Space, Time, and Choice:
A Unified Approach to Flexible Personal Scheduling

Vicky Bilbily
University of Toronto
vicky@cs.toronto.edu

Elaine Huynh
University of Toronto
elaine@cs.toronto.edu

Karan Singh
University of Toronto
karan@dgp.toronto.edu

Fanny Chevalier
University of Toronto
fanny@dgp.toronto.edu

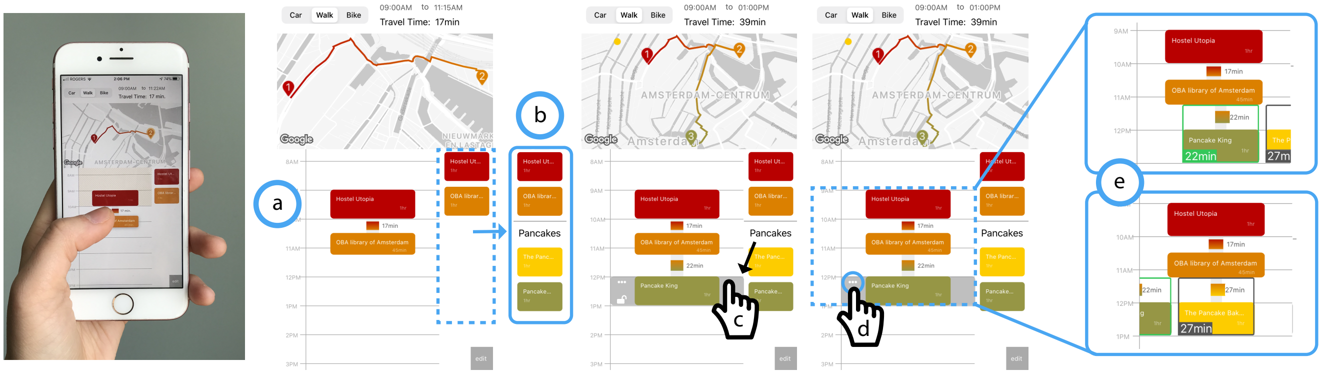

Figure 1: Space-Time Planner enables users to interactively (re-)specify fixed and semi-flexible space-time constraints in a fluid unified workflow when planning an itinerary. In our proof-of-concept prototype, the main view (a) includes an interactive schedule for connecting locations to temporal constraints, visualizing the time spent at each location and travel time between them; a synchronized map view; and a place palette with user-curated locations of interest that are used to build the schedule. (b) *Requirement groups* may be added to the palette, e.g., a collection of possible places for lunch, which help to maintain flexibility and automate optimizations. In this example (c), a *one-of* group is added, from which the system automatically selects one location option in the group that optimizes a given criteria (e.g. travel time, distance traveled). The location alternatives are available to be browsed in-situ by the user, if their preferences differ from the proposed optimization (d, e).

ABSTRACT

In the context of increasingly busy lives and mobility constraints, we present a unified space-time approach to support flexible personal scheduling. We distill an analysis of the design requirements of interactive space-time scheduling into a single coherent workflow where users can manipulate a rich vocabulary of spatio-temporal parameters, and plan/explore itineraries that satisfy or optimize the resulting space-time constraints. We demonstrate our approach using a proof-of-concept mobile application that enables exploration of the inter-connected continuum between task scheduling (temporal), and multi-destination route mapping (spatial). We evaluate the application with a user study involving an itinerary reproduction task and a free-form planning task. We also provide usage scenarios illustrating the potential of our approach in various contexts and tasks. Results suggest that our approach fills an important gap between route mapping and calendar scheduling, suggesting a new research direction in personal planning interface design.

CCS CONCEPTS

• **Human-centered computing → User interface design**.

KEYWORDS

space-time planning; scheduling; maps; mobile user interface

ACM ISBN 978-1-4503-8635-7/21/10…$15.00
https://doi.org/10.1145/3472749.3474764

ACM Reference Format:
Vicky Bilbily, Elaine Huynh, Karan Singh, and Fanny Chevalier. 2021. Space, Time, and Choice: A Unified Approach to Flexible Personal Scheduling. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21), October 10–14, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3472749.3474764

1 INTRODUCTION

It is noon and mid-way through a "busy" day: four meetings done, teaching at 2PM, another meeting at 4, dinner and movie-date at 7; in-between, you hope to post a letter, buy some wine, grab lunch, and squeeze in a 2km run. It's doable, you planned it all out!
A friend calls. They are in town at the museum, and want to meet for lunch. You could meet them at the noodle shop near the museum, but that is nowhere near a post office, nor a wine store. The bistro nearby does offer wine, but the service is slow, and getting back by 2PM will be tight.
Two messages roll in at lunch—one person wants to pick up a book, and the other would like to reschedule a meeting. Pushing back your 4PM will give you time to drop off the book and buy wine, but what about the run, the letter, and dinner before the movie? It's doable, but perhaps not all of it, and requires a change in plans!

The interwoven space-time permutations of our lives are full of personal preferences and constraints, route planning to multiple destinations, and temporal schedules to consider. At the two ends of the space-time spectrum, we efficiently navigate space with digital maps, and schedule our days by blocks of time in digital calendars. There is, however, less computational support for the intersection between space and time in our lives: the scheduling and *exploration* of spatio-temporal itineraries.

In complex scenarios, we usually pick the first option that comes to mind: a conservative sub-optimal plan, or worse–an unrealistic one that is doomed to disappoint. When we do consider our options more thoroughly, we find ourselves inefficiently ping-ponging between map, calendar and web queries: how long to get from here to there? There to here? What time does this place close? When do I need to be back? There may be unforeseen circumstances, and desires and constraints change: any plan needs to be able to adapt as it unfolds. We thus present an analysis of the personal space-time planning domain (Figure 2), and a personal space-time planner prototype that allows users to interactively create, choose, and adapt space-time schedules (Figure 1).

Prior art in space-time planning is primarily focused on the ends of the space-time spectrum, optimizing travel routes (space) or task schedules (time), from an a priori set of user-specified destinations and times. However, personal space-time planning is an ongoing dialogue between the digital planning tool and its user. Users provide the tool with goals and constraints, and then interactively choose between admissible plans generated by the tool to produce an optimal itinerary that the planning tool monitors, notifying and engaging the user as and when the plan needs to be revised (Figure 2).

We thus begin by analyzing a variety of personal space-time planning scenarios, in which we discover a need for greater flexibility and differing user interactions than those addressed in existing space-time frameworks in HCI [4], social sciences [35], transportation [36] and urban planning [1]. We first formulate and propose a unified space-time workflow to capture these scenarios (§2), and then position prior art relative to this workflow (§3). We note the paucity of research on interfaces enabling user exploration in space-time planning systems via the support of **user-motivated** actions, and design a novel prototype mobile application (Figure 1) that combines direct manipulation, enabling rich and flexible specification of space-time constraints, together with spatial and temporal representations, providing integrated support for both spatial and temporal reasoning, while exploring viable schedule alternatives satisfying the constraints (§4).

We validate our approach in two ways. First, we conduct an evaluation of the prototype implementing core functionality of our approach with a user study involving an itinerary reproduction task and a free-form planning task (§5), and find promising results suggesting that our approach is a worthwhile step in the direction of interface design for personal planning tools. We also present usage scenarios, illustrating the potential of our approach in supporting a variety of spatio-temporal scheduling tasks (§6).

Our contributions are:

(1) a unified space-time workflow with accompanying discussion of the space-time planning problem domain from a personal planning standpoint (§2), that allows us to identify gaps in the literature and opportunities for improved support of user-motivated actions (§3).
(2) a general approach to the problem that addresses flexibility of defining and re-defining space-time constraints for constructing and evolving a schedule, fluidly exploring alternatives, as well as supporting integrated spatial and temporal reasoning (§4.1).
(3) a prototype implementing a core subset of features from our described space-time workflow (§4.2), as well as its evaluation through a study investigating the initial reactions to our prototype and to the foundations of our approach (§5).
(4) additional usage scenarios, showcasing how our approach can be extended to support planning in a dynamic yet flexible context, as well as over multiple days (§6).

As a formative paper on flexible space-time scheduling, we begin the discussion by providing first-step solutions, shedding light on many non-trivial design and user-interaction questions. The goal of our work is to expose as many interesting problems in the space as we can, and to demonstrate the overall potential of our framework.

2 SPACE-TIME PLANNING WORKFLOW

The characterization of humans in space and time has been explored across geography [32], urban planning [1], economics, and other social studies. While an exhaustive review is beyond the scope of this paper, of note among these disciplines is the time-geography framework [35], central to which is the idea that time and space are inseparable, and that a plethora of physical and societal constraints dictate the space and time, and consequently the activities, available to humans as individuals. Such a framework has been used as the basis to observe, understand, and influence human behaviour, in the context of these externally imposed restrictions. Complementary models focus attention inwards, describing human activity as it is shaped by personal values and motives, which result in choices [7]. We draw inspiration from these frameworks in developing a workflow to assist individuals in holistically describing and planning their own space-time behaviour.

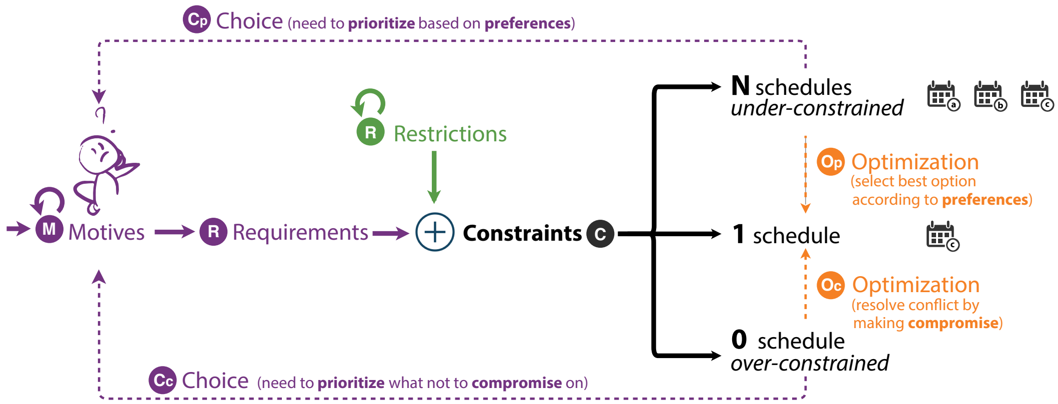

Figure 2: The iterative space-time planning workflow. Aspects that originate from the user are indicated in purple, such as user motives (M) which translate into the requirements (R). External factors include restrictions (R) that combined with user requirements (R) form overall space-time constraints (C), which may result in 0, 1, or multiple viable schedules. The process of determining one solution is iterative as it requires making choices to prioritize requirements, either based on preferences (Cp), or by making compromises (Cc). Computational approaches aiming at optimizing certain criteria ((Op), (Oc)) can provide further support to identify alternatives, while maintaining user agency over the different options that present to them.

The mobility agenda model [44] addresses a similar goal, albeit motivated by automation [38] and public transit. The authors identify spatial and temporal information needs, a notion of spatial flexibility, and the influence of user- and externally-originating factors on creating dynamism in a mobility agenda. Our work dives deeper into these notions and connects them further by defining a holistic workflow, focusing on interaction mechanisms to support user decision-making in dynamic contexts, via a mix of automation and user exploration.

We build a model that facilitates the design thinking for a user interface (UI) supporting personal planning. We describe the inputs and outcomes of the personal space-time planning process, and distinguish between factors that are **user-motivated or user-defined**, **external factors** impacting scheduling that are out of the user's control (eg. business hours of venue or traffic flow), and **system** optimization objectives (eg. reduce travel time, find the shortest route, or maximize time at a venue). We present these in a single, unified workflow, illustrated in Figure 2.

2.1 Space-Time Constraints

Scheduling an event is an attempt to satisfy some **motive** (Figure 2-(M)) by (1) allocating adequate time, and (2) determining an appropriate environment to fulfill the motive. These criteria can be expressed in terms of spatial, temporal or spatio-temporal **requirements** (Figure 2-(R)). Consider the simplistic example of Asa, who after class, wants to swim for about an hour. The temporal requirements include an approximate duration (one hour) and time constraints, both hard (after class at 5pm) and soft (before dinner). The spatial requirements are a venue with a swimming pool.

Spatio-temporal **restrictions** (Figure 2-(R)) act on top of these requirements, further constraining the potential space-time options. Whereas requirements can change with user motives, restrictions are external constraints (eg. a pool that closes at 5:30pm does not allow an hour for swimming). These spatio-temporal requirements and restrictions form the set of **constraints** (Figure 2-(C)) to be accounted for when planning.

Constraints can also be classified by their **abstraction level**. So far we have discussed **event-level** constraints, which are isolated from other potential events in a schedule. **Sequence-level** constraints result from the emergent interaction between events. Table 1 provides an example list of spatio-temporal constraints and their characteristics. We illustrate how these constraints can characterize real-life scenarios using a collection of simple examples in Table 2. Returning to Asa's scenario, a sequence-level *requirement* could be location precedence, i.e., she must go home to pick up swim gear before going to the pool. A sequence-level *restriction* would be the travel time from class to home to the pool, which may eliminate pool options that violate the temporal requirements.

2.2 Choice

Constrained space-time problems for automation typically produce a single optimal solution (Figure 2: 1 schedule). User objectives for personal scheduling however, can be both under-constrained and approximate, making the preferred solution among multiple viable solutions, subject to user choice (Figure 2: *N* schedules). Equally, complex over-constrained schedules may have no viable solution (Figure 2: 0 schedule), and require user choice in relaxing one or more requirements to produce a solution. A major contributing factor to the size of the solution space is the level of **flexibility** in these requirements. Back to Asa, she has no preference of pool, as long as it meets the other constraints. Table 3 shows how varying levels of flexibility present themselves temporally and spatially.

Users rarely provide perfectly constrained requirements making **choice** an intrinsic part of personal space-time planning. In over-constrained scenarios , users must interactively prioritize their motives, requirements, and flexibility, by making **compromises** (Figure 2-(Cc)), such as loosening a time constraint, adding location options, or removing an event from the schedule altogether. Automated **optimization** guidance can help a user choose on what to

Table 1: Spatio-temporal constraints and their characteristics

Constraint C	Abstraction Level	Source R/R	✥ Spatial/ 🕓 Temporal
DUR: Duration	Event	Requirement	🕓
SET: Start/end time	Event	Requirement	🕓
BND: Time boundary	Event	Requirement	🕓
ENV: Environmental/aesthetic quality	Event	Requirement	✥
FCT: Functional property	Event	Requirement	✥
HOP: Hours of operation	Event	Restriction	✥🕓
PRG: Programme	Event	Restriction	✥🕓
PRE: Precedence	Sequence	Requirement	✥🕓
ROT: Route property	Sequence	Requirement	✥🕓
TVL: Travel time / traffic	Sequence	Restriction	✥🕓
SCH: Transit schedule	Sequence	Restriction	✥🕓

Table 2: Example scenarios and corresponding constraints (see Table 1)

Example scenario	Constraints
Go swimming after class	**DUR**, **BND**, **FCT**, **HOP**
Planning movie out with friends	**DUR**, **SET**, **FCT**, **PRG**, **TVL**, **SCH**
Picking-up son and drop him at soccer	**SET**,**FCT**,**PRE**,**TVL**
Buy brushes, eat, before painting class	**SET**, **FCT**, **PRE**, **HOP**, **TVL**, **SCH**
Fit in errands in a busy day of meetings	**BND**, **FCT**, **HOP**, **TVL**
Organize a tour of a city	**BND**, **ENV**, **FCT**, **HOP**, **ROT**, **TVL**
Design a scavenger hunt	**SET**, **BND**, **ENV**, **FCT**, **PRE**, **TVL**
Plan a road trip for family	**DUR**, **ENV**, **FCT**, **HOP**, **ROT**, **TVL**

Table 3: Level of flexibility of spatial and temporal requirements

	Fixed	Semi-flexible
✥ Spatial Requirement	Exact location	Set of locations, defined by environmental and aesthetic qualities (**ENV**), functional qualities (**FUN**), and/or user-curated subset.
🕓 Temporal Requirement	Exact start and end time (**SET**), and thus exact duration (**DUR**)	Any combination of fixed and flexible constraints, including start before / end by (**SET**, **BND**), and min / max duration (**DUR**)

compromise (Figure 2-Oc). In under-constrained scenarios, users must choose between equally viable **alternatives**. The system can automatically **optimize** the solution relative to preset **preferences** (Figure 2-Op), such as total travel time, greatest accessibility, quietest space, or lowest carbon emission. In all scenarios automation should play a supporting role to user agency [42], and ongoing **choices** can be used to re-define user **preferences** (Figure 2-Cp).

When required to make a choice, users may discover other personal preferences that were not considered before [34]. For example, Asa may pick one viable pool over another because it is next door to her favorite ice-cream shop (adding a spatial requirement), a preference she would not have considered, had she not been prompted by the situation.

2.3 The Iterative Workflow

We thus define an iterative space-time planning workflow, demonstrated in Figure 2, that combines user interactive spatio-temporal constraints and choice, with constrained optimization. The user begins with some **motive** (Figure 2-M), expressed as spatio-temporal **requirements** (Figure 2-R), as initial input into the planning process. External **restrictions** (Figure 2-R), are applied on top of the requirements. The interaction between requirements and restrictions form zero or more schedules, which the user refines by specifying **preferences** or making **compromises** (Figure 2-Cp, Cc), providing the desired schedule or grounds for further iteration.

Planning involves constant re-evaluation, as users will often deviate due to unforeseen reasons from the schedule. Plans may change as one goes about the day [10, 39] impacting requirements, or external constraints may change unexpectedly (see looping arrows in Figure 2). The inputs and outcomes of the spatio-temporal planning workflow thus should be constantly re-evaluated. Computational **optimization** (Figure 2-Oc, Op) supports the dynamic evaluation of viable options; but the user must also be able to implicitly or explicitly update their requirements, and make choices (Figure 2-Cc, Cp) dynamically, mid-way through a planned schedule.

We propose a prototype based upon this workflow (Section 4.2) to validate the foundations of our approach. The prototype supports a core subset of the presented space-time constraints (Table 1) allowing us to gauge interest and user engagement with the concept of space-time scheduling in a user study (Section 5).

2.4 Limitations

Our framework is based upon an amalgamation of space-time planning scenarios that we have observed, experienced, and imagined. Surveying the literature has both further informed our framework and confirmed the validity of our general thread of ideas. Though we have confidence in this work, we acknowledge that our intuitive approach could, naturally, be limited. Future work may require us to re-approach the framework should alternative models improve, extend, or challenge our initial description of the general space-time planning domain.

3 RELATED WORK

We now give a general overview of relevant works that tackle similar space- and/or time-scheduling problems. We begin with a look at how spatial and temporal reasoning are supported in existing tools and research, followed by a review of work focused on space-time planning specifically.

3.1 Spatial and Temporal Reasoning

There are a variety of existing tools that serve to aid users in the planning process. On the one hand, there is the class of common navigation tools such as Google Maps [16] and its close cousins [17, 31] that offer support for A-to-B or multi-destination travel queries. These tools mainly facilitate *spatial reasoning*, in that they focus on providing users with directions, along with indications of travel time and traffic, integrated support of public transit schedule, as well as support for adding destinations based on their function (e.g., a gas station along the way). HCI researchers have explored ways to enhance these general-purpose digital mapping tools, further supporting spatial-related reasoning and interaction. SpaceTokens [30], for instance, provides widgets that allows users to quickly and easily access curated sets of important locations—each token acts as a proxy for a location that may be outside of the map viewport, which can then be selected and interacted with (e.g., for route creation without the need to navigate the map canvas). Block Party [48] is an augmented reality solution that provides a range of different views about a neighborhood to improve spatial awareness and give a general feel for the area when visiting a new location.

Our approach integrates several ideas from these tools to support spatial reasoning and differs in that we propose more of an integration of such interactive maps with scheduling support. We borrow from SpaceTokens' concept of location proxies, extending them to collections of locations, each with an assigned meaning that translates into constraints to help our system calculate a viable plan. Such meanings include "fit as many of these locations as possible" given temporal constraints associated with each location (user-defined, or imposed by the location), or "choose one of these locations", accounting for cases similar to when a user may be interested in fitting in a coffee break at some point, but may not necessarily wish to constrain the break to a single specific coffee shop that they should drop by.

Indicating travel duration is important in the space-time domain: when it comes to evaluating distances, travel time better represents the *cognitive distance* that people actually think about while creating travel plans [28]. Isochronal cartography embraces this idea, by distorting a map based on temporal data rather than being truthful to geographical reality [20, 21, 24], enhancing temporal reasoning with respect to locations. While an interesting hybrid combining space and time, such an ego-centric view only provides just-in-time information about locations within a temporal reach from a given point of origin. We instead opt for seamless integration of traditional mapping in a geographical map supporting multi-destination, tightly coupled with a time-based representation typical of scheduling-support tools.

On the other hand, there is the class of mainstream scheduling tools [2, 3, 15], which are the typical digital, interactive equivalent of our old physical agendas, conceptually speaking. Unlike location-based assistance tools, scheduling tools provide more support for *temporal reasoning* through a time-focused representation. Assigning locations to events is generally possible in the form of text, with a few [2, 3] also enabling addition and visualization of travel time in the calendar. This allows the viewer to explicitly account for location-related constraints in their agenda, albeit these constraints are not contextualized in a spatial representation.

Some scheduling systems provide support for optimization and alternatives, relevant to our goals. These include generating alternative schedules based on user-specified constraints [11], visualization of time constraints using availability bars [12], and support for semantic groupings of tasks for automation, including multi choice events [5]. These works are primarily concerned with algorithmic solutions to scheduling optimization problems, with little to no support for the location aspect. We build upon this existing work in the scheduling domain, adding further interactivity and extending ideas to *space*-time planning.

3.2 Space-Time Planning Systems

The HCI literature includes work that concerns motion in space-time [4], which mainly investigates data presentation, for the purposes of nostalgia [33, 40], task planning [25], and community events [9, 19]. Systems that support both space- and time- planning exist for specific domains, such as home finding [43] and tourism.

Travel itinerary generation tools consider space and time constraints in addition to eliciting priorities and preferences from users to integrate into scheduling optimizations. Most of the research in this domain aims to improve the algorithms behind tourism recommendation systems [6, 13, 14, 36] (i.e., Op, and Oc). One angle at which past works have attempted to augment the recommendation process is by involving the user through personalization. Some examples include explicitly prompting users upfront for their preferences [8], or implicitly defining priorities using information related to the subject (e.g., using a driver's past trajectories to identify driver preferences [46]; looking at a user's travel history to determine which tourist locations to recommend [29]).

Various studies have demonstrated that users like to customize their digital itineraries both while planning and while travelling [18, 37, 39, 41], and while personalization may loosely address to this, there is room for additional work looking at how to better support user customization and exploration throughout the entirety of the planning process and beyond. (e.g., after an initial plan has been made, while users are following through with their plans).

In many of the aforementioned works, more of a focus has been placed on improving algorithms and heuristics, and discussion related to the design of accompanying user interfaces is sparse or treated as a secondary consideration. In looking at more interactive systems for tour planning, some add an interactive layer to recommendation systems, allowing users to adjust the final output to either prompt regeneration of itinerary suggestions [8, 22, 26, 47], or to allow tweaks of existing plans [27]. Aurigo [45] is more sophisticated in that it provides an iterative workflow for point-of-interest selection—we tackle similar user-in-the-loop goals, but with a specific investment in the scheduling component of plan-creation, which is not the focus of their system. Other systems that allow calendar-based editing of itineraries only let users make tweaks to otherwise wholly system-generated solutions [23, 27].

Overall, there is much valuable work looking at how to improve the optimization aspect of the space-time planning workflow, as has been well-explored in HCI and other fields, such as operations research. There is, however, less attention dedicated to better understanding how to support user-motivated actions and iterations in this realm. We build on prior works to further understand the factors that may impact the construction and ongoing re-definition of a viable schedule, as well as identify where user input would be desirable, and how it could be supported in an in-situ, continuous workflow. We contribute an approach that allows the user to specify fixed and semi-flexible constraints within and between groups of events via direct manipulation, situated within spatial and temporal contexts supporting spatio-temporal reasoning. We discuss the design of user interfaces to facilitate exploration of options and alternatives, introducing more flexibility into space-time planning systems, while laying the groundwork to fully leverage the power of computational optimization from prior works.

4 SUPPORTING SPACE-TIME PLANNING: APPROACH AND PROOF-OF-CONCEPT

We now discuss our own attempt at improving the spatio-temporal constraints domain. In conjunction with past works looking at how to strengthen system optimizations, we seek to contribute to the overall effort in space-time planning by addressing sections of the workflow that have not been as thoroughly explored: the user-motivated actions. We first present an overview of our design goals and then describe our proof-of-concept prototype.

4.1 Our General Approach

Building on our synthesis of the problem domain and workflow (§2) and past works in this field (§3), we distill a set of design goals for an interactive tool addressing gaps in supporting personal planning.

G1. *Support spatio-temporal reasoning*: provide visual representations of interactions between spatial and temporal factors to facilitate decision-making and preference construction that align with **user motives** (Figure 2-M).

G2. *Enable workflow flexibility*: facilitate seamless definition and re-definition of both flexible and fixed spatio-temporal **requirements** (Figure 2-R), for workflow flexibility, quick iterations, and adaptability to change.

G3. *Minimize busywork*: leverage automation to minimize manual repetitive input and busywork computations by users, by calculating viable options satisfying **constraints** (Figure 2-C) formed by **requirements** (Figure 2-R) and **restrictions** (Figure 2-R), and **optimize** (Figure 2-Op, Oc) criteria to help the user resolve under- and over-constrained situations.

G4. *Maintain user agency*: support **choice** between alternatives, by letting the user (re-)define **preferences** (Figure 2-Cp) or agree upon a **compromise** (Figure 2-Cc).

In order to facilitate our goals encompassing both expressiveness and seamlessness, we require a rich vocabulary for expressing requirements **(G2)** within both spatial and temporal contexts. These include, e.g., setting fixed temporal constraints such as pinning an event at a given time, or specifying a large time window within which a given task (with its associated location) could be completed. To achieve this, we opt for a spatial representation, i.e., a geographical map tightly coupled with a temporal representation, i.e., a calendar-based visualization, where users can perform direct manipulations to (re-)define these requirements as they reason about space and time in conjunction **(G1)**.

Complex user motives call for combinations of inter-connected constraints to be defined; ones that stretch over multiple events, or that span more than one single fixed time or place due to inherent flexibilities, e.g., when a group of errands can all be accomplished at any time within the same given stretch of afternoon—though none are absolutely mandatory—or when deciding upon one event for the evening removes the need to schedule alternatives.

To encapsulate these semi-flexible situations, as well as the exploration of their various viable solutions, we introduce the concept of **requirement groups**: combinations of spatio-temporal requirements expressed collectively within a unified construct **(G2)**. A requirement group comprises of multiple locations sharing the same requirement property, and specifies interactions between these locations; it is treated as a whole when constructing a schedule. Such properties include, for instance, semi-flexible spatial constraints—such as choosing one of the multiple locations in the group to achieve a particular motive (e.g., one of the post offices to drop off a letter, but *any* post office would do)—or sequential requirements, such as a need to visit one location before another, though squeezing other tasks in between is possible.

The rich expression of constraints and their flexibility enabled by requirement groups also introduces increasing complexity, necessitating computational support to resolve the many within- and between- requirement group constraints. At a minimum, the user should be presented with a set of viable options that they may then browse through and select from. They should also be notified when no solution exists, prompting them to relax their constraints instead of being left to figure it out themselves through trial an error **(G3)**. Advanced optimization algorithms can further facilitate the user's task of choosing between multiple options, e.g., by sorting the possible alternatives based on certain criteria, or by prioritizing certain constraints to resolve an impossible schedule. A good balance must be achieved between computational support **(G3)**, fluid and quick interactions **(G2)**, and user agency **(G4)**.

4.2 The Space-Time Planner Prototype

We built Space-Time Planner (Figure 1), a mobile application implementing the core functionality of our general approach to creating space-time schedules in a flexible workflow (Figure 2). In this initial proof-of-concept, we support a subset of the spatio-temporal constraint domain, zoning in specifically on travel-time restrictions and their impact on spatial and temporal decision making. Our design combines a schedule (Figure 3a (left)), a map (Figure 3a (top)), and a palette of locations (Figure 3a (right)). Basic spatial and temporal requirements are connected to form an *event block* by dragging a location from the place palette directly to the schedule **(G2)**, resulting in a new event block with fully fixed spatial and temporal requirements. Upon the addition of further events, the system automatically inserts travel-times between locations on the schedule **(G3)** and displays the route on the map **(G1)**. We include mechanisms for resolving temporal constraint conflicts that are discussed below in a walkthrough.

Requirement groups are implemented as bucket-like blocks containing collections of events with semi-flexible constraints, and thus the potential for many viable scheduling options. We use optimization functions to compute and present to the users a set of alternative solutions to choose from. The group ranks these options by travel time, presenting the option with the least travel time by default—to swap-out for other optimization functions would be a trivial modification. Importantly, regardless of the underlying computational algorithm to rank the set of possible solutions, users are always allowed to browse and compare the alternatives themselves, should they be curious or have differing preferences **(G4)**. In the following system walkthrough, we demonstrate the specifics of the two requirement groups that we have developed: one-of groups and as-many-of groups.

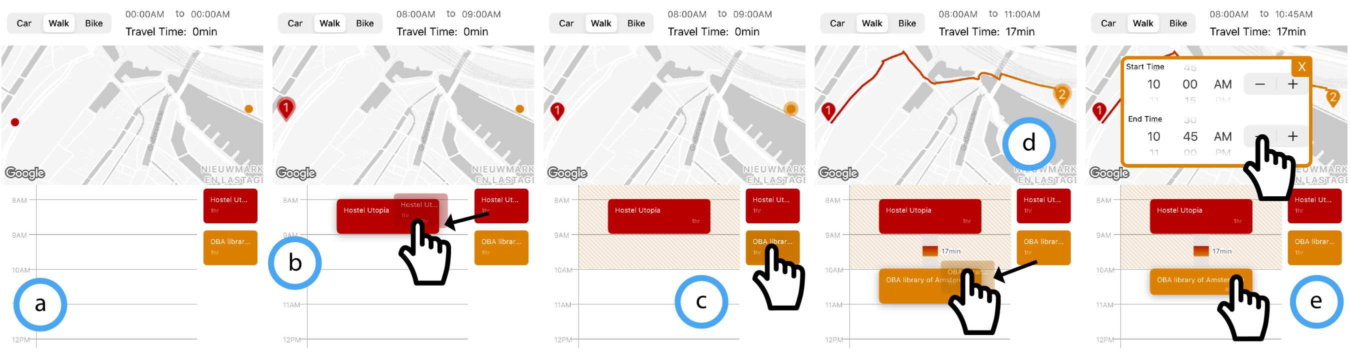

Figure 3: The basic flow of adding locations to the schedule. Locations are dragged into the schedule from the *place palette* (b). Hours of operation are represented via hatching of closed hours in the schedule (c), and made visible when interacting with (i.e., holding or dragging) a location. Adding a second event to the schedule (d) creates a route between the locations on the *map view*, visualizing a corresponding travel time gradient in the *schedule view*. The grey area surrounding the travel time gradient in the schedule represents extra time. Tapping on an event block opens up a corresponding time-editing dialog for start, end, and duration times (e).

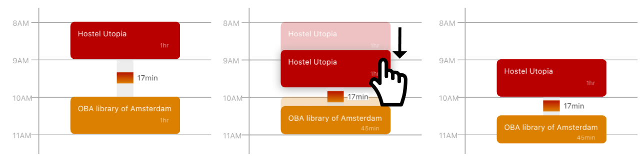

Figure 4: Temporal conflicts are not possible in Space-Time Planner. The block that is currently being modified can be placed wherever the user desires and pushes surrounding blocks as needed, keeping them as close to their original position as possible.

4.3 System Walkthrough: A Day in Amsterdam

James has a day to spare in Amsterdam before he leaves the city to visit a friend in the countryside. His friend has given him a list of recommendations to guide his day. He uses Space-Time Planner to help figure out how to fit in these suggestions, which ones to fit, and when to fit them.

Known Requirements: Initial Inputs – James opens up the tool, where he has already loaded some locations into the *place palette* on the right (Figure 3a). He has a good idea of how to start the day, so he begins there, creating a basis to figure out the yet undecided parts of his itinerary. James plans to leave his hotel for the day around 9AM, so he drags the hotel location into the schedule according to this temporal requirement (Figure 3b). Next, as a lover of books and architecture, he wants to visit the central library, the top floor of which has a fantastic views on sunny mornings. As he holds down on the location in the palette, he notices that it does not open until 10AM (Figure 3c), so drags it in from the palette, accounting for this time restriction (Figure 3d). He will probably not spend a full hour there—he adjusts the duration to 45 minutes, by tapping on the event block and using the time-dialog box that appears (Figure 3e).

Resolve Time Constraints: Pushing – James has some extra time in the morning now. He was optimistic about an 8AM wake-up anyway – being more realistic, he decides to wake up at 9AM, to leave the hotel at 10AM. To reflect this temporal change of mind, James shifts the hotel block, and the library event block gets pushed forward in time accordingly, as the system automatically resolves conflicts between his new input and the travel-time restrictions **(G3)** (Figure 4). As he moves the hotel block, he observes the shadow of the initial block states remaining on the schedule, demonstrating the relative adjustment that the system is making. Satisfied with this change, he releases the hotel block.

Curate Group of Locations: Palette Editing – James must decide on the rest of his day now. His list of recommendations includes a few Dutch pancake houses, which he is curious to try. He trusts his friend's taste, open to any option that involves pancakes. To help him choose, James decides to create a *constraint group*. He needs to add the locations to his palette, then build the group. Tapping "edit" from the bottom-right of the interface (Figure 1a) opens up a screen where James can edit his place palette. He taps "+place" and uses a search dialog to add all restaurant locations, after which James opens up the group creation dialog by tapping "+group". He creates a corresponding "pancakes" group, setting it to be a requirement group of type *one-of*. In the place palette, he drags the lunch locations into his new group and returns to the schedule view, with a now updated place palette (Figure 1b).

Pick One Of Several Options: One-Of Groups – *One-of groups* assist in selection between different location options for a given block of time. The curated set of locations provided to a one-of group serves as a flexible location requirement, and situating it in the schedule sets its temporal requirements. With his new lunch group, James will be able to fix a time for lunch, but keep the location options flexible. As he drags the group into the schedule to start at 12PM, the total travel time for each permutation of the schedule that would result from the different locations is automatically calculated. By default, the presented location for the one-of group event is the option that minimizes the travel time, here the restaurant closest to the library (Figure 1c). James can browse how the different alternatives cause changes in route and travel time (Figure 1d, e), but he will wait until later, as the one-of group will continue to adapt while he builds his schedule.

Fit In As Many As You Can: As-Many-Of Groups – Next, James considers his recommended activities for the day. He wants to know what there is time to see, and what order makes the most sense to

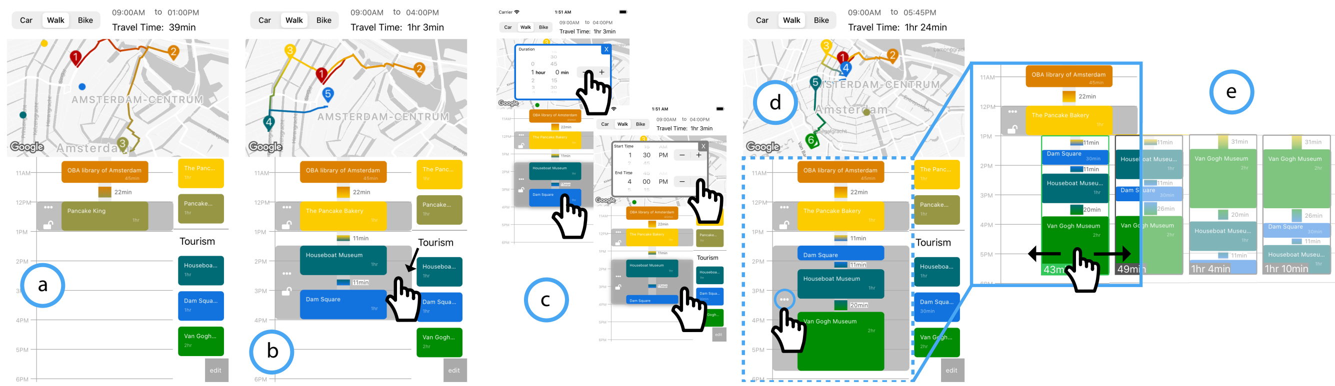

Figure 5: Before (a) and after (b) adding an *as-many-of* group to the schedule. The group automatically schedules the largest subset of locations that can fit within the block's duration, in the arrangement that results in the least possible travel time. The pre-existing *one-of* group also updates to minimize travel time upon this change (a to b). The overall block duration of a group can be adjusted, as can the durations of independent destinations within an *as-many-of* group (c). The map updates according to such changes (d). Alternative schedule permutations can be browsed in-situ by tapping on the "..." icon (e).

Figure 6: By holding down on the time-ticks of the schedule, absolute-time along a route is visualized on the map. The user may scroll down the ticks and the absolute-time animates across the route accordingly.

see it in. Another type of requirement group serves this purpose: the *as-many-of group*, a collection of events that have fixed locations and durations, but no start/end times. Rather, the group as a whole has a start and end time, serving as temporal boundaries to all events in the group. The system creates permutations to fit as many events within the temporal boundaries as possible.

James picks out a handful of activities that look interesting to see what will fit, adding them to his place palette. He creates an *as-many-of group* and names it "Tourism", then drags the locations into the group in the place palette. He can set the default duration for individual activities, and decides to allocate two hours for the visit of the Van Gogh Museum.

With the group now set up (Figure 5a), James drags it into his schedule after lunch. He notices that the lunch location has updated to minimize travel time within the rest of his itinerary (Figure 5b). Now paying attention to the *as-many-of group*, the group only fits two activities at first. He considers that he would probably spend less time at Dam Square and adjusts its duration to 30min (Figure 5c). If he only had 2 hours to spare the current schedule would suffice, but he has more time than that. He taps on the group background to expand its duration (Figure 5c), and watches it update; now 3 locations are able to fit, and his overall route updates accordingly in the map as well (Figure 5d).

Browse Alternatives – James wants to see how choosing other permutations of his activities would impact his schedule. He taps the "..." to scroll through the options (Figure 5d), and they appear embedded into the schedule (Figure 5e). He can select the 49min travel-time option and lock it in place using the lock icon if he is set on it, then the system will no longer optimize travel time when other things change. If he keeps it unlocked, the group will continue to update if he decides to add more activities to his itinerary, or some options for dinner later. James returns to his list of recommendations and continues to consider. It is nice to keep his options open, yet have it all laid out in front of him at the same time.

Previewing Itinerary: Time Along Route – The next day, James is ready to set out to meet his friend in the village Boekel, in time for lunch at 12PM. His route passes through an area with tulip fields—he would love to snap a picture on the drive over. Unfortunately, it is raining this morning. The forecast predicts that the sun should come out after 11AM, will he be passing through the fields around then? He uses his schedule to make the query by dragging his finger along the time-ticks on the left, which produces an overlay over the route that is in sync spatio-temporally (Figure 6). It looks like he might catch the tulips just after the rain.

4.4 Implementation

Space-Time Planner is an iOS application written in Swift using the XCode development environment. It uses the Google Maps and Places iOS SDKs to create the map view and obtain point-of-interest details, and the Google Directions and Distance-Matrix APIs to obtain routes for rendering and the travel time matrices with which requirement group optimizations are calculated. We borrow icons from FontAwesome (https://fontawesome.com/license).

The scheduling algorithm is not novel, but we describe its inner-workings for the sake of transparency and reproduciblity. Our scheduler has 2 steps: 1) local optimization by determining all viable options for the as-many-of groups, and 2) global optimization by determining which combination of options result in the itinerary with the minimum travel time. For step 1, we implemented a brute force algorithm that, for each as-many-of-groups, browses through every permutation of the set and returns the configurations that are temporally viable and maximize the number of visited locations. For step 2, we find every sequence of 1 or more consecutive requirement groups, and generate a list of the possible combinations of options from each group by performing a cartesian product between their sets of options. We return the solution that minimizes total travel time, including the travel times into and out of each sequence, to fixed events outside of them.

5 USER EVALUATION

In order to learn more about the potential and limitations of our approach, we conducted an in-person usability study (before the pandemic) with target users using our proof-of-concept prototype. Our goal was twofold: 1) we sought qualitative feedback on the space-time planning concept, workflow, and approach as a whole, and 2) we tested the usability of our particular solution, Space-Time Planner. Our study encompasses a structured replication task to familiarize participants with the prototype, followed by a semi-structured open-ended task, to encourage more natural explorations.

Participants and Apparatus. We recruited 12 participants (7 male/5 female, ages 19–35, mean = 27.75) through word-of-mouth convenience sampling, university mailing lists, and snowballing—we acknowledge that these methods have the potential to result in some desirability bias within our results. Participant occupations include physician (n=1), sales associate (1), marketing manager (1), researcher (1), and graduate (6) and undergraduate student (2). All participants reported using smartphones on a daily basis. The prototype was operated in-person on an iPhone 7 (4.7 inch screen) running iOS 13.3. At the time of the study, an earlier version of the prototype was used, which did not yet include the hours of operation (Figure 3c), nor the absolute-time preview (Figure 6) features.

Overall Procedure. After filling out a demographics form, participants were provided with a description of the prototype and concept, through a self-contained simple scenario demonstrated by the experimenter. Participants then completed a walk-through tutorial with the tool, following along with a slideshow. They were encouraged to go at their own pace and to ask any questions. When ready, participants were provided with general instructions to the tasks: to complete them as best as they could without assistance, and to think aloud. Each task was described in text on a sheet of paper. Participants were instructed to go at their own pace, but that we would interrupt the task after a certain duration to keep them on track. Tasks were audio, video, and screen recorded. Following the tasks, participants filled out a brief usability questionnaire and took part in a semi-structured interview. The study took about 1 hour. Participants were given a $15 gift card as compensation.

Tasks. Each task was centered around a scenario. The purpose of the first task (T1) was to familiarize participants with the interface and capture usability issues. The provided scenario required participants to plan out a route for an urban arts festival. We provided a pre-loaded set of locations corresponding to the locations described in the scenario. In the scenario, the participant's schedule begins by leaving work at 5PM. After work, they aim to see many different art installations with equal priority, except for one of top priority and at a fixed time, 10PM. The rest can be seen at any time. Participants were asked to create a schedule for this scenario within 10min, and to let the experimenter know which art installations they will be able to see given the constraints.

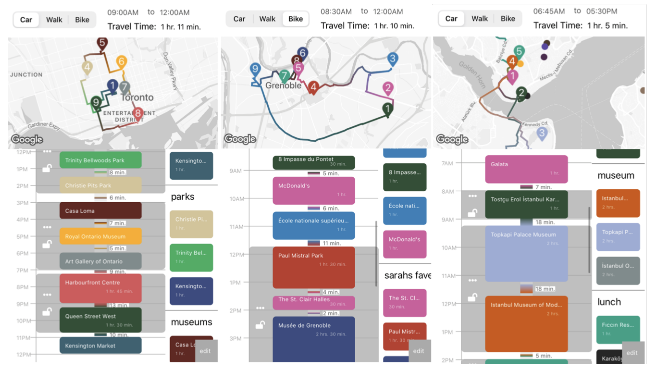

Figure 7: Examples of participant-created schedules.

We designed two smaller follow-up tasks following the main portion of T1 to simulate change of plans, further exposing participants to requirement group behaviours. In the first one, dinner must be added to the schedule, and the participant must determine which of two potential dinner locations saves more time, as well as report on which art installations they are now able to fit in their schedule given the added event. In the second follow-up, one of the installations has a technical difficulty, so participants must remove it from their schedule and report on changes. Participants were given 3min for each of the follow-ups.

The second task (T2) was open-ended. The purpose of this task was to give the participants an opportunity to use the interface more organically. In this scenario, a friend is visiting a city that the participant is familiar with, and is requesting an itinerary for their day. Participants were encouraged to select any location they like. Some guidelines were provided, such as times to wake up and get home, as well as general activity ideas, none of which were strict requirements. Participants were given 15min to complete T2.

5.1 Results

Below we report on main insights from the study, gathered from task observations, questionnaire results (partly summarized in Figure 8), and interview analysis.

5.1.1 General Feedback. In general, the prototype was received positively. In our post-study questionnaire, participants responses to *"I think I would like to use this system in my own life"* were all high on the 5-point Likert scale (5 "strongly agree", 7 "agree"). Everyday-life usage was repeatedly noted, mainly for errands and busy days. One participant noted that it could be especially useful to help him to work around a recurring, distant appointment, often scheduled inconveniently in the middle of his day (P11). The travel scenario from T2 was mentioned by participants to be relatable, and six indicated they could imagine using Space-Time Planner for their own trip planning, or for showing a friend around where they live. All participants leveraged the vocabulary of spatio-temporal

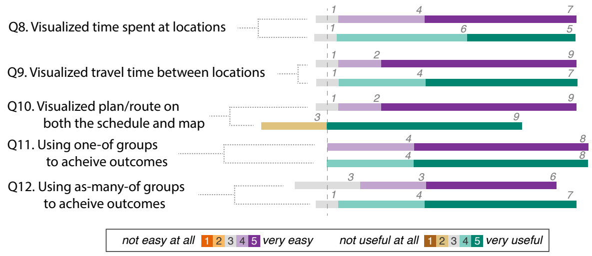

Figure 8: Summary of participants rating of the ease of use and usefulness of features.

parameters offered by our proof-of-concept to create interesting itineraries in various cities such as Toronto (Canada), Grenoble (France), or Istanbul (Turkey) (Figure 7).

*5.1.2 Supporting Spatio-Temporal Reasoning **(G1)**.* When using status quo tools to reason about multi-destination plans in their everyday lives, many participants noted that there is often travel-time related tedium or guess-work involved: *"you kind of have to guess what the most efficient time is like most efficient path."* (P3). In using the system, participants unanimously commented on the clarity of Space-Time Planner with respect to understanding time and the impacts of travel-time restrictions between locations, also reflected in the positive ratings of the visualizations in terms of ease of use and usefulness (Figure 8,Q8-9). We witnessed confidence in participants' grasp of time while performing the tasks, as articulated by P11: *"it improved kind of a conceptualization of the time that I actually have to spend"*. This confidence enabled users to realize and explore different options, for example, wondering if there would be enough time for an added activity, and being able to confidently approach space-time scheduling questions *"Does she have time [to go there]? Let's see, I think she might... Oh, she does have time!"* (P10).

There were other temporal factors participants expressed consideration of while scheduling, which were not explicitly visualized by the system. Some examples are hours of operation for a museum (P8) (a feature which was later added to our prototype, post-study), and daylight hours for going swimming outdoors (P2). Many noted the impact traffic could have at different times of day, which they wanted to see visualized explicitly. Though the exact information was not directly available, it is noteworthy that participants vocalized these holistic space-time scheduling questions as they performed the tasks.

Visualizing the map and the schedule side by side was found "very useful" by most participants (9) (Figure 8, Q10), who noted its value as an all-encompassing overview: *"it's nice to see the numbers and also the colors telling me the order of places I'm going"* (P10). One participant had tried to create such visualizations on her own: *"I feel like I've had two tabs open the calendar and the map before; I've done that before it so it's nice to just have it all in one little view"* (P10). Participants noted that the linked visualization provided a *"justification for the plan"* (P3), to help conclude that their temporal schedule mapped to *"reasonable [spatial] results"* (P1).

On the other hand, three participants did not find much value in the combined view, rating it as "not really useful"; admitting that they did not look at the map much at all as they were focused on the schedule input. One said: *"I just wonder what additional information map gives me, I'm not sure"* (P6). Three participants expressed desire for Yelp-like features to provide supplemental details and support for point-of-interest selection.

*5.1.3 Constraint Expression **(G2)**.* Defining basic spatio-temporal requirements by dragging and dropping locations into the schedule posed no difficulties to participants during the tasks. However, this changed when schedules became more crowded, resulting in accidental event pushing. A desire to temporally "pin", "lock", or "fix" specific events was expressed unanimously (we illustrate how such feature fits within our approach in a usage scenario in §6). Although participants expressed appreciation for the event-pushing feature conceptually for the purpose of automatically resolving time conflicts—*"very useful compared to Google calendar...not creating those impossible stuff" (P7)*—our implementation allowed for undesired pushes to go unnoticed while participants were focused on authoring their schedules, causing frustration and confusion.

Requirement groups were understandable and useful (Figure 8, Q11-12), subject to remarks such as *"it's easy to use and makes sense"* (P10). Participants noted that they fit well with their mental model.

One-of groups in particular were described as "simple", "easy to understand", "natural" (P7), and "flexible" (P5). As-many-of groups, although useful, posed some extra problems with regard to automated pushing. Adjusting the blocks to fill the time between two fixed places, participants often made a series of tiny adjustments in attempts to create a perfect fit, such as in T1: *"I'm going to extend the duration of this thing of the art group as much as possible without moving the [other event]"* (P3). Locking events into place could alleviate this issue. However, the repeated resurfacing of this issue with the as-many-of block in particular reveals that the *block* analogy could be further improved to better align with users' expectations, as P12 describes: *"the way I'm seeing it is that I want to like fill the time in between these things and the way it's [the system] seeing it is I have to spend two and a half hours doing stuff, so it's like there's a mismatch of how I'm imagining how it's imagining."*

Participants used workarounds when the as-many-of groups could not support all the requirements they had in mind: prioritizing an event above others within the group; scheduling a fixed event in the midst of the group; restricting some locations to particular timings due to daylight; defining precedence requirements. Workarounds involved splitting up as-many-of groups into multiple groups, pulling particular events out of the group to put straight into the schedule, and searching the alternatives view for a route that satisfies the problem. (However, P11 noted this last workaround as a pro, not a con.) We demonstrate in usage scenario (§6) how richer system capability could overcome these limitations.

*5.1.4 Supporting diverse workflows **(G2)**.* Most participants took a streamlined approach to the open-ended task (T2) by creating all their places and groups in the palette first, then dragging them into the schedule relatively chronologically. When dragging locations into the timeline, participants unanimously began with temporally-fixed events (according to the task, not differentiated in the interface) first. Some would proceed chronologically from there, or others would fill in all the temporally-fixed events to *"create a skeleton"* (P3) or *"have a bit of structure"* (P11) before moving along to temporally-flexible events. Two participants had highly iterative approaches, and the tool successfully supported them in doing so.

P10 iterated on her creation multiple times, looking to the tool to inspire further planning. Seeing that there was more time available in the schedule, and looking at the map, P10 noted that perhaps the friend, Amy, could see where she went to school, which was near the accommodation she had picked out. She experimented accordingly: *"Do I want Amy to see where I went to school? Does she have time? Let's see, I think she might ... Oh, she does have time"*. After finalizing the schedule, she went back in to adjust further: *"She has more time for shopping"* (P10).

P11 went back and forth between modifying the place palette and adding items to the schedule. As he developed new ideas, he was able to create new groups and adjust accordingly. He looked to the as-many-of alternatives view repeatedly for inspiration: *"maybe it would be a bit more fun to actually go to Plato's Closet and then go to the arcade bar."*

*5.1.5 Automation **(G3)**/**(G4)**.* In existing tools, participants were familiar with comparing travel times: *"how long does it take from A to B? and then what is like from A to C? and then manually do it"* (P2). Thus they found value in Space-Time Planner as *"this does all that calculation for you"* (P2). The as-many-of groups were especially well-received with respect to the automated sequencing: *"it's better to not have to think about that part"* (P10).

When prompted during the post-study interview to discuss the trade-off of manual control and automation available in the tool, almost all participants said they would prefer a middle ground, with one participant indicating his preference for a fully automated system. Many participants were pleased with the provided balance, making comments such as *"at the moment, it's sitting in a nice place already"* (P4) and *"that's like the best place to be, in the middle"* (P2). Comments were made not to move towards a more automated or more manual system overall, but to increase capacities in both directions, described by one participant as *"fine-tuned control over the automations"* (P12). *"It helps you if you're indecisive, they'll choose this for you. But if you do have an opinion of like, what you want to do, like you have to make a decision, then you have the control to do that too"* (P2). Many noted that they like the convenience of automation in general, but prefer the ability to edit the results.

6 SCENARIOS

In our proof of concept, we included the core functionality to give users a sense of how a tool building on our general unified workflow approach might work and to gauge initial reception. We now present a few mock usage scenarios that touch upon other aspects of the space-time planning workflow, to demonstrate both the vast potential to extend our proposed solution, and the complexity inherent in the overall problem space.

6.1 Enterprise: Movie Set Production Assistant

Leyla is a movie production assistant on a movie set. Her responsibilities involve running around both the film set and the city, to gather material necessary to keep the production running smoothly. While a small cog in a large machine, her personal schedule create dependencies that impact the entire film. Through this scenario (Figure 9), we step through a series of iterations of a space-time plan, demonstrating how our expressive approach supports users trying to be organized and efficient in a constantly evolving context.

In Figure 9a, we see a snippet of Leyla's day: she clocks in at HQ and attends the team-debrief. She then has a list of errands to accomplish before she assists with a set in the afternoon, before which she hopes to have some free time to take a break and answer e-mails. This set of requirements is simply captured by a familiar Space-Time Planner (§4) setup that includes an as-many-of group.

Group nesting – One of the errands involves picking up lighting equipment, which Leyla can collect from one of various warehouses around the movie production grounds. The options for this particular motive can be captured by a one-of group, which could in turn be nested within the larger as-many-of group of errands by dragging one palette into the other, adding one more entry to the "Tasks" palette, but this one allows flexibility of location (Figure 9b). When calculating options, the system simply computes all possible permutations resulting from the nesting. In this scenario, nesting is visualized using a dark gray background, and browsing through alternatives remains possible at any level. Adding group nesting to our current vocabulary of constraints enables a wide range of rich, complex scenarios.

Precedence groups – At HQ in the morning, Leyla expects to find clothing to be dry-cleaned at her desk. Instead, she learns it needs to be picked up from the costume department, then cleaned. We can facilitate the expression of this sequence-level requirement by introducing another constraint group type: the precedence group. A must come before B, or in this case, the clothes must be picked up before they are brought to the dry-cleaners (Figure 9c). Leyla creates this group and adds it to the "Tasks" palette. Together, the different constraint groups and nesting encapsulate all of her requirements. The tool can now present a valid schedule permutation that optimizes both travel distance and duration so that she does not have to think twice about it—or, if she *does* want to think twice about it, the system supports her thought process of querying the different possibilities by swapping through options, and dragging nested events and blocks around.

Expressing fixed requirements: pinning – Leyla's day is more full, but still manageable, until she is informed that the intern is away today and someone else will need to arrange the coffee and treat platters at the team-leads meeting. This inflexible event falls in the midst of Leyla's other tasks, but she ought to adapt, or the whole production would experience delays. Leyla pulls up the space-time tool to re-organize her plans. To establish a requirement that is fully fixed in both time and space, we can introduce a simple mechanism for expressing requirements: the pin, as shown in Figure 9d. With her other constraints already set up, it is easy to sit back and let the system work around the pinned event, to quickly understand how this change impacts her day—her break is getting smaller and smaller, but she will have saved the day, and will still be able to handle all her responsibilities in good time.

Real-time updates: conflict resolution – Given the changes in her schedule, Leyla has completed her morning tasks but is left with a shorter break than she had hoped. Still, she wanders off for a walk to relax. During this walk, she spends some time walking away some distance that does not match the plan proposed by the system. To adjust to her going "off script", the tool continuously updates taking into account these new implicit constraints. Leyla still does

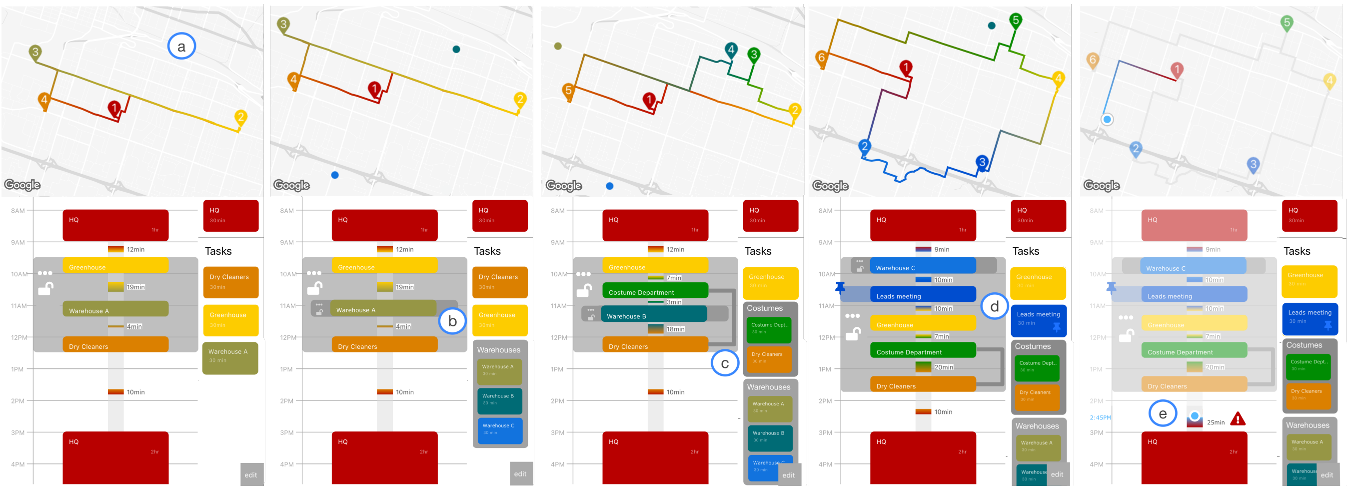

Figure 9: Iterations on the space-time schedule of a movie-set production assistant. (b) and (c) demonstrate nesting functionalities in the as-many-of group: a nested one-of group and a nested precedence group, respectively. (d) demonstrates the definition of a fully-fixed, pinned event, and additionally how the system optimizations work around it, updating with each newly defined constraint. In (e), with the many constraints already in place, the system is able to react to just-in-time changes and, even in parts of the schedule that the user decides to leave empty, and can notify the user of a potential incoming conflict.

not realize that she is approaching the brink of becoming late for her next responsibility, but as it is already aware of when and where she is required to be next at a fixed time and location, the Space-Time Planner can notify her of a potential conflict (Figure 9e). Given her current location, would she like to push back her next commitment? Cancel it? Be late? Real-time conflict resolution is a challenging problem to solve from a user interaction and perspective, and can be reserved for future work. However, the main takeaway here is that the basis provided in our space-time framework can easily be extended to support such in-situ functionality, whether that means automatically making changes on Leyla's behalf, or simply letting her know of the situation. Leyla, having been notified of her potential tardiness, decides to take her time walking back to HQ anyway—she has had a hectic day and deserves some slow time, even if that means being a few minutes late per the plan.

6.2 Multi-Day: Attending a Conference

Colin is attending a conference in Vancouver and balancing time between talks, social events, and other sightseeing activities. In Figure 10, we feature a multi-day itinerary—it consists of events Colin is sure he wants to attend, including the confirmation that he has time to go hiking on Sunday and return in time for his evening train. He uses as-many-of groups: one to help him narrow down his recreational activities (Figure 10a), and another one to handle the talks and presentations he is interesting in (Figure 10b), but that he is not necessarily committed to attending. All events are optional, but each of the talks has its own fixed time—a combination of constraints that is expressed by pinning the event in time, while keeping it in the as-many-of group. If he feels like packing his Sunday really tightly, he could stick in some more activities in the morning. He is not sure yet, but it is nice to see all the possibilities in one place, for activities both at and outside the conference.

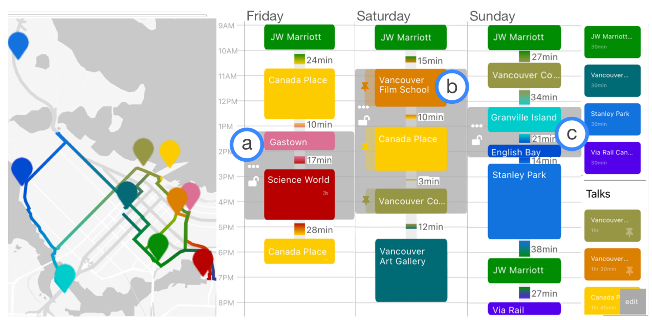

Figure 10: The multi-day space-time schedule of a conference attendant, balancing flexible recreational activities (a, c) with scheduled talks. There are a number of talks that he is equally interested in going to, but many of which overlap—he represents their equal priority using an as-many-of group, and expresses their inflexible scheduling constraints by using pins *within* the group (b).

In this scenario, we bring space-time planning at scale, across multiple days. This poses the interesting question of how to best leverage the power of the as-many-of group (or any requirement group), when the time over which to satisfy these motives is scattered over a larger schedule. One option here would be to expand the whole block from Friday through to Sunday, but this would require explicitly specifying additional constraints, e.g., not scheduling something during the night or when Colin does not feel like doing anything at all. Another approach would consist of adding some form of "memory" to the constraints groups, that is, specifying multiple non-contiguous time windows when to satisfy the associated constraints.

7 DISCUSSION

Our study results suggest that our approach fills an important gap between route mapping and calendar scheduling. Participants were excited by the idea and engaged in the tasks, referring to the tool as useful and practical. The many suggestions we received on ways to extend the system and ideas featured in our mock scenarios are all inscribed in our general framework (Figure 2), affirming the validity and potential of our personal "space-time and choice" planning approach. Valid remarks arose about "too much planning" perhaps limiting real-world adoption, which can be evaluated by improving the system and deploying a field-study in future work.

The requirement groups supporting the definition of both soft and hard requirements are one of the key ingredients of our approach. Participants were able to grasp these constructs conceptually and praised the possibilities that such a tool enables, provided the algorithmic back-end computes a reasonable set of solutions. Our study suggests that there is value in further exploiting the power of requirement groups: in particular, the as-many-of block in our implementation presents some limitations, in that the insertion of a fixed event "within" the block is currently not supported. While our scenarios (§6.2) illustrate the expressive power of extending the current implementation, the interaction and visualization of such new feature is subject to future work.

We opted for the block analogy to remain consistent with the other requirement groups and regular fixed events, but find that other metaphors could align better with the users' mental model and expectations. As discussed in our multi-day scenario (§6.2), an avenue for exploration is to allow users to "brush" the period(s) within which they would like the system to try to fit events from an as-many-of, precedence, or other requirement group. Future work should also investigate ways to better support manual iterations over a system-generated solution, without compromising simplicity and usability of the interface.

Just like any tool supporting creative authoring, one of the main challenges resides in finding the right balance between automation-enabled guidance and user input to control the final outcome. For one, it is delusional to believe that we can design a solution where user inputs completely capture their preferences a priori, for the simple reason that preferences involve too many inter-dependant factors, can be ill-defined, and inherently dynamic. Automation methods need such input to provide *the* optimal solution for a particular user at a particular time, but we posit it would be too demanding on a user to be required to specify all of the parameters constantly. In contrast to other works, we opted for an approach that provides sufficient automation support to prompt the user with viable solutions, but largely relies on fluid, direct manipulation interaction to circle through and iterate upon seed solutions as the user refines their preferences. We currently leverage travel time as the main factor to optimize while generating solutions. Our tool could be easily extended to account for multiple weighted criteria such as accessibility, but feel that the benefits of optimization should not come at the cost of tedious user input of a large set of parameters, something supported by our user study.

On this note, another challenge we faced lies in how the system should behave when conflicts emerge. For example, participants found our "push" functionality intuitive, but not always predictable. System response to changes in under-constrained schedules is ambiguous and the expected behavior is often subjective. Sometimes a user may expect their changes to remain local and not impact the rest of the schedule even in the presence of conflicts, and in others, the expected behaviour is that the system globally re-optimize the schedule to generate a feasible plan. The pushing mechanism was deemed intuitive since on the fly resolution of conflicts is generally desirable. Consider however, a "pull" mechanism where moving an event earlier, could pull subsequent events with it, and/or extend the duration of all semi-flexible events until the next fixed event. These interaction design choices can be explicit user preferences, set to the dominant expected behavior, or a default outcome that would be cumbersome for a user to manually specify. Bearing in mind discoverability and limited input problems on mobile devices, the user input vocabulary can also be augmented to disambiguate common and conflicting behavior. Expected behaviors can further be learned in context from user interaction over time.

Finally, our approach could support scenarios spanning multi-day planning, as well as just-in-time, in-context adaptability in the face of uncertainty and evolving plans. While our proof-of-concept and study were designed with such qualities in mind, and our scenario give a gist at how to pursue such direction, we have yet to explore the unique design and interaction challenges in supporting seamless transitions between a micro-scale personal schedule over the course of a day, and macro-scale planning spanning longer periods of time. We also envision our approach as being able to adapt to incorporate dynamic space-time opportunities into a current schedule, and prompting users on these opportunities [10]. Overall, the positive study results and opportunities featured in usage scenarios both indicate that personal space-time planning is a rich area for future research.

8 CONCLUSION

We address the problem of interactive personal scheduling in space and time, to bridge the gap between digital calendars and digital maps, supporting space-time reasoning in a unified, flexible workflow. Our synthesis of the personal space-time scheduling domain provides a foundation for a prototype application, and can facilitate design thinking for future work on interactive tools that address this problem. Our proof-of-concept application on a mobile device demonstrates the utility of our approach, and results from our study suggest a need for such applications that judiciously combine user interaction and choice, with space-time optimization. We present a few scenarios to showcase how our prototype could be extended to accommodate changes in real-world scenarios, and offer ideas for future work in the area.

ACKNOWLEDGMENTS

We thank the members of the DGP lab and reviewers for their valuable input. This work was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] J Anderson. 1971. Space-time budgets and activity studies in urban geography and planning. *Environment and Planning A* 3, 4 (1971), 353–368.

[2] Apple. 2020. Apple Calendar. https://support.apple.com/en-ca/guide/calendar/welcome/mac

[3] Busy Apps. 2019. BusyCal. https://www.busymac.com/busycal/

[4] Benjamin Bach, Pierre Dragicevic, Daniel Archambault, Christophe Hurter, and Sheelagh Carpendale. 2014. A review of temporal data visualizations based on space-time cube operations.

[5] Jacob Bank, Zachary Cain, Yoav Shoham, Caroline Suen, and Dan Ariely. 2012. Turning personal calendars into scheduling assistants. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2667–2672.

[6] Joan Borràs, Antonio Moreno, and Aida Valls. 2014. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications* 41, 16 (2014), 7370–7389.

[7] F Stuart Chapin Jr and Henry C Hightower. 1965. Household activity patterns and land use. *Journal of the American Institute of Planners* 31, 3 (1965), 222–231.

[8] C. Chen, D. Zhang, B. Guo, X. Ma, G. Pan, and Z. Wu. 2015. TripPlanner: Personalized Trip Planning Leveraging Heterogeneous Crowdsourced Digital Footprints. *IEEE Transactions on Intelligent Transportation Systems* 16, 3 (2015), 1259–1273. https://doi.org/10.1109/TITS.2014.2357835

[9] Xiang'Anthony' Chen, Sebastian Boring, Sheelagh Carpendale, Anthony Tang, and Saul Greenberg. 2012. Spalendar: visualizing a group's calendar events over a geographic space on a public display. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*. ACM, 689–696.

[10] David Dearman, Timothy Sohn, and Khai N Truong. 2011. Opportunities exist: continuous discovery of places to perform activities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2429–2438.

[11] Andrew Faulring and Brad A Myers. 2005. Enabling rich human-agent interaction for a calendar scheduling agent. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1367–1370.

[12] Andrew Faulring and Brad A Myers. 2006. Availability bars for calendar scheduling. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems*. ACM, 760–765.

[13] Damianos Gavalas, Vlasios Kasapakis, Charalampos Konstantopoulos, Grammati Pantziou, Nikolaos Vathis, and Christos Zaroliagis. 2015. The eCOMPASS multimodal tourist tour planner. *Expert systems with Applications* 42, 21 (2015), 7303–7316.

[14] Damianos Gavalas, Charalampos Konstantopoulos, Konstantinos Mastakas, and Grammati Pantziou. 2014. Mobile recommender systems in tourism. *Journal of network and computer applications* 39 (2014), 319–333.

[15] Google. 2019. Google Calendar. https://calendar.google.com/calendar/

[16] Google. 2020. Google Maps. https://www.google.ca/maps

[17] SoLocal Group. 2020. Mappy. https://en.mappy.com/

[18] Dean MG Hargreaves and Toni Robertson. 2010. Planning travel as everyday design. In *Proceedings of the 22nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction*. ACM, 21–24.

[19] Blaine Hoffman, Harold Robinson, Keith Han, and John Carroll. 2012. CiVicinity events: pairing geolocation tools with a community calendar. In *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications*. 1–8.

[20] Sungsoo Ray Hong, Yea-Seul Kim, Jong-Chul Yoon, and Cecilia R Aragon. 2014. Traffigram: distortion for clarification via isochronal cartography. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 907–916.

[21] Sungsoo Ray Hong, Min-Joon Yoo, Bonnie Chinh, Amy Han, Sarah Battersby, and Juho Kim. 2018. To Distort or Not to Distort: Distance Cartograms in the Wild. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 628.

[22] Hsun-Ping Hsieh, Cheng-Te Li, and Shou-De Lin. 2012. Exploiting Large-Scale Check-in Data to Recommend Time-Sensitive Routes. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing* (Beijing, China) *(UrbComp '12)*. Association for Computing Machinery, New York, NY, USA, 55–62. https://doi.org/10.1145/2346496.2346506

[23] Inspirock. 2019. Inspirock. https://www.inspirock.com/

[24] Farah Kamw, AL-Dohuki Shamal, Ye Zhao, Thomas Eynon, David Sheets, Jing Yang, Xinyue Ye, and Wei Chen. 2019. Urban Structure Accessibility Modeling and Visualization for Joint Spatiotemporal Constraints. *IEEE Transactions on Intelligent Transportation Systems* (2019).

[25] Angela Kessell and Christopher Chan. 2006. Castaway: a context-aware task management system. In *CHI'06 extended abstracts on Human factors in computing systems*. ACM, 941–946.

[26] Yohei Kurata and Tatsunori Hara. 2013. CT-planner4: Toward a more user-friendly interactive day-tour planner. In *Information and communication technologies in tourism 2014*. Springer, 73–86.

[27] Rui Borges Lopes, Tiago Coelho, and Beatriz Sousa Santos. 2016. Visually supporting location and routing decisions in tourist trip planning: An exploratory approach. In *2016 20th International Conference Information Visualisation (IV)*. IEEE, 236–241.

[28] Alan M MacEachren. 1980. Travel time as the basis of cognitive distance. *The Professional Geographer* 32, 1 (1980), 30–36.

[29] Abdul Majid, Ling Chen, Gencai Chen, Hamid Turab Mirza, Ibrar Hussain, and John Woodward. 2013. A context-aware personalized travel recommendation system based on geotagged social media data mining. *International Journal of Geographical Information Science* 27, 4 (2013), 662–684. https://doi.org/10.1080/13658816.2012.696649 arXiv:https://doi.org/10.1080/13658816.2012.696649

[30] Daniel Miau and Steven Feiner. 2018. SpaceTokens: Interactive Map Widgets for Location-centric Interactions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 248.

[31] Waze Mobile. 2020. Waze. https://www.waze.com/

[32] Tijs Neutens, Frank Witlox, Nico Van De Weghe, and PH De Maeyer. 2007. Space–time opportunities for multiple agents: a constraint-based approach. *International Journal of Geographical Information Science* 21, 10 (2007), 1061–1076.

[33] Heike Otten, Lennart Hildebrandt, Till Nagel, Marian Dörk, and Boris Müller. 2015. Are there networks in maps? An experimental visualization of personal movement data. In *IEEE VIS*.

[34] John W Payne, James R Bettman, David A Schkade, Norbert Schwarz, and Robin Gregory. 1999. Measuring constructed preferences: Towards a building code. In *Elicitation of preferences*. Springer, 243–275.

[35] Allan Pred. 1977. The choreography of existence: comments on Hägerstrand's time-geography and its usefulness. *Economic geography* 53, 2 (1977), 207–221.

[36] Sergejs Pugacs, Sven Helmer, and Markus Zanker. 2017. A Framework for Comparing Interactive Route Planning Apps in Tourism.. In *EnCHIReS@ EICS*. 18–25.

[37] Richard Schaller and David Elsweiler. 2014. Itinerary recommenders: how do users customize their routes and what can we learn from them?. In *Proceedings of the 5th information interaction in context symposium*. ACM, 185–194.

[38] Felix Schwinger, Fabian Ohler, and Karl-Heinz Krempels. 2018. Flexible Multicriterial Agenda Planning in Public Transit Systems. (2018).

[39] Susan I Stewart and Christine A Vogt. 1999. A case-based approach to understanding vacation planning. *Leisure Sciences* 21, 2 (1999), 79–95.

[40] Alice Thudt, Dominikus Baur, and Sheelagh Carpendale. 2013. Visits: A spatiotemporal visualization of location histories. In *Proceedings of the eurographics conference on visualization*. 79–83.

[41] Tuomas Vaittinen and David McGookin. 2016. Phases of Urban Tourists' Exploratory Navigation: A Field Study. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM, 1111–1122.

[42] Judy Wajcman. 2019. The digital architecture of time management. *Science, Technology, & Human Values* 44, 2 (2019), 315–337.

[43] Di Weng, Heming Zhu, Jie Bao, Yu Zheng, and Yingcai Wu. 2018. Homefinder revisited: finding ideal homes with reachability-centric multi-criteria decision making. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 247.

[44] Tobias Wienken, Cindy Mayas, Stephan Hörold, and Heidi Krömker. 2014. Model of mobility oriented agenda planning. In *International Conference on Human-Computer Interaction*. Springer, 537–544.

[45] Alexandre Yahi, Antoine Chassang, Louis Raynaud, Hugo Duthil, and Duen Horng Polo Chau. 2015. Aurigo: an interactive tour planner for personalized itineraries. In *Proceedings of the 20th international conference on intelligent user interfaces*. ACM, 275–285.

[46] Bin Yang, Chenjuan Guo, Yu Ma, and Christian Jensen. 2015. Toward personalized, context-aware routing. *The VLDB Journal* 24 (02 2015). https://doi.org/10.1007/s00778-015-0378-1

[47] Hyoseok Yoon, Yu Zheng, Xing Xie, and Woontack Woo. 2012. Social itinerary recommendation from user-generated digital trails. *Personal and Ubiquitous Computing - PUC* 16 (06 2012), 1–16. https://doi.org/10.1007/s00779-011-0419-8

[48] Huiyuan Zhou, Aisha Edrah, Bonnie MacKay, and Derek Reilly. 2017. Block Party: Synchronized Planning and Navigation Views for Neighbourhood Expeditions. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 1702–1713.