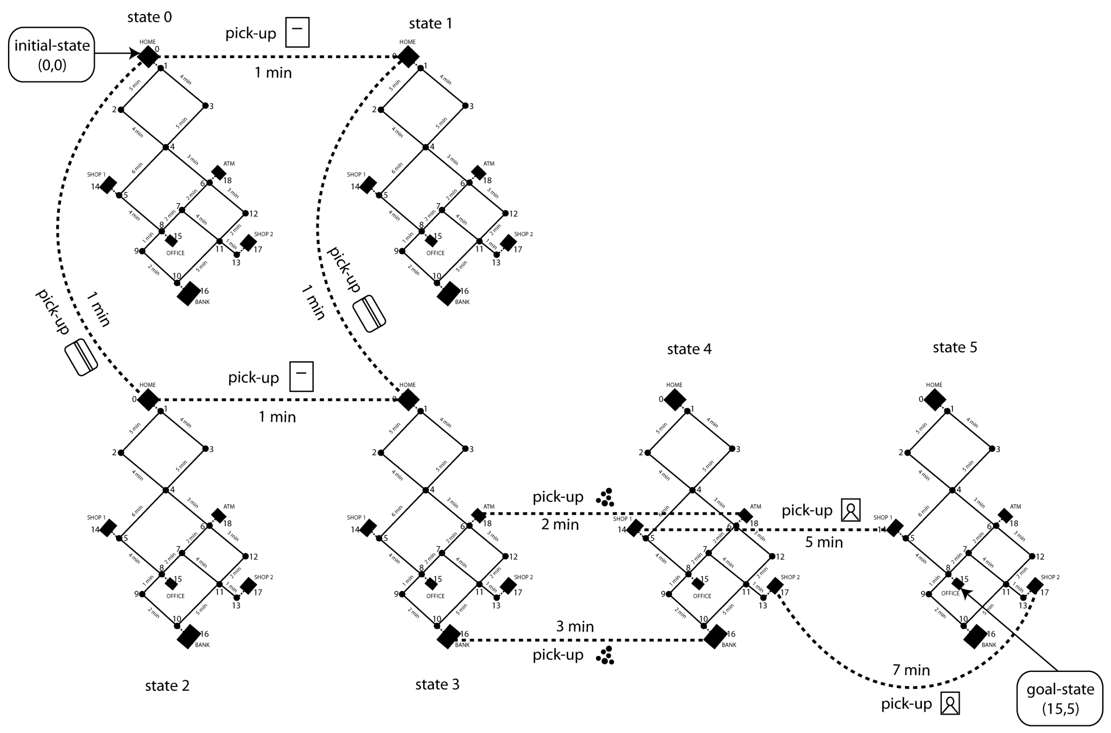

Fig. 5. A combined state is defined as a pair of an individual location state and an equipment state. To switch into the next equipment state a manipulation transition is needed (dotted lines), to move from a location to another a locomotion is required (street edges). The combination rules depict where a manipulation can take place.

applications, i.e.: routing. By using a simple example we investigated a formalism that translates it into a*shortest path problem*. Using mathematical category theory, two semantically different state transition networks were combined into a single one and a plan that represents spatial behavior was computed.

The work highlighted some of the issues that arise when trying to combine navigation with ordinary tasks. Foremost the need of an affordance based place model that integrates small scale objects. A GIS aiming at personal tasks, has to offer a more user centric representation of the environment. For our daily tasks the geometric or map metaphoric information is less important than activities, objects and people. Therefore we introduced the notion of *equipment*, that we believe plays a prominent role in our daily life. Not having a photograph when intending to apply for a passport results in failure.

Another serious issue is the question of complexity. We simplified the model by including two affordances: *pick-up* and *locomotion*. But there are infinitely more actions possible. Thus: what level of granularity is necessary for such an application? Do we need to consider actions like ”pay” or ”open door” that take place inside a shop?

In a previous paper [1] we envisioned PIM-tools that act pro-actively by alerting us in situations that threaten the success of a task. Therefore we would not just need an optimal solution, but a representation of all possible solutions, enabling it to determine when engagement is necessary. Further the question of how to handle several tasks needs to be investigated.

Further the computation of paths based on other *costs* are possible, hence *cheapest* rather than *fastest* (e.g.: Find the cheapest path for my shopping list!).

The study can be seen as a first step towards the merging of PIM-tools and GIS. Hopefully in future we can solve some of the hindrances discussed, leading to task aware and more intelligent calendars.

References

1. Abdalla, A. and Frank, A. U. (2011). Personal Geographic Information Management. In Proceedings of the Workshop on Cognitive Engineering for Mobile GIS, Belfast, USA,. CEUR Workshop Proceedings.
2. Abdalla, A. (2012). Latyourlife: A Geo-Temporal Task Planning Application. In Advances in Location-Based Services, Lecture Notes in Geoinformation and Cartography, (pp 305-325). Springer Berlin Heidelberg
3. Abdalla, A. and Frank, A. U. (2012). Towards a spatialization of PIM-tools. In GIZeitgeist 2012: Proceedings of the Young Researchers Forum on Geographic Information Science, Muenster. ifgiPrints.
4. Asperti, A. and Longo, G. (1991). Categories, Types and Structures - An Introduction to Category Theory for the Working Computer Scientist. Foundations of Computing. The MIT Press, 1 edition.
5. Dijkstra, E. W. (1959). A note on two problems in connection with graphs. Numerische Mathematik, Volume 1, pages 269-271.
6. Downs, R., Stea, D., et al. (1977). Maps in minds: Reflections on cognitive mapping. Harper & Row New York.

7. Frank, A. U. (2008). Shortest path in a multi-modal transportation network: Agent simulation in a product of two state-transition networks. KI Künstliche Intelligenz, Volume 3, pages 14-18.
8. Gärling, T. and Golledge, R. (2000). Cognitive mapping and spatial decision-making. Cognitive Mapping: Past, Present, and Future. Routledge, London, page 47.
9. Gibson, J. (1979). The Ecological Approach to Visual Perception. Erlbaum. Books.
10. Golledge, R.G., and T. Grling. (2003). Spatial behavior in transportation modeling and planning. In K.G. Goulias (Ed.), Transportation Systems Planning: Methods and Applications (pp. 3/1), CRC Press, New York.
11. Golledge, R. and Gärling, T. (2004). Cognitive maps and urban travel. Handbook of transport geography and spatial systems, Volume 5, pages 501-512.
12. Gould, P. and White, R. (1986). Mental Maps. Allen & Unwin. Sabine.
13. Hägerstrand, T. (1970). What about people in regional science? Papers of the Regional Science Association,Volume 24, pages 7-21.
14. Heft, H. (1996). The ecological approach to navigation: A gibsonian perspective. The construction of cognitive maps, pages 105-132.
15. Johnson, M. (1987). The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason. University of Chicago Press. Source: David Mark.
16. Jones, W. and Teevan, J. (2007). Personal information management PIM. (pp. 14), Univ. of Washington Press.
17. Jordan, T., Raubal, M., Gartrell, B., and Egenhofer, M. (1998). An affordance-based model of place in gis. In 8th Int. Symposium on Spatial Data Handling, SDH, Volume 98, pages 98-109.
18. Kuipers, B. (1978). Modeling spatial knowledge. Cognitive Science, Volume 2(2), pages 129-154.
19. Lepouras, G., Dix, A., Katifori, A., Catarci, T., Habegger, B., Poggi, A., and Ioannidis, Y. (2006). Ontopim: From personal information management to task information management. Personal Information Management: Now That We are Talking, What Are We Learning?. page 78.
20. Raubal, M. (2001). Ontology and epistemology for agent-based wayfinding simu- lation. International Journal of Geographical Information Science, Volume 15(7), pages 653-665.
21. Raubal, M., Miller, H., and Bridwell, S. (2004). User-centred time geography for location-based services. Geografiska Annaler: Series B, Human Geography, Volume 86(4), pages 245-265.
22. Raubal, M., Winter, S., Temann, S., and Gaisbauer, C. (2007). Time geography for ad-hoc shared-ride trip planning in mobile geosensor networks. ISPRS Journal of Photogrammetry and Remote Sensing, Volume 62(5), pages 366-381.
23. Russell, S. and Norvig, P. (2010). Artificial intelligence: a modern approach. (pp. 3/66), Prentice hall.
24. Stern, E. and Portugali, J. (1999). Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes, chapter Environmental Cognition and Decision Making in Urban Navigation, (pp. 99-118). John Hopkins Press.
25. Thompson, S. (1996). Haskell - The Craft of Functional Programming. International Computer Science Series. Addison-Wesley.
26. Timpf, S., Volta, G., Pollock, D., and Egenhofer, M. (1992). A conceptual model of wayfinding using multiple levels of abstraction. Theories and methods of spatio-temporal reasoning in geographic space, pages 348-367.