Notification Agents for Mobile Route Guidance

von Diplom-Informatiker
Markus Schaal

von der Fakultät IV - Elektrotechnik und Informatik
der Technische Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Betreuer:	Dr. Ralf-Detlef Kutsche (TU Berlin)
	Prof. Dr. Hans-Joachim Lenz (FU Berlin)
	Prof. Dr. Herbert Weber (TU Berlin)
Gutachter:	Prof. Dr. Herbert Weber (TU Berlin)
	Prof. Dr. Hans-Joachim Lenz (FU Berlin)

Tag der wissenschaftlichen Aussprache: 4. Mai 2004

Berlin 2004

D 83

Zusammenfassung

Mobile Kommunikation stellt neuartige Herausforderungen an die Bereitstellung, Verwaltung, und Übermittlung von Informationen. Während einerseits ein gewaltiges Überangebot an Informationsbruchstücken existiert, das auch mit fortschrittlichen Technologien für die Suche und Indizierung von Wissen kaum überschaubar gehalten werden kann, fehlen dem Menschen andererseits oft genau die wesentlichen Kerninformationen, um eine bestimmte Aufgabe besser oder gar überhaupt zu bewältigen.

Auf dem entstehenden Gebiet der Informationslogistik werden daher Technologien erforscht und entwickelt, die eine optimale und an individuellen Bedürfnissen orientierte Informationsversorgung ermöglichen. Ein allgegenwärtiges Anwendungsfeld ist hierbei die mobile Reiseunterstützung, insbesondere die Benachrichtigung des Reisenden über alternative und bessere Reisewege, die sich aus unvorhersehbaren Veränderungen der Reisemöglichkeiten ergeben können, beispielsweise durch das Ausfallen bzw. Einsetzen von Zügen.

Der Hauptbeitrag dieser Arbeit besteht in der Konzeption eines Verfahrens für die Auswahl von individuellen Benachrichtigungen, das auch zukünftige Benachrichtigungsmöglichkeiten unter einer dann verbesserten Informationslage berücksichtigt. Zwei konkrete Szenarien demonstrieren die individualisierte und fortlaufende Benachrichtigung des Reisenden über den besten Reiseweg.

Die Unsicherheit über zukünftige Situationen und die Informationsverbesserung durch Abwarten wird quantifiziert und durch probabilistische Modelle dargestellt. Die Topologie des Verkehrsnetzes wird berücksichtigt und dient als strukturelle Vorlage für die Erstellung von Influenzdiagrammen zur Lösung von Benachrichtigungsentscheidungen.

Bei der probabilistischen Routenplanung wird üblicherweise die Route mit dem besten Erwartungswert einer Zielfunktion ausgewählt. Im Gegensatz dazu werden in dieser Arbeit auch zukünftige Benachrichtigungsmöglichkeiten eingeplant und die beste Benachrichtigungstrategie wird gewählt anstatt der besten Route. Dadurch lässt sich ein besserer Erwartungswert der Zielfunktion erreichen als bei den herkömmlichen Verfahren.

Abstract

Ubiquitous and mobile computing has become an important field in computer science. The provision, management and transmission of information within computer systems and for human-computer interaction is required for many applications and the individual needs of the mobile user are manifold. Due to information overflow it is difficult for the user to access specific pieces of information while key information for decision makers is often missing.

Technologies for the emerging field of information logistics are studied in order to enable an optimal information supply with respect to the individual users needs. A dominant application area for this is mobile route guidance, especially when it comes to messaging about alternative routes. Such alternative routes might be advantageous in case of unexpected incidents such as the emergence of an additional train (or cancellation) so that the traveller may reach his/her destination earlier.

The main contribution of this thesis is the conceptualization of a method for the selection of individual notifications for mobile route guidance which considers future notification options and the information quality gain to be achieved. Two scenarios are used for the demonstration of continuous individual messages for the traveller notifying him about the best route to follow.

Explicit models for the uncertainty about future situations are introduced and the uncertainty reduction by waiting for more information is quantified. The topology of the transportation network is considered and used as a structural model for the construction of influence diagrams as decision models. These influence diagrams answer the questions about the best notifications to be sent both with respect to time and content.

Our approach goes beyond probabilistic route planning, which selects a route with maximum expected utility. Other than that, future notification options are considered and the best notification policy is selected rather than the best route. By this, the expected utility can be improved with respect to the standard approach.

Acknowledgements

First of all, I would like to thank Roland Müller for uncounted scientific and personal discussions. Without him, this thesis would not exist. Thanks to Prof. Hans-Joachim Lenz and Dr. Ralf-Detlef Kutsche for continuous supervision and support. Prof. Herbert Weber and the Berlin-Brandenburg Graduate School on Distributed Information Systems (Speaker: Prof. Oliver Günther) provided a good research environment. I would like to thank Marion Fehlker and Marie-Anne Winter (also Aaron and Sarah Winter) for their emotional support. The following people supported me scientifically (and also morally) in various situations (alphabetical order): Dmitri Asonov, Bettina Berendt, Aleksander Binemann-Zdanowicz, Felix Bübl, Susanne Busse, Juliane Dehnert, Lukas Faulstich, Matthias Fritz, Annika Hinze, Marcus Jürgens, Thomas Kämpke, Ulf Leser, Jana Lewerenz, Felix Naumann, Mattis Neiling, Torsten Schlieder, Glenn Shafer, Myra Spiliopoulou, Srinath Srinivasa, Asuman Sünbül, Thiemo Voigt, Nicole Weicker and Karsten Weicker. Many others helped me and aren't mentioned here. Thanks.

Contents

Chapter 1

Introduction

This thesis is about *notification planning.* The application domain of route guidance will be used for motivation, for the study of the subject and for evaluation of results.

Our goal is a better information supply for the traveller. An approach for traveller notification about better routes during his journey is developed in this thesis. Such *notifications* in the domain of personal travel assistance must compete with the traveller's own knowledge about the best route to follow. Consider the following situation:

> On a motorway a congestion is announced. The car navigation system proposes to take another route. This decision is overruled by the driver since he knows another alternative which he can take later.

Thus, an intelligent decision support system must compare current and future *route* alternatives with respect to the *default route.* Future decisions are based upon the information available in the future. Therefore, we develop a model for the relevant *information state* and its evolution in time. An important issue are the two effects of time passing by: on the accessibility of route alternatives (the so-called *loss of options*) and on the improved information for decisions (the so-called *information quality gain*).

Considering these issues, we propose the concept of *notification planning* for the decision about individual *notifications.* The application of this concept leads to the so-called *i-Alert* service:

> The *i-Alert* service is an active and individual information service providing on-trip notifications about routes to follow in order to reach the traveller's goals.

Recent and future developments in mobile technology provide the technological basis for the *i-Alert* service.

The rest of this chapter is structured as follows. First the general idea of a trade-off between information quality gain vs. loss of options will be sketched out (Section 1.1). Next the idea of information logistics will be introduced in Section 1.2 as a framework for implementing active information services. The *i-Alert* service will be described in Section 1.3. Finally an outline of the whole thesis is given in Section 1.5.

1.1 Information Quality Gain vs. Loss of Options

This section describes the trade-off between *information quality gain* and *loss of options* and its implications for notification planning. The *information quality gain* is the improved value of a future information state for taking certain decisions. The *loss of options*, i.e. the loss of route alternatives, is caused by passing junctions without notification. For notification planning, both issues have to be considered.

We employ event trees (cf. Shafer et al. [SGS00]) for comparing alternative chronicles, i.e. sequences of events and actions. Event trees show the partial order of future events. Fig. 1.1 shows a situation of travelling along alternative routes. Events are represented by circles and arrows denote event ordering. Events occurring at logically connected points in time are grouped together by rectangles.

The situation is as follows: A traveller approaches a junction which is called first junction in the sequel. The first event (e_1) is the event of the user's arrival at the first junction. There, the user may either follow the road straight-on (e_2) or he may take a right turn (e_3). Following the road, he will arrive at the second junction (event e_4). From there, he may either make a left turn (e_5) or keep going straight-on (e_6). Finally, the user will reach his destination (events e_7, e_8 and e_9 respectively). The traveller's current default route is labelled by black circles, i.e. the he intends to follow the road straight-on at the current junction and to take a right turn at the next junction. Two alternatives exist:

1. to make the right turn at the first junction
2. to make a left turn at the second junction

However, these decisions are restricted by time and previous decisions, i.e. the first alternative can only be chosen earlier than passing the first junction and the second alternative cannot be chosen when the first alternative has been chosen. Furthermore, the conditions for deciding about the second alternative may be different from the conditions for the decision about the first alternative since time is passing and events may happen between both decisions.

In principle, these effects are well-known in the decision theory literature (cf. e.g. [Bat00]). In particular, the *stopping problem* is a decision problem, where information quality gain is traded vs. loss of options. The stopping problem is usually

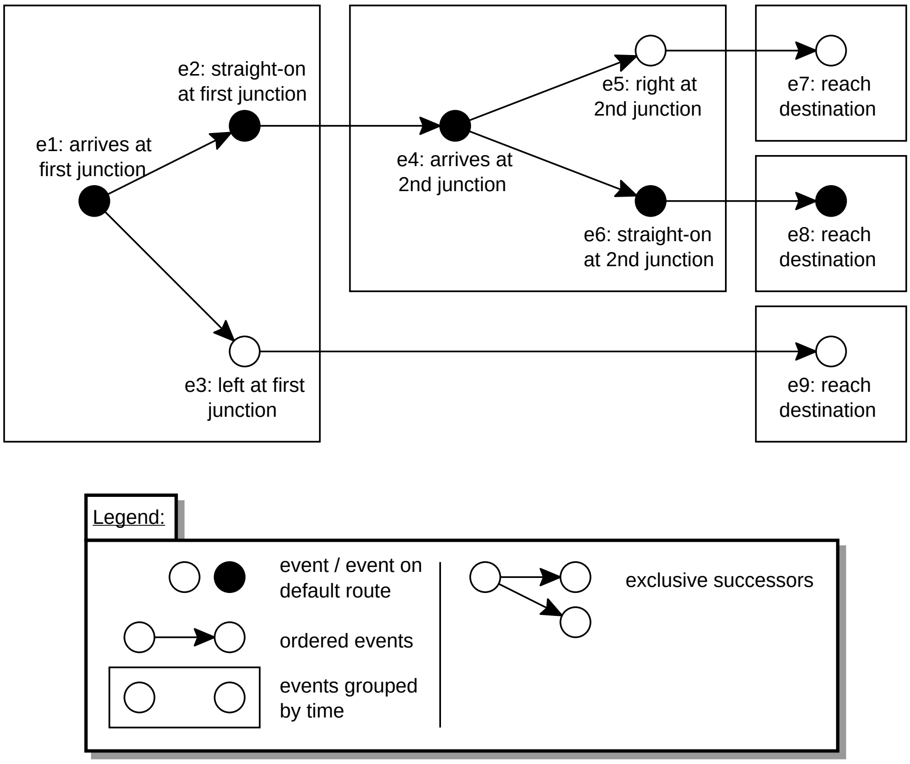


Figure 1.1: Event Tree for the 2-Junction Problem

described as marriage problem or job assignment problem. The job assignment problem is the stopping problem on whether to assign a job to the current job seeker or waiting for the next one to be interviewed. The decision for continuation (not stopping) is based upon the expectation, that a better job candidate can be selected in the future. The information gained about future job candidates by interview can be compared to the information gained about travel time on distinct routes *by waiting*. The loss of future job candidates by stopping can be compared to the loss of routes by the traveller passing junctions (and not notifying him timely about these routes).

Time is an important concept here. The information quality gain obtained between decisions depends both upon the temporal distance (or duration) between these decisions and on the absolute location in time (or time point) of the first decision. Alternatively, the information quality gain between two decisions depends upon the time points of both decisions. The loss of options, i.e. the loss of route alternatives occurs by waiting. The traveller will follow his default route even without an explicit notification.

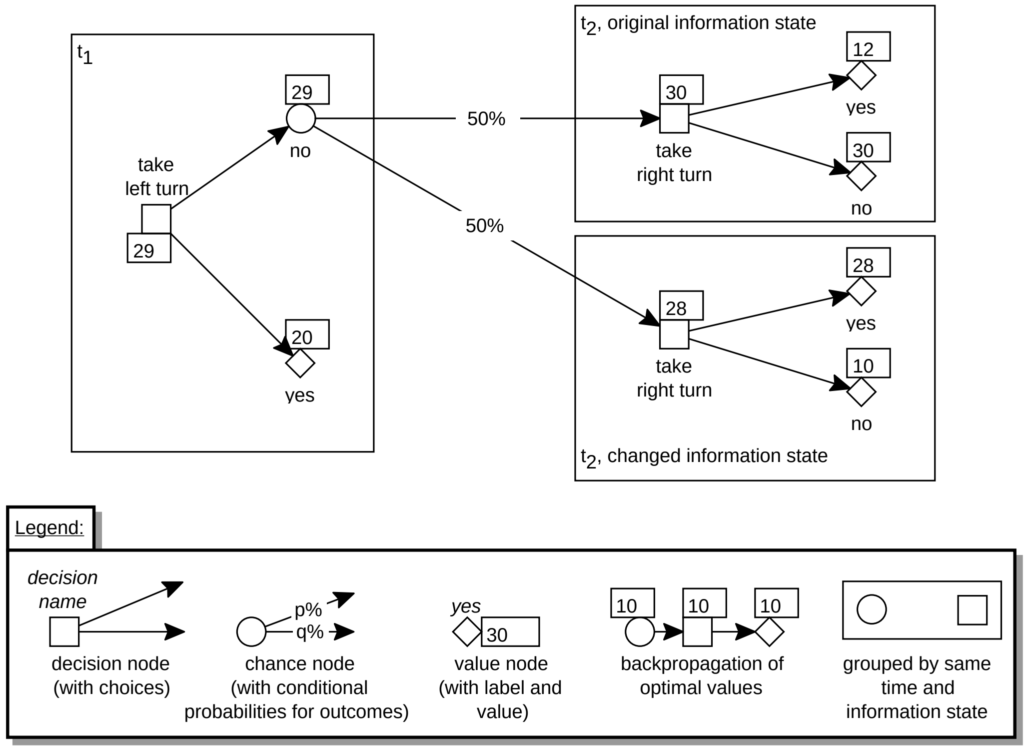


Figure 1.2: Notification with Future Decisions considered

In order to model the information quality gain and the loss of options, the following questions are studied in this thesis:

1. Which options for future notification decisions are to be considered? This depends upon the network topology, travel times and the time points of traveller's route decision.

2. Upon which information are future decisions to be based, i.e. what is the likelihood of certain information states in the future conditioned upon the current state?

3. What is the expected utility (for the traveller) of certain decisions - now and in the future? For this, an utility model is needed.

The decision tree (or game tree) in Fig. 1.2 sketches out the decision about notification. Comprehensive studies of decision trees can be found in Shafer [Sha97] and Pearl [Pea91].

Route decisions are represented here by squares at time points t_1 and t_2. Time point t_1 is the decision point for taking the first alternative, i.e. t_1 is short before arriving at the first junction and enables an explicit decision on whether or not to

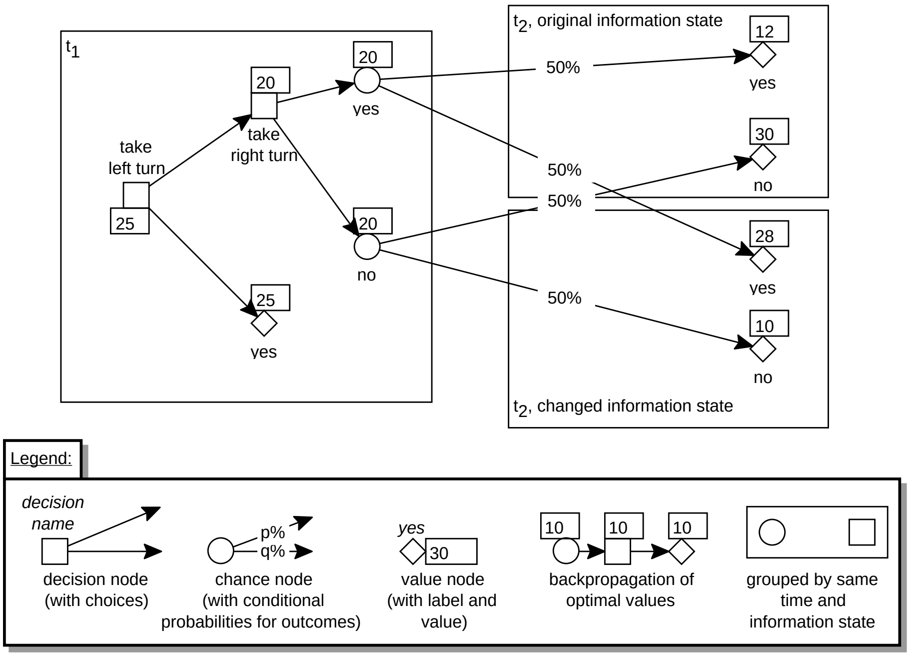


Figure 1.3: Notification without future Options considered

take this alternative. The same holds true for time point t_2 with respect to the second alternative to be taken at the second junction. Time is passing between the decisions. Therefore, a chance node is shown as a circle between the decisions resulting in different information states for the second decision at time t_2. Nodes are grouped here by same occurrence time and same information state.

Each path in the decision tree ends in a value node (shown as a diamond). The value of a value node defines the utility of that path of the decision tree, i.e. of the sequence of route choices and information states. The concrete values shown here are artificial and have been chosen for illustration purposes. Depending on the future information state, the second decision is to be taken differently. Note, that the backpropagated value for following the road straight ahead at the first decision is the expectation of the optimal second decisions value.

This approach of considering the value of future notification options for the decision about current notifications is called *notification planning* in this thesis. The contribution of this thesis is the study of notification planning for route guidance and the development of techniques for treating real-world problems, thereby providing a conceptual framework for the incorporation of notification planning in future information services. The following issues are studied:

- Modelling information states and their evolution.

- Modelling and computation of the decision about current notification.

- Assessment of transition probabilities for the information state.
- Modelling and assessment of the traveller's utility.

The simpler decision about the complete route at current time, i.e. without future decisions considered, is sketched out in Fig. 1.3. Here, the second decision is assumed to be taken at the current time t_1, i.e. without improved information. In this case, the utility of the current decision about the complete route is simply the expectation of the utility with respect to future information states. The best choice here is to take the right turn at the first junction. With this, the second decision cannot be made and the option for taking an alternative route at the second junction is lost.

1.2 Information Logistics

Information logistics[1], is a rapidly emerging methodology for next generation information systems. The term *information logistics* both characterizes a new architecture for information systems and a new class of services to be implemented within such information systems. The technological change which is responsible for the emergence of information logistics is the availability of mobile devices such as mobile phones and connected organizers.

The Short Message Service (SMS) provides a basis for information services with notifications sent to the user at any time. Generally, the ability to communicate at any time and any place by use of different devices has led to the definition of the following so-called information logistics dimensions:

- *Time:* The user can be notified at any time. Thus, the best time for notification can be selected with respect to the availability and cost of communication channels such as SMS, FAX, email and video streaming. Content can be selected with respect to the time of notification. User-oriented aspects (situation-specific reception, times of unavailability, etc.) can be considered.
- *Location:* The user can be notified at any place (not only at home or at work). Therefore, the content can be selected according to his spatial location context; other contextual information may be considered as well.
- *Content:* The content can be customized with respect to notification time and place. In fact, the actual content does not need to be created until shortly before notification time, i.e. the information can be created just-in-time. A similar approach has been described for information agents (cf. Rhodes [Rho00]).

[1] The term *information logistics* has been coined at the Fraunhofer Institut für Software- und Systemtechnik (ISST), cf. Deiters and Lienemann [DL01] and Deiters et al. [DLP03].

- *Mode of Transmission:* The mode of transmission can be *push*, i.e. the information system plays an active role in the communication between information system and user. Previously, communication steps were usually triggered by the user (*pull* or *poll*). An earlier approach for shifting the initiative from the information system towards the user has been known as Mixed-Initiative-Assistance (cf. Ferguson et al. [FAM96] and Horvitz [Hor99]).

- *Presentation:* Last but not least, the presentation of the content has to be adapted to the technical features of the presentation device. Ideally, the content can be described independently from the device and the presentation is adapted automatically. [Lew00] gives a thorough discussion of the topic and provides many valuable solutions.

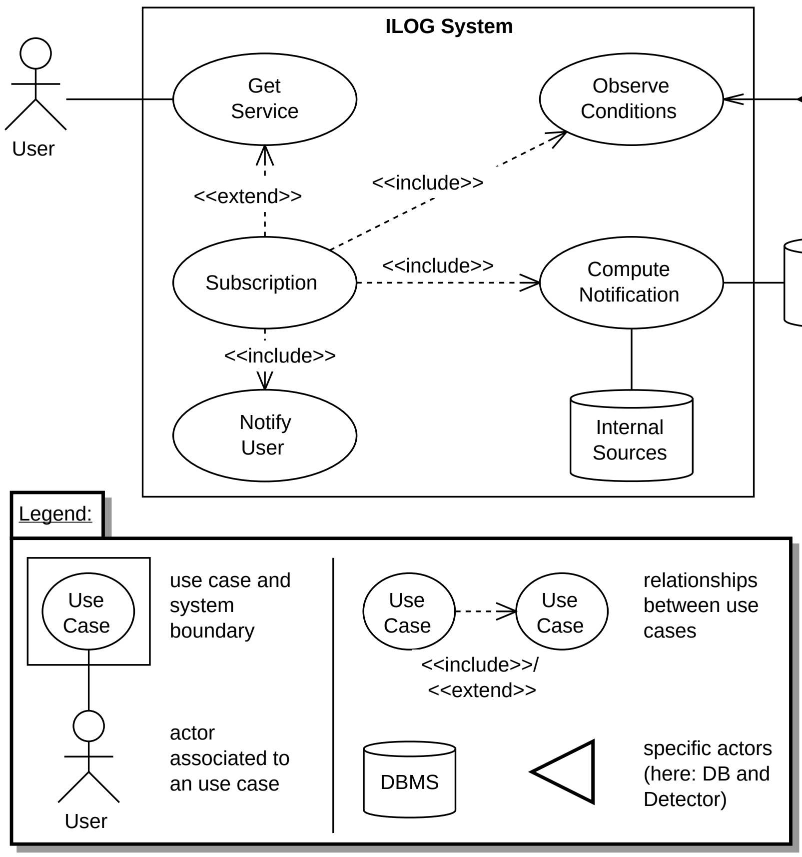


Figure 1.4: Use Case: An Information Logistics System

Evidently, the creation of Information LOGistics (ILOG) services does not only generate a new quality of information for the user, it also creates a big challenge for computer science to provide the technological framework for the implementation of such services. This thesis is a response to this challenge, focussing on providing the *right content* at the *right time and place* for mobile route guidance.

An ILOG system can be thought of as being a combination of an active database (for information model, activation rules and activation), a user notification system (for localization of users, identification of communication channels and message transmission), a decision/ planning system (for the decision about time and content of notifications) and a definition system (for generation of activation rules from users information demand). A use case diagram for such a system is depicted in Fig. 1.4.

The *user* gets a service from the ILOG system (use case *Get Service*). If continuous tracking of events is required for the service, the service is extended and becomes a *Subscription*. A subscription includes the observation of external conditions (*Observe Conditions*), the computation of the appropriate content (*Compute Notification*) and the delivery of the content to the user (*Notify User*). The computation of the content is done with the help of external information sources and the internal knowledge base.

1.3 The *i-Alert* Service

The *i-Alert* service uses information about the time and the location of the traveller in order to provide notifications with the right time and content. By this, the *i-Alert* service is a typical information logistics service.

For the traveller, the *i-Alert* service is an extension of route planning. After planning an initial route, the traveller may choose to get on-trip route guidance for a selected route. The *i-Alert* service observes relevant events and informs the traveller about alternative routes. This will be illustrated by the following scenario.

Scenario 1 *Mr. Miller (a businessman) is on the way to a meeting which is located in Potsdam (near Berlin) and scheduled for* $3:00$ *pm. He is sitting in a train from Hamburg to Berlin Zoologischer Garten (a railway station) and has the following options for reaching his final destination after getting of in Berlin Zoologischer Garten:*

- *to use an ICE (a superfast train) and to get off in Potsdam.*
- *to use a local train to get to Potsdam.*
- *to use a taxi to get to Potsdam.*

These alternatives are called ICE, train *and* taxi *respectively. Mr. Miller plans to take the ICE, since this is the fastest connection according to pre-trip route planning. However, unexpected events such as train delays or bad weather conditions can change the optimal route.*

Mr. Miller has a personal organizer with a wireless connection, i.e. he can use internet services while travelling and may receive messages at any time. Suddenly, his organizer beeps and displays a message:

```
You may improve Your chance of timely arrival in Potsdam
by taking a taxi from Berlin Zoologischer Garten instead of
the ICE. Please confirm.
(This service is brought to You by i-Alert )
```

Mr. Miller confirms. A short time later, the ICE arrives in Berlin and Mr. Miller takes a taxi to Potsdam. The taxi driver is smiling. Today, he has many passengers due to construction work and irregular railway traffic between Berlin and Potsdam.

In order to act like this, the information system (which implements the *i-Alert* service) needs to know about the route of the traveller (e.g. *plans to take the ICE*), about route alternatives (e.g. *train* and *taxi*) and about the changing situation in the transportation network (e.g. *construction work* and actual delays).

Fig. 1.5 illustrates a sequence of messages and stimuli that may have led to the *i-Alert* notification of the scenario.

After planning his meeting and the time constraints for timely arrival in Potsdam (*planTrip*), Mr. Miller asks the *i-Alert* service for the best route to use from Hamburg to Potsdam with arrival time 3 pm (*requestRoute*). The Information System (IS) chooses the ICE and informs Mr. Miller accordingly and asks him for confirmation (*returnRoute*). He confirms (*confirm*) which causes the *i-Alert* system to register a subscription for the whole trip from Hamburg to Potsdam. Incoming events are processed but not every event causes a notification. Two examples are depicted here. The plane delay (*event(plane late)*) is ignored for obvious reasons while the ICE delay (*event(ICE late)*) could have been relevant for our traveller even though it isn't. Finally, the train delay *event(train late)*) causes the information system to notify Mr. Miller accordingly (*notifyRoute*). Mr. Miller confirms again (*confirm*) thereby telling the *i-Alert* system about a route change. Finally, when Mr. Miller reached his final destination, the subscription gets unregistered (*unregister*).

The *i-Alert* service combines numerous features which cannot be found in industrial applications. These features are discussed here:

- The *i-Alert* service is provided on-trip (as opposed to pre-trip or post-trip[2]). While countless applications[3] for pre-trip route planning exist, there is only a

[2] Route planning and electronic reservation and ticketing are typical examples for pre-trip services, while service evaluation and the clearing of travel expenses are post-trip services. Car navigation systems provide on-trip services.

[3] Some existing solutions for public transport: HAFAS (`http://bahn.hafas.de`), EFA (`http://www.efa.de`) and fahrinfo (`http://www.bvg.de/plan/fahrinfo.html`)

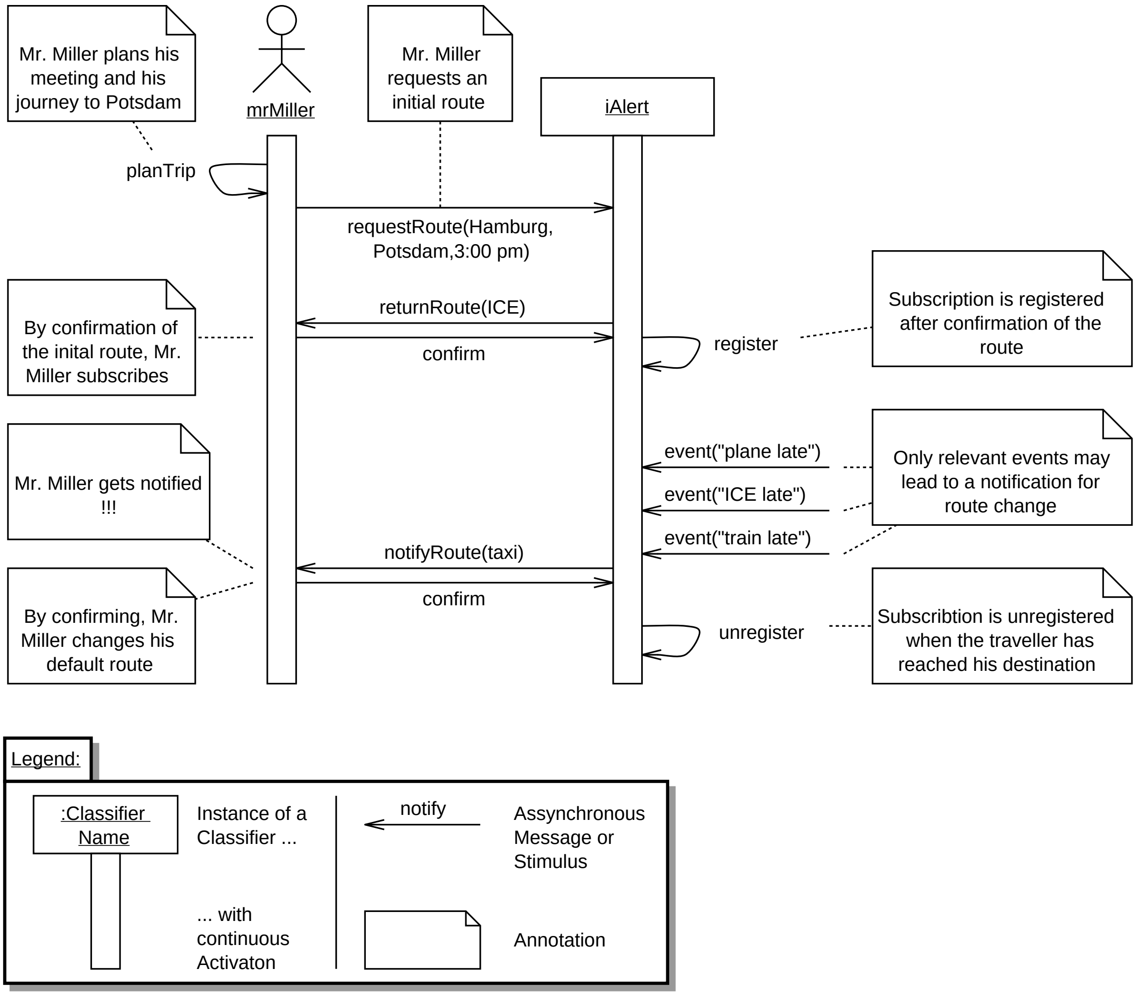


Figure 1.5: Sequence Diagram for Scenario 1

few applications available for on-trip route planning. Navigation systems for cars are an example which is widely known. A major difficulty with on-trip route planning is the unavailability of real-time data.

- The *i-Alert* service integrates data from both individual and public transportation. This is usually not found in existing applications. It is very difficult to integrate route information stemming from different transportation providers, mainly because these providers want to keep their informational autonomy.

- The *i-Alert* service builds on *notification planning*, a specific method developed in this thesis for the consideration of future notification options in addition to the current ones.

Currently, the *i-Alert* service is a vision even though the technology for the implementation of notification services exists. This thesis provides a theory for notification planning, i.e. the reasoning about the right time and content for notification. However, other obstacles exist. Despite reality, we will assume to have unlimited access to real-time transport information, i.e. the existence of event notification services for train delays, congestions, etc. is assumed across all organizations providing transport information.

A few words on *roles*, *systems* and *services* for clarification:

- We will talk about the *user* of a system and about the *traveller*. Both are *role names*. In this thesis, a *user* is always a *traveller* and vice versa.
- Unless otherwise explicitly stated, the terms *information system*, *i-Alert system* and *ILOG system* will always refer to an *i-Alert* system, i.e. to an *information system* that offers the *i-Alert* service.

1.4 Scenarios

In this thesis, we will use two scenarios as running examples. These scenarios have been studied and published earlier (cf. Schaal and Lenz [SL01] and [SL03]). The *road scenario* is an example for individual transportation, where departure times are flexible. The *train scenario* is an example for scheduled transportation, where departure times are scheduled and specified by time tables. Further differences and specialities will be discussed below.

1.4.1 Road Scenario

Let us assume a traveller who travels from location a and is heading towards location g. He has three alternative routes A, B, and C, as shown in Fig. 1.6. Each of the routes consists of road segments and passes different locations b-f, e.g. route A passes through locations b, c, e and consists of road segments (a, b), (b, c), (c, e) and (e, g). The condition of road segments (c, e), (d, e) and (d, f) is represented by state values kept by the information system. We consider state values f (*free*) and c (*congested*) here. The state of the road segments changes over time. The traveller follows route C as default route unless notified otherwise.

Based on current state values, the following question is to be answered: Is it appropriate to notify the traveller at the current time about an alternative route to be taken? To support this decision, secondary questions arise: What is the utility of notifying the traveller at some later time, e.g. supporting the choice between route B and C at location d? For measuring the utility of future notifications, additional questions are to be answered: Given the state on the segments (c, e), (d, e), and (d, f) at current time t_{curr}, what is the likelihood for a free road segment at a future time $t' > t_{curr}$?

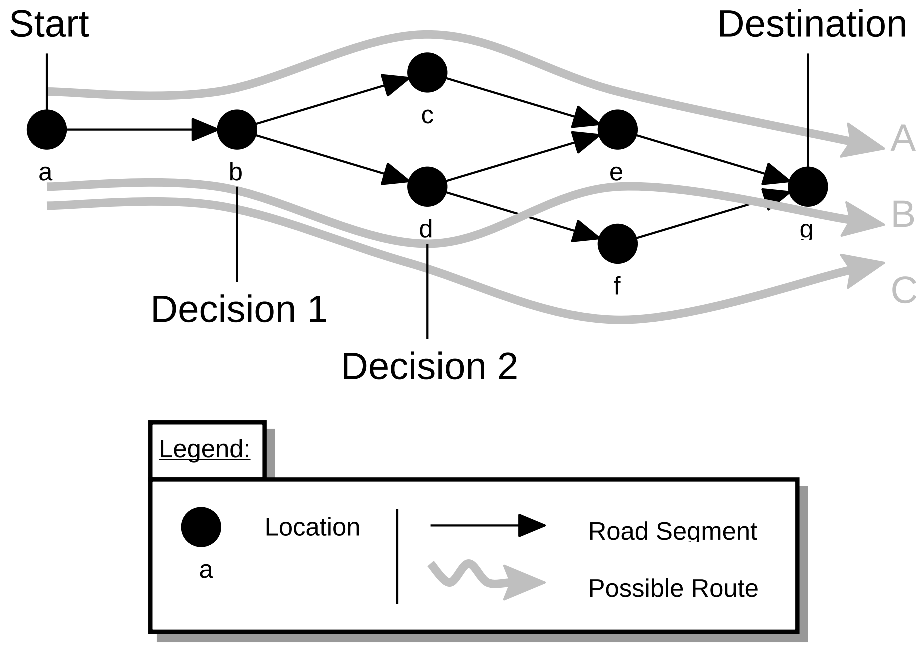


Figure 1.6: Route Choices for Road Scenario

1.4.2 Train Scenario

A traveller can reach his destination on two distinct routes (cf. Fig. 1.7). The first route (by train) is expected to be faster than the second route (by taxi), but the taxi is usually available without waiting, while the train may depart late, which may cause a delay for the traveller. An eventual delay of the train is given by state values for the train, either *delayed* or *unknown*. This information state about an eventual train delay changes over time.

The traveller starts at location a with unknown default route. This is modelled by assuming equal likelihood for both route selections. The traveller may change his default route until he reaches location b where both routes depart.

The information system sends one notification to the traveller at the most. The traveller wants to meet a certain deadline (start time of a meeting) and the probability for this is to be maximized.

1.4.3 Scenario Classification

The two scenarios given in Section 1.4.1-1.4.2 represent the problems studied in this thesis. For both scenarios, future notification options are to be considered and the information state at future decision time points is to be modelled. The road scenario features a known initial route and two exclusive notification options for two different

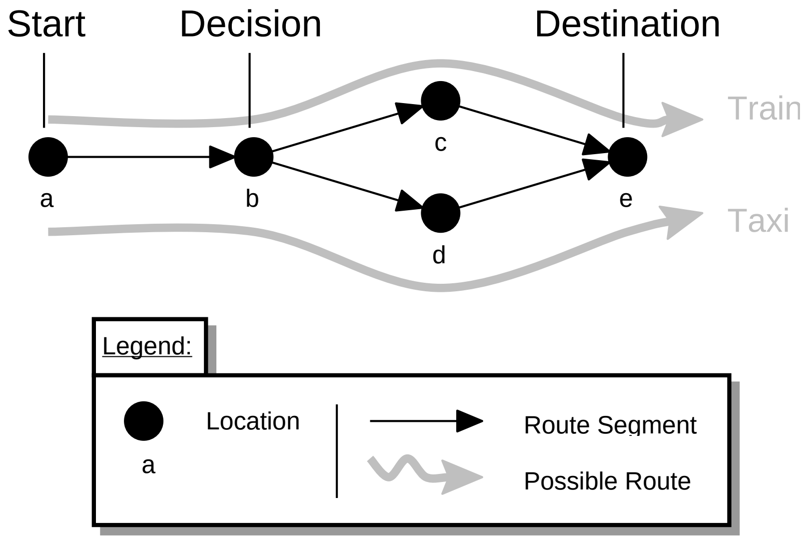


Figure 1.7: Route Choices for Train Scenario

decision points. The train scenario is restricted to one decision point but it features the selection of an appropriate notification time point and uncertainty about the initial route of the traveller.

Notification Planning is based upon the following models, which will be used for scenario classification:

- The *transportation network* models the possible routes of the traveller.
- The *default route* models the route to be followed by the traveller in the future. The default route is either deterministic or probabilistic, i.e. a probability can be assigned for different routes instead of considering a specific default route only.
- The *traveller's location* models the current location of the traveller.
- The *notification effect* models the impact on the default route caused by notifications which are (to be) sent to the user.
- The *information state* models the current knowledge about the situation on the transportation network.

	Road Scenario	Train Scenario
default route	deterministic	probabilistic
traveller location	yes	yes
notification effect	direct	based on timeliness
transportation network	individual	public/ scheduled
information state	*free* or *congested* per road segment	*none* or *delayed* per train

Table 1.1: Scenario Classification

The specialities of both scenarios with respect to these model parts are discussed below and shown in Table 1.1.

- The *default route* is the route according to which the traveller will move in the future unless notified to take another route. In absence of a default route, the traveller will follow distinct routes with certain probabilities. The *road scenario* assumes the existence of a *deterministic* default route, i.e. the route to be taken by the traveller is known before he departs. The *train scenario* contains uncertainty about the traveller's default route, i.e. a *probabilistic* default route is given.

- The *traveller's location* is used in both scenarios.

- The *notification effect* is the influence of a notification on the default route. In the road scenario, it is assumed that this influence is direct, i.e. a notification simply alters the deterministic default route. In the train scenario, the influence of a notification depends on the timeliness of the notification, i.e. whether the notification occurs prior to the time point of the traveller's choice. Since travel time is probabilistic, the notification effect is probabilistic as well. Specifically, the influence of a notification on the traveller's default route decreases while he is approaching a junction.

- The *transportation network* is essential for the estimation of travel times. The road scenario represents the case of individual transport while the train scenario represents the (scheduled) public transport. The major difference between both is the modelling of the information state. In the road scenario, one variable per route segment is sufficient, while the train scenario requires one variable per scheduled vehicle.

- The *information state* changes in time. In the road scenario, we consider two different state values, i.e. *free* and *congested* per road segment. In the train scenario, we consider two different state values as well, namely *none* and *delayed* per train. Note, that the information state at current time is deterministic, i.e. it has an unambiguous current value.

1.5 Thesis Outline

Important concepts both from the application domain (e.g. transportation networks, travel time, etc.) and from the applied methods (e.g. decision trees, random variables, etc.) are introduced in Chapter 2.

Two prerequisites for the implementation of the proposed notification agents, namely the distributed fastest paths generation and the estimation of transition probabilities between information states, are discussed in Chapter 3.

The transformation of a specific problem into a decision problem is performed by employing influence diagrams as a compact representation, cf. Chapter 4.

The thesis is complemented by a concept for embedding notification planning into the currently emerging ILOG systems, cf. Chapter 5.

Chapter 2

Conceptualization

In this chapter, the concepts required for notification planning are identified and described and models for these concepts are specified.

The identification and description of concepts is undertaken here by means of prose, glossary entries and Unified Modelling Language (UML) [UML01]. The specification of a model is undertaken here by the definition of mathematical models, in some cases complemented with graphical models (cf. Appendix D for an overview on graphical modelling techniques used here).

2.1 Overview of Concepts

This section gives an overview for the identified concepts and thus provides the agenda for this chapter. By studying the scenarios provided in Section 1.4 the following basic concepts have been identified (with short descriptions):

- the *traveller* (or: user), who moves through a transportation network and uses the information system.
- the *transportation network*, which serves as a frame of reference for describing both the travellers location and the routes for travelling (change of location, movement).
- the movement of the traveller without notification, modelled by the so-called *default route*.
- the user's goals, modelled by a *utility function* for arrival time.
- the *notification* sent to the traveller and the *notification effect*, i.e. the result of sending a notification to the traveller.

We want to choose the right notifications in order to meet the travellers preferences. However, the state of the transportation network changes over time: congestions appear and dissolve, it may start to rain, etc. Therefore the following advanced concepts are also needed:

- the *information state* which represents the state of the transportation network.
- the information state evolution (represented by *transition probabilities*), i.e. a description of the changes of the information state while time passes by.
- the *time- and state-dependent travel time* along routes in the transportation network.

All of these concepts will be described within this chapter and formal models will be provided.

2.2 Basic Notations and Graph Theory

The following conventions for algebraic representations will be used:

- Both *entities* and *variables* (or entity placeholders) are denoted by lower case letters, e.g. a location l, a decision point dp or an event occurrence e.
- The *domain* of a variable is either given extensionally by a *set* or intensionally by a *class*. Both sets and classes are denoted by upper case letters, e.g. a set of locations L, an event class E, a set of decision points DP, etc.
- The membership of an instance or a variable is denoted by the $\in$-Symbol even for class-membership, e.g. a location $l \in L$ or an event occurrence $e \in E$.
- random variables are denoted by boldface letters, e.g. probabilistic duration $\mathbf{d}$, probabilistic arrival time $\mathbf{t}$, random information state $\mathbf{s}$, etc.
  - $E(\mathbf{x})$ is the expectation of the random variable $\mathbf{x}$, $E(\mathbf{x} \mid \mathbf{y} = y)$ is the expectation of the random variable $\mathbf{x}$ conditioned on the outcome y for the random variable $\mathbf{y}$.
  - $P(\mathbf{x} = x)$ is the probability of $\mathbf{x}$ to have outcome x. $P(\mathbf{x} = x \mid \mathbf{y} = y)$ is the conditional probability of $\mathbf{x}$ to have outcome x if $\mathbf{y}$ has outcome y.
- sets are enclosed by curly brackets, e.g. route set $R = \{r_1, r_2, r_3\}$ while sequences and tuples are enclosed by round brackets, e.g. route $r = (l_1, \ldots, l_n)$.
- instances are denoted as tuples and their parts are referenced by the dot-notation, e.g. a notification n consisting of time point t and route r is denoted as $n = (t, r)$ with time point $n.t$ and route $n.r$.

The notion of graphs and trees is widely used in graph theory (cf. e.g. Lawler [Law76]). We will give the definitions of graphs and trees here for convenience (transportation networks and decision trees are based on graphs and trees respectively).

Definition 2.1 (Graph) *A graph (V, E) consists of a finite set V of vertices and a set $E \subseteq V \times V$ of edges.*

Definition 2.2 (Path) *Let $G = (V, E)$ be a graph. A sequence $(v_1, \ldots, v_n)$ is a path in G if $(v_i, v_{i+1}) \in E$ for $1 \leq i < n$.*

Definition 2.3 (Connected Graph) *A graph $G = (V, E)$ is connected if for any two vertices $v_1, v_2 \in V$ there is a path $(v_1, \ldots, v_n)$ in G.*

Connected graphs are needed for the definition of transportation networks (cf. Section 2.4).

Definition 2.4 (Cycle) *Let $G = (V, E)$ be a graph. A path $(v_1, \ldots, v_n)$ in G is a cycle, if $v_1 = v_n$.*

Definition 2.5 (Acyclic Graph) *A graph (V, E) is acyclic if it has no cycles.*

In an acyclic graph, the successors of a vertex are called children, the predecessors are called parents of the respective vertex. The parents of a vertex v are denoted by $Par(v)$, the children are denoted by $Childs(v)$.

Definition 2.6 (Root Vertex) *Let $G = (V, E)$ be an acyclic graph. A vertex $v \in V$ is a root vertex, if $Par(v) = \emptyset$.*

Definition 2.7 (Tree) *A tree (V, E) is an acyclic graph with exactly one root vertex and one parent for all other vertices.*

Trees are needed for the definition of decision trees (cf. Section 2.7).

2.3 Time

Reasoning about time plays a major role for notification planning in the domain of route guidance. Not only the description of the traveller's past, present and future movements, but also the description of signals received and actions to be planned by the information system require a time model for reference.

Dates of events and temporal distances between events are to be modelled. Some examples for temporal specifications found in the scenarios are given next:

- *Time Point*, e.g. time points of departure and arrival (at a location), time point of a train departure or a train arrival, time point of a notification, time point of an event, etc.
- *Interval*, e.g. time interval during which a specific condition is true.
- *Duration*, e.g. duration for travelling from location A to location B.

Time points have no duration. Time intervals consist of two time points, the first being the start time or left side of the time interval, the second being the end of the time interval. A duration is the length of a time interval.

2.3.1 Information Model

A *temporal specification* is an explicit reference to a time point, an interval or to a duration. An information model for *temporal specifications* is given in Fig. 2.1. This model builds on previous information models for time developed at Computergestützte Informationssysteme, TU Berlin (CIS) (cf. Kutsche et al. [KSWLM95], Busse et al. [BKS97] and Busse et al. [BK00]).

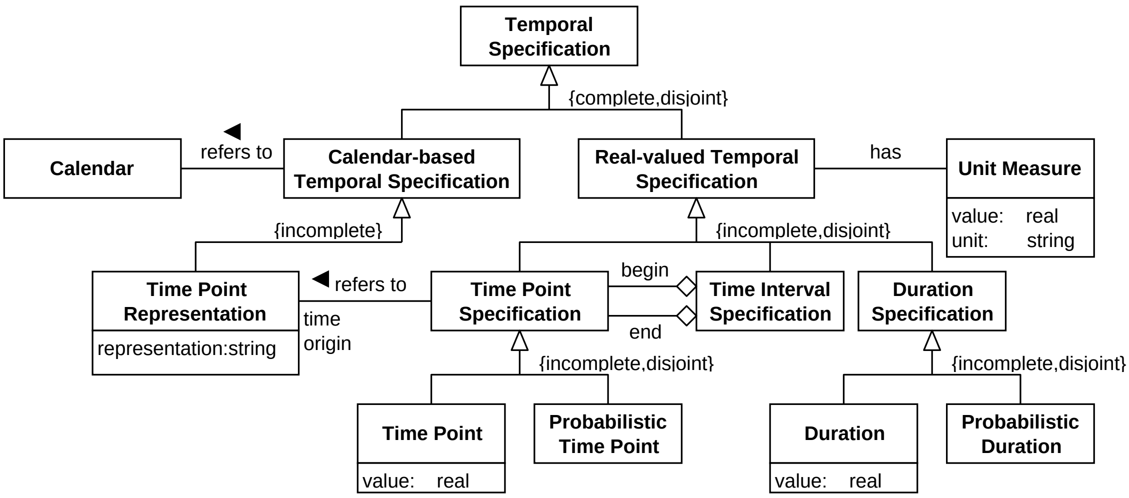


Figure 2.1: Information Model for Temporal Specifications

Temporal specifications might be imprecise or uncertain and time points might have an intrinsic granularity. Thus, *granularity*, *precision* and *uncertainty* are optional attributes of temporal specifications:

- *Granularity:* A time point is often given with respect to a calendar and thus has an intrinsic granularity such as day, hour, minute, or second.

- *Precision:* An imprecise time point can be represented by a time point together with information about its precision. Precision can be given as maximal deviation (a duration).

- *Uncertainty:* A time point or a duration can be uncertain. In this case, the respective temporal specification is given by a probability distribution over a set of temporal specifications of the respective type, e.g. an uncertain time point is given by a probability distribution over a set of certain time points.

We will omit the issues of granularity and precision in this thesis. Time points are point-like here, i.e. they do not have an extension in time. Temporal specifications can be uncertain, but probability distributions are give over sets of precise temporal specifications.

We distinguish *Calendar-based Temporal Specification* and *Real-valued Temporal Specification* as specializations of *Temporal Specification*.

Calendar-based temporal specifications refer to a *Calendar*. The *Time Point Representation* is the only specialization considered here. The *representation* of a *Time Point Representation* is typed as a string, e.g. its ISO 8601[1] representation. A *Time Point Representation* has an implicit granularity. For example, *1999-01-01* (First of January, 1999) has granularity *day*, while *1999-01-01 12:33* (12:33 on the same day) has granularity *minute*. Since we ignore this granularity, all time point representations are identified with the start of the time interval implicitly given by the chosen granularity, e.g. *1999-01-01* is the same as *1999-01-01 00:00:00*.

Real-valued temporal specifications have a *Unit Measure*. The *Unit Measure* is given by real-typed value and unit (e.g. *second*, *minute*, *hour*, etc.). We distinguish *Time Point Specification*, *Time Interval Specification* and *Duration Specification* as real-valued temporal specifications. A *Time Point Specification* is a generic type for specializations *Time Point* and *Probabilistic Time Point*, while *Duration Specification* is a generic type for specializations *Duration* and *Probabilistic Duration*. A *Time Point Specification* refers to a *Time Point Representation* which specifies the *time origin* for its translation into a *Time Point Representation*, i.e. the time point specification is anchored at the time origin. The *value* of a Time Point is typed by the real numbers $\mathbb{R}$. For example, a *Time Point* with value *10*, unit measure *1 minute* and time origin *1999-01-01 12:33* specifies the time point which is exactly 10 minutes after 12:33 on January 1st, 1999. A *Probabilistic Time Point* is the representation of an uncertain time point and will be given by a probability measure. Details will be discussed below.

The value of a *Duration* is typed by the set of real numbers $\mathbb{R}$, i.e. for instance a duration of 30 seconds can be modelled with value *30* and unit measure *1 second* but also with value *0.5* and unit measure *1 minute*. A *Probabilistic Duration* is the

[1]ISO 8601 is an international standard for the representation of date and time.

representation of a uncertain or imprecise duration and will be given by a probability measure. Details will be discussed below.

For reasoning about time, we will use the real-typed values of durations and time points together with their probabilistic variants. Within the same context, time points and durations will have the same time origin and granularity without further notice. The following excerpt from the glossary describes the time concepts from the time model that will be used.

Time Point A *time point* (notation: t) has a value, a time origin and a unit measure. The value of a *time point* is typed by the real numbers $\mathbb{R}$. A *time point* is related to a real-world time point by specification of time origin and unit measure. The value of the time point is the measurement of the difference between the time origin and the real-world time point to be modelled. Within a context, time origin and unit measure are omitted and time points are identified by their value, e.g. time point $t = 23$.

Duration A *duration* (notation: d) has a value and a unit measure. The value of a *duration* is typed by the real numbers $\mathbb{R}$. The unsigned difference between any two time points is a duration. Within a context, the unit measure is omitted and durations are identified with their value, e.g. duration $d = 3$.

Time Interval The ordered set of the values of the time points in a closed *time interval* (notation $[t_1, t_2]$) is an interval of real numbers with t_1 being the value of the beginning time point and t_2 being the value of the end time point.

Probabilistic Time Point A *probabilistic time point* (notation: boldface $\mathbf{t}$) is given by a probability space $(T, \mathcal{T}, P)$ where $T \subseteq \mathbb{R}$ is a finite set of time points, a σ-algebra $\mathcal{T}$ is the power set of T and the probability measure $P : \mathcal{T} \to [0, 1]$.

Probabilistic Duration A *probabilistic duration* (notation: boldface $\mathbf{d}$) is given by a probability space $(D, \mathcal{D}, P)$ where $D \subseteq \mathbb{R}$ is a finite set of durations, σ-algebra $\mathcal{D}$ is the power set of D and the probability measure $P : \mathcal{D} \to [0, 1]$.

2.3.2 Other Approaches

While the information model in the previous section is sufficient for our purposes, other time models have been proposed for automated reasoning about time.

- *Interval Arithmetic for Time:* It has been pointed out by Allen, that real-world time points rarely correspond to points in time rather than to time intervals. Therefore, he developed his well-known interval arithmetic for time. In Allen [All83], a time point is viewed as an interval and 13 cases have to be distinguished for the relationship between two time point (intervals).

- *Discrete Time:* With discrete time, time is isomorph to the natural numbers. Most approaches for discrete time introduce the concept of a *chronon* as the smallest piece of time. Bei this, each time point can be assigned to a specific chronon which is represented by its count (a natural number).

- *Branching Time:* Other than linear time, time can be modelled as a partially ordered set, thus allowing alternative futures and pasts to be represented. This is applicable for symbolic models of time, where time points are not viewed as being isomorphic to the real or natural numbers.

- *Temporal Logic:* Temporal logic provides predicates such as *next*, *sometime*, *eventually in the future* for the description of the time-dependent truth-values of prepositions.

An overview on discrete time models, branching time models, and temporal logics (with further references) is given in Özsoyoğlu and Snodgrass [OS95].

2.4 Transportation Networks

The movement of the traveller is described in terms of transitions between locations, e.g. Mr. Miller moves from location a to location b. The means of transportation is not of interest here, it can be anything from walking to flying. Instead, we are only interested in the travel time between locations. The travel time may depend upon certain time-dependent parameters such as congestions and weather conditions. The travel-time also depends on the time of departure, especially in the case of scheduled transportation, where the waiting time for the next train is to be considered.

Definition 2.8 (Transportation Network) *A transportation network (L, E) is a connected graph consisting of a location set L and an edge set $E \subseteq L \times L$.*

The condition of a transportation network (e.g. congestions, weather, etc.) is modelled by the so-called information state (cf. Section 2.5). The information state evolves in the course of time. The information state is the information system's base for predicting travel times (cf. Section 2.8).

2.4.1 Locations

In the context of mobile route guidance, locations occur in many different ways. They are the points of departure and arrival, they are used for the description of routes, travel times are expressed in terms of temporal distance between locations and last but not least, the travellers location is to be tracked.

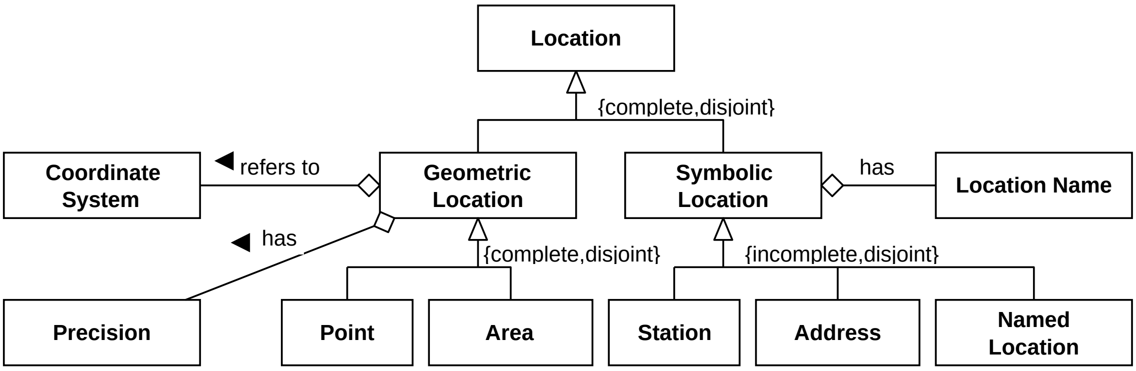


Figure 2.2: Information Model for Locations

Here, a *location* is a point in space or a connected subset of the space, where a traveller can be located. An information model for locations is given in Fig. 2.2 and described next.

Leonhardt [Leo98] distinguishes symbolic and geometric location models. Accordingly, we distinguish *symbolic locations* and *geometric locations*. A symbolic location is given by a unique *location name* while a geometric location is specified by coordinates with respect to some *coordinate system*.

A geometric location is either a *point* (without extension) or an *area*. Geometric locations are useful for the interpretation of positioning data (e.g. stemming from a GPS-sensor) and for displaying them on a map. A geometric location has a *precision* which gives the accuracy of its coordinates. Geometric locations are not studied here.

Generally, we will use symbolic locations as nodes in a transportation network. A symbolic location is an *address*, a (public transport) *station* or any other *named location*, e.g. a platform within a station.

The travellers current location is an important input for notification planning. The symbolic location of the traveller can be derived by two distinct methods:

1. The travellers symbolic location can be derived from the initial default route and from the travel time combined with the sequence of events along the route.

2. The travellers geometric location can be detected by a location sensor (e.g. by GPS) and subsequently transformed into a symbolic location.

The detection of the traveller's geometric location is hidden away from the application. A location model and context component is under development which will be able to detect the symbolic location of a traveller (cf. Haseloff [Has01] and pending German patent no. 102 01 859.6). Even if this kind of positioning fails, the symbolic location of the traveller can be estimated directly from a model.

Example 1 (Locations) *Mr. Miller travels from his home in Berlin to Stuttgart Hauptbahnhof (locations are* emphasized*): He departs from his home address (*Müllerstr. 45*) and arrives at underground station* U Leopoldplatz*. Then, he takes the underground from* U Leopoldplatz *to* U Zoologischer Garten*. Mr. Miller changes from* U Zoologischer Garten *to* DB Zoologischer Garten, platform no. 9*. Subsequently, a superfast train starts from* DB Zoologischer Garten, platform no. 9 *and arrives at* DB Hauptbahnhof Stuttgart*.*

With Example 1, each emphasized location can be modelled as an element of the location set L. Each location is a symbolic location, i.e. an address (e.g. *Müllerstr. 45*), a public transport station (e.g. *U Leopoldplatz*) or a named location (e.g. *DB Zoologischer Garten, platform no. 9*).

Relationships between any two locations like *identical*, *overlapping*, *contained within each other* or *non-overlapping* and the issue of the granularity of a location are not considered here. However, both the locations represented by the nodes of the transportation network and the interconnecting edges have to be carefully selected for route planning. This will be illustrated by an example with different levels of granularity.

Example 2 (Mixed Granularities) *Let transportation network* (L, E) *with location set* $L = \{m, b, s, b1, b2\}$ *and edge set* $E = \{(m, b), (b, s), (m, b1), (b1, b2), (b2, s)\}$ *be given. Location symbols* m*,* b*,* s *refer to Munich, Berlin and Stockholm railway station, respectively. Location symbols* $b1$ *and* $b2$ *refer to platforms* 1 *and* 2 *within Berlin railway station. The situation is depicted in Fig. 2.3 (with containment between locations informally sketched). Here, edges connect Munich (m) with Berlin*

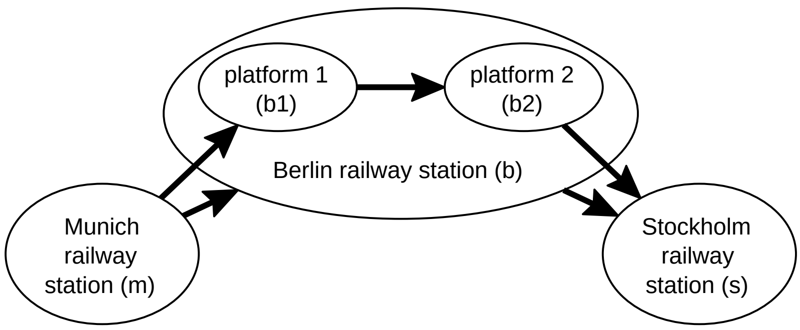


Figure 2.3: Transportation Network with Mixed Granularities

(b) and Berlin, platform 1 ($b1$*) respectively. Platform 1 (*$b1$*) and platform 2 (*$b2$*) in Berlin are also connected via an edge.*

In the previous example, any edge between between Berlin railway station (b) and the platforms contained in it ($b1$ and $b2$) would be inappropriate.

2.4.2 Routes

The *edges* of a transportation network are the primitive connections between locations. Edges are primitive in the sense, that no other location will be passed while moving along an edge.

A *route* in a transportation network is a path, i.e. a sequence of locations without repetitions and with subsequent locations being connected by edges, cf. Def. 2.2.

Definition 2.9 (Route) *Let (L, E) be a transportation network. A route $r = (l_1, \dots, l_n)$ in (L, E) is a path in (L, E).*

Route concatenation is denoted by a dot, i.e. $r = r_1 \cdot r_2$ denotes that r is a concatenation of r_1 and r_2.

The order of two locations l_1 and l_2 contained in a route r is denoted by $<_r$, i.e. $l_1 <_r l_2$ denotes the fact, that l_1 is prior to l_2 in route $r = (\dots, l_1, \dots, l_2, \dots)$.

We introduce *pre-routes* for later use. A *pre-route* is a prefix of another route.

Definition 2.10 (Pre-Route) *Let r be a route. r' is a pre-route of r if and only if a r'' exists so that $r = r' \cdot r''$.*

A pre-route relationship between r' and r is denoted by $r' \subset r$ (r' is a pre-route of r). The following notations are introduced for the case $r = r' \cdot l$:

- $head(r) = r'$ is the head of route r and
- $last(r) = l$ is the last location of route r.

For *notification planning*, we assume that all feasible routes from start to destination are in a *start-destination route set*.

Definition 2.11 (Start-Destination Route Set) *Let (L, E) be a transportation network. Let $l_{start} \in L$ be a start location and $l_{dest} \in L$ be a destination. A start-destination route set $R(l_{start}, l_{dest})$ in this transportation network is a set of routes where each route $r \in R(l_{start}, l_{dest})$ leads from the start location to the destination, i.e. r is a sequence of locations $r = (l_{start} = l_1, \dots, l_n = l_{dest})$ with $l_k \in L$ for $1 \leq k \leq n$ and $(l_k, l_{k+1}) \in E$ for $1 \leq k < n$.*

2.5 Information States

The *i-Alert* system must predict travel times for notification planning. A travel time depends on the time-dependent conditions of the transportation network (e.g. delays, congestions, weather conditions, etc.).

The current conditions on the transportation network are modelled by the so-called *information state*.

Definition 2.12 (Information State) *An information state s is an element of the finite n-dimensional information state space $S = S_1 \times \cdots \times S_n$ and each component S_i is a finite set of states for $1 \leq i \leq n$.*

The information state at time t is denoted by $s(t) \in S$. The single state sets S_i refer to global conditions (e.g. $S_i = S_{weather} = \{sunny, rainy\}$), single edges in the transportation network (e.g. $S_i = S_{ij} = \{free, congested\}$ for edge (i, j)) or even to single departures (e.g. $S_i = S_{ij}^{no.1221} = \{ontime, late\}$ for train no. 1221 on edge (i, j)).

The elements of $s \in S$ are called sub-states of s, e.g. $s_1 \in S_1$ is a sub-state of s.

Uncertain information states are modelled by random variables $\mathbf{s}(t)$ with probability space $(S, \mathcal{S}, P)$ where $\mathcal{S}$ is a σ-Algebra for S and P is the probability measure on $\mathcal{S}$.

Example 3 (Information State on Road Segments) *In the road scenario, the state of an edge is either* free *or* congested*, i.e. $S_{ij} = \{free, congested\}$ for any edge (i, j). If we only consider edges (c, e), (d, e) and (d, f) (in this order), then we have an information state space $S = S_{ce} \times S_{de} \times S_{df}$ (global states are omitted). $s(t) = (congested, congested, free)$ denotes the information state, where there are congestions on edges (c, e) and (d, e) but free flow on edge (d, f) at time point t.*

Example 4 (Information State for Train Departures) *In the train scenario, the state for the train departure is either* unknown *or* delay*, i.e. $S_{bc}^{no12} = \{unknown, delay\}$ for edge (b, c) where $no12$ is the identifier of the specific train departure. If we consider only this train departure, then the information state space is simply given by $S = S_{bc}^{no12}$. $s(t) = unknown$ is the information state, where no delay warning for this train has occurred until time point t.*

The modelling of the information state space, i.e. the selection of relevant global states and relevant states per edge is a crucial task. On the pragmatic side, it is important to have a small model in order to support fast computations. On the theoretical side, the information state should contain all relevant information for the prediction of travel times. Generally the complete event history, i.e. all events recorded up to now is the total information available at present. However, we do not want to have all possible event histories in our information state space. Therefore, the information state keeps a simple representation of the events happened so far.

The following example demonstrates a variety of possibilities for the modelling of the information state.

Example 5 (Modelling the Information State) *We want to process information about the traffic flow situation on a specific road. All congestions warnings and cancellation messages are recorded in the* event history*. The following possibilities arise (among many others). The information state keeps:*

- *The timestamps of both congestion warnings and cancellation messages of the last 24 hours.*
- *The total duration of congested situations during the last 24 hours.*
- *The last message only (either congestion warning or cancellation thereof).*

These examples are strong simplifications with respect to the event history keeping no information about old events and only little or no information about the occurrence time of recent events. Different event histories may lead to the same state.

For the travel scenarios considered here, sub-states per road segment (for the road scenario) and per train departure (for the train scenario) seem to be appropriate.

We introduce *information state transition probabilities* (shorthand: transition probability). Transition probabilities are denoted by $p_{ss'}(t,t') := P(\mathbf{s}(t') = s' \mid \mathbf{s}(t) = s)$ referring to the probability of the information state to be equal to s' at a time point $t' > t$ conditioned on the evidence for the information state equal to s at the time point t.

2.6 Events

Events are widely used for action triggers, as state change indicators and for the discrete description of processes. We are interested here in the following event types:

- changes of the information state, so-called *information state changes*,
- departures and arrivals of the traveller at certain locations, so-called *movement events* and,
- *time events*[2].

[2] A *time event* is the occurrence of a time point which has been previously set.

Other event types such as database state changes (due to deletes, inserts or updates) or signals received (in message systems) are not of interest here. Note, that information state changes can be interpreted as database changes and that any information state change is caused by a received message. However, we will look here at information state changes as our basic entities.

An *event* is an instantaneous occurrence of interest at a point in time (according to Chakravarty et al. [CKAK94]).

Numerous other event concepts exist. Sometimes, events are allowed to be non-instantaneous. Other definitions make a distinction between actions and events[3]. A brief comparison with other event concepts is given here:

- The UML [UML01] defines events as specifications of significant occurrences with location in time and space, i.e. non-instantaneous occurrences are not explicitly excluded. Note, UML distinguishes four different types of state change triggers, namely ChangeEvents (change of boolean expression), TimeEvents (expiration of a deadline), SignalEvents (reception of an asynchronous signal) and CallEvents (synchronous system call).
- In state-charts (cf. Harel [Har87]) and state diagrams (as part of the UML), events are instantaneous occurrences triggering state transitions, i.e. the event concept is limited to a specific purpose.
- The Reference Model of Open Distributed Processing (RM-ODP) [ISO95] does not provide an event definition, but the *action*-concept is closely related. However, actions usually have a duration.

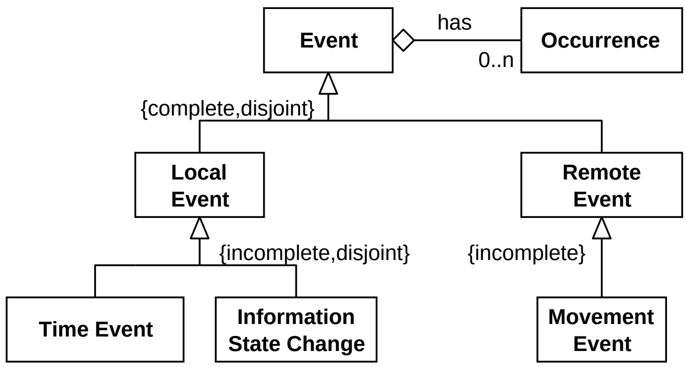


Figure 2.4: Information Model for Events

The information model for events is given in Fig. 2.4. An *event* has zero or more *occurrences*. We distinguish *local events* and *remote events*. Local events occur

[3]An event is observed (by the observer), while an action is made (by the actor), but this distinction is not always precise and will not be discussed here.

at the place of observation and are observed at the time of occurrence. Remote events do not occur at the place of observation and cannot be observed at the time of occurrence. *Information state change* and *time event* are specializations of *local event* while *movement event* is a specialization of *remote event*.

Instances of *occurrence* are called *events* in the sequel, while instances of *class event* are called *event classes*. *Events* are denoted $e_1, e_2, \ldots$. Event classes are denoted $E_1, E_2, \ldots$. If an event e_i is an instance of an event class E_j, then we annotate $e_i \in E_j$ as *element*-relationship. This relationship is non-exclusive, i.e. $e_i \in E_j$ and $e_i \in E_k$ is possible even with $E_j \neq E_k$. The occurrence time of an event e is denoted by $e.t$.

We give examples for different kinds of events:

Example 6 (Information State Change) *Consider an edge (i, j) in a transportation network with $S_{ij} = \{free, congested\}$. Let event class E_{cong} denote the occurrence of an information state change from* free *to* congested. *Then, event $e \in E_{cong}$ with $t_e = e.t$ denotes the change from $s_{ij}(t_e-) = free$ to $s_{ij}(t_e+) = congested$. t_e- and t_e+ denote time points short before and short after the events occurrence time t_e.*

We are not interested in the exact occurrence time of information state changes. Rather, we are interested in the probability of a certain state $s(t_2)$ at some time point t_2 conditioned on the evidence about a certain state $s(t_1)$ at some some time point $t_1 < t_2$. These transition probabilities have been introduced in Section 2.5 and their assessment will be discussed in Section 3.2.

Example 7 (Movement Event) *Consider that a traveller intends to depart from a location by train. $E^{traveller}_{depart}$ and E^{train}_{depart} are the respective event classes denoting the departure of the traveller and the train respectively. These event classes are uniquely determining that at most one occurrence is possible. Both events occur simultaneously, i.e. if there is an occurrence $e_{traveller} \in E^{traveller}_{depart}$ then there is also an occurrence $e_{train} \in E^{train}_{depart}$ with $e_{traveller}.t = e_{train}.t$ and vice versa.*

We do not specify a language for the explicit specification of movement events. The semantics of an event class will be given informally and only within an unambiguous context.

Example 8 (Time Event) *A time event is given by an explicit time point of occurrence. Let E_{15} denote the time event occurring at time point* 15. *Consequently, $e \in E_{15}$ with $e.t = 15$ is the only occurrence of this time event, E_{15} is unique.*

2.7 Situation and Decision Trees

We employ trees as a general notion for the modelling of alternative futures. A comprehensive study of reasoning based on event and decision trees can be found in Shafer [Sha97].

For the application domain of mobile route guidance, a situation of the traveller is characterized by

- the time point,
- the location,
- the information state,
- the determined route.

While time point, location and information state should be clear, the *determined route* needs some explanation. The *determined route* is the determined part of the travellers default route. The *determined route* has two parts describing past and future movements. Past movements end at the situations location, future movements start from the situations location. A *determined route* can only be extended, it is never changed. Other than the *determined route*, the travellers *default route* is not modelled by a situation. The *default route* is the route which will be followed by the user unless otherwise notified. The *default route* is an extension of the determined route up to the destination. Only the extension can be changed at some later time.

Definition 2.13 (Situation) *A situation is a 4-tuple* (l, r, s, t) *consisting of a location* l*, a determined route* r*, an information state* s *and a time point* t*.*

Example 9 (Second Junction in Road Scenario) *The traveller is at location* b *in the introductory road scenario of Fig. 1.6. Let* t_b *denote the actual time point* $s(t_b) = congested$ *is the information state at this time. The determined route is given by* (a, b, d)*, i.e. the next location* d *is already determined. The situation sit is given by*

- $sit.l = b$*, the location of the traveller,*
- $sit.r = (a, b, d)$*, the determined route,*
- $sit.s =$congested*, the information state and,*
- $sit.t = t_b$*, the current time.*

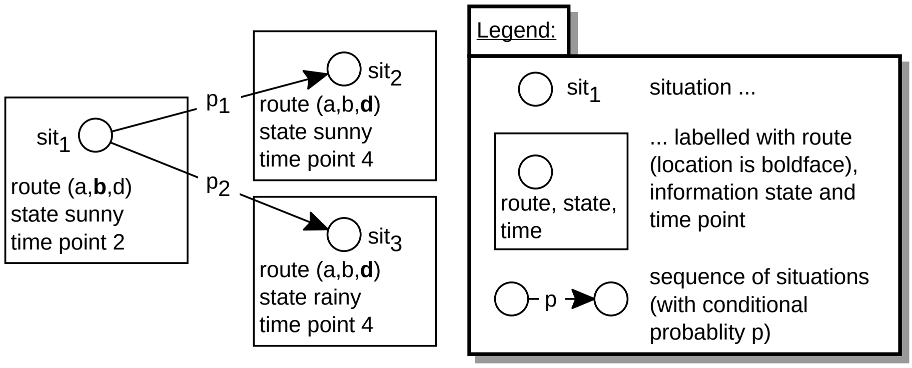


Figure 2.5: Example for an Information State Transition

The traveller's movement can be modelled as a *game against nature*, where the *player* (traveller or information system) takes moves for the improvement of the utility of the arrival time while nature chooses the information state at a later time point conditioned on the information state at some earlier time point. In a normal game, the *opponent* tries to minimize the utility of the player. That is not the case here. However, the arrival time and thus the utility of the traveller depends heavily on the development of the information states. Models for the travellers utility are given in Section 2.9.

Route decisions as extensions of the determined route are the moves of the traveller, while nature's moves are the random *information state transitions* (influencing the travel time) occurring between specified time points.

We will look at *information state transitions* first.

Example 10 (Information State Transition in Road Scenario) *The traveller is assumed to be at location b and uses the determined route (a, b, d). The actual time is $t = 2$. The current information state is given by $s(2) = sunny$. This is modelled by the leftmost situation sit_1 in Fig. 2.5.*

One out of two information states $s(4) \in S = \{sunny, rainy\}$ may arise at time point 4 when the traveller is located at location $l = d$. Situation sit_2 represents the situation of the traveller having travelled to location d at time point 4 with information state $s(4) = sunny$. Situation sit_3 represents the situation of the traveller having travelled to location d at time point 4 with information state changed to $s(4) = rainy$. Conditioned on situation sit_1, sit_2 occurs with probability p_1 and situation sit_3 occurs with probability p_2. In fact, the probabilities p_1 and p_2 are the transition probabilities $p_{sunny,sunny}(2, 4)$ and $p_{sunny,rainy}(2, 4)$ (cf. Section 2.5 for the introduction of transition probabilities and Section 3.2 for the assessment of transition probabilities.).

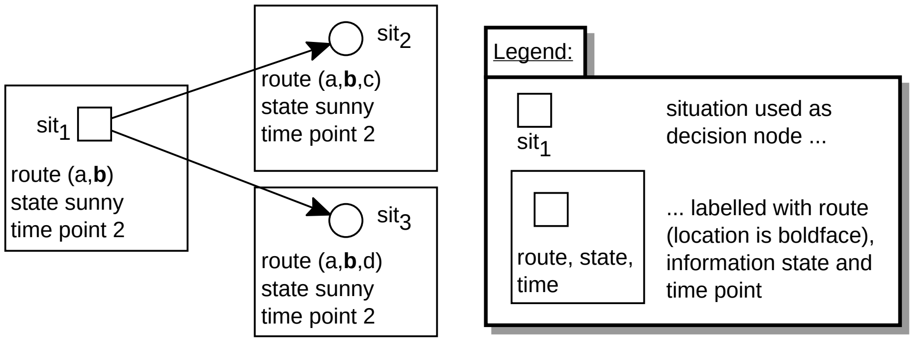


Figure 2.6: Example for a Route Decision

Now we look at *route decisions*. A route decision is the selection of subsequent locations, i.e. an extension of the determined route. Route decisions are assumed to be taken instantaneously, i.e. with unchanged time, information state and location.

Example 11 (Route Decision in Road Scenario) *The traveller is assumed to be at location b and used the determined route (a, b). The actual time point is 2. The current information state is given by $s(2) = sunny$. This is modelled by the leftmost situation sit_1 in Fig. 2.6.*

Now, the determined route is to be extended (a route decision has to be taken). Two successor nodes exist for possible route decisions (a, b, c) or (a, b, d). Situation sit_2 models the situation of the traveller having decided for route (a, b, c) and situation sit_3 models the situation of the traveller having decided for route (a, b, d). The situation sit_1 is used as a decision node which is indicated by depicting it as a rectangle.

A *decision tree* is a tree of situations where a situation together with its children either forms an information state transition or a route decision. The leave nodes of a decision tree have a certain value of reward.

Definition 2.14 (Decision Tree) *A structure $(CN, DN, LN, PC, cp, val)$ is a <u>decision tree</u> iff*

- $N = CN \cup DN \cup LN$ *is a finite set of situations, partitioned as*
  - CN*, the set of chance nodes*
  - DN*, the set of decision nodes*
  - LN*, the set of leave nodes*
- $PC \subseteq (CN \cup DN) \times N$ *is a set of parent-child relationships,*
- (N, PC) *is a tree,*

- *leave nodes don't have children, i.e.* $VN = \{sit \in N \mid Childs(sit) = \emptyset\}$,
- $cp : CN \times N \to [0,1]$ *assigns a conditional probability to any parent-child relationship starting from a chance node.*
- *chance nodes (together with their children) model information state transitions, i.e. for all* $sit \in CN$*:*
  - $\forall_{sit_1, sit_2 \in Childs(sit)} : sit_1.l = sit_2.l$ *(all children have the same location)*
  - $\forall_{sit' \in Childs(sit)} : sit.r = sit'.r$
    (all children have the same route as the parent)
  - $\forall_{sit_1, sit_2 \in Childs(sit)} : sit_1 \neq sit_2 \implies sit_1.s \neq sit_2.s$
    (all children represent different information states)
  - $\forall_{sit_1, sit_2 \in Childs(sit)} : sit_1.t = sit_2.t$ *(all children have the same time)*
  - $\forall_{sit' \in Childs(sit)} : sit.l <_{sit.r} sit'.l$
    (on the route, the children's location is after the parents location)
  - $\forall_{sit' \in Childs(sit)} : sit.t < sit'.t$
    (the children's time point is after the parents time point)
  - $\sum_{sit' \in Childs(sit)} cp(sit, sit') = 1$
    (the conditional probabilities of the children sum up to one)
- *decision nodes model route decisions, i.e. for all* $(sit_1, sit_2) \in DN \times N$*:*
  - $sit_1.l = sit_2.l$ *(the location remains unchanged)*
  - $sit_1.r \subseteq sit_2.r$ *(the route is extended)*
  - $sit_1.s = sit_2.s$ *(the information state remains unchanged)*
  - $sit_1.t = sit_2.t$ *(the time point remains unchanged)*
- $val : LN \to \mathbb{R}$ *assigns a value to any leave node.*

Example 12 (Partial Decision Tree for Road Scenario) *This example is a combination of example 10 and example 11. The traveller is assumed to be at location* b *and uses the default route* (a, b, c, e, g)*. The actual time point is* 2*. The current information state is given by* $s(2) = sunny$*. This is followed by a route decision and by a information state transition. The resulting decision tree is depicted in Fig. 2.7.*

The values of state nodes sit_4*,* sit_5 *and* sit_3 *are given by the decision tree.*

Decision trees can be evaluated, i.e. values can be assigned to all nodes (not only leave nodes). Let *eval* denote the evaluation of a node. For a given decision tree (CN, DN, LN, cp, val), any node $sit \in N = CN \cup DN \cup LN$ is evaluated according to the following rules:

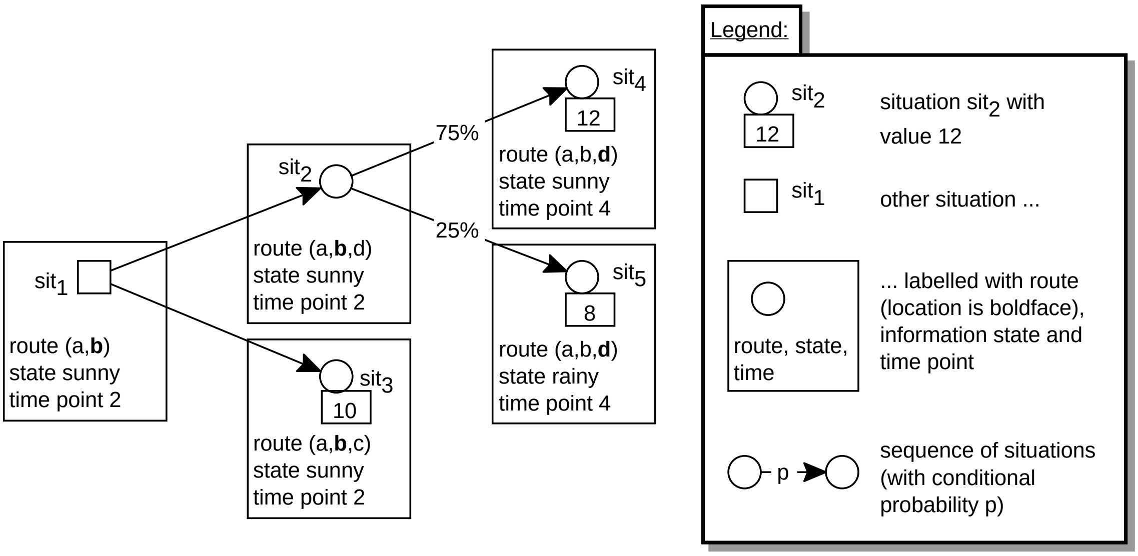


Figure 2.7: Example for a Decision Tree

- For a leave node $sit \in LN$, the evaluation is the value of that node: $eval(sit) = val(sit)$.
- For a decision node $sit \in DN$, the evaluation is the maximum of all children's values:

$$eval(sit) = \max_{sit' \in Childs(sit)} eval(sit')$$

- For a chance node $sit \in CN$, the evaluation is the conditional expectation of the children's values:

$$eval(sit) = E_{sit'|sit}\, eval(sit') = \sum_{sit' \in Childs(sit)} cp(sit, sit') \cdot eval(sit')$$

Example 13 (Decision Tree Evaluation) *The decision tree from Example 12 is evaluated. The value of state node sit_2 is calculated by the conditional expectation of the value of the successor, i.e. $eval(sit_3) = 0.75 \cdot eval(sit_4) + 0.25 \cdot eval(sit_5) = 0.75 \cdot 12 + 0.25 \cdot 8 = 11$. The value of state node sit_1 is calculated by the maximum of the values of its successors, i.e. $eval(sit_1) = \max_{sit' \in \{sit_2, sit_3\}} eval(sit') = 11$.*

Situation trees are used in this thesis for the modelling of an uncertain future without decisions and values. For ease of definition, we define situation trees by use of decision trees using a dummy function instead of a value function.

Definition 2.15 (Situation Tree) *A structure (CN, LN, PC, cp) is a* <u>*situation tree*</u> *iff $(CN, \emptyset, LN, PC, cp, LN \rightarrow \{0\})$ is a decision tree.*

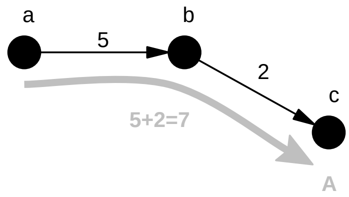


Figure 2.8: Constant Edge and Route Labels

A *situation tree* is a decision tree without decision nodes and without values for the leave nodes.

Bayesian networks and influence diagrams are employed in Chapter 4 for the compact representation and evaluation of the decision problems represented by such decision trees.

2.8 Travel Time

Next we model the travel time along edges and routes of the transportation network. By this, the time points of future situations can be computed. These time points are needed since

- the utility for the traveller depends on the *arrival time* (a time point) at the destination and
- the *time points of future notification options* (time points) are directly used for notification planning.

The travel time as duration along a route will be computed from the travel time along the single edges. This is illustrated in Fig. 2.8 for constant edge labels. The travel time along route A is the sum of the single edge labels, i.e. $5 + 2 = 7$ minutes.

Most path planning algorithms employ constant edge labels for the computation of shortest routes in a network. The following aspects of travel times in transportation networks cannot be modelled by constant edge labels:

1. *Waiting for Public Transport:* In the case of public transport, where transportation takes place with respect to a schedule, the waiting time (a duration) for the next vehicle (e.g. a train) has to be added to the transition time of the vehicle. The waiting time depends on the arrival time at a location.
2. *Peak-hours and Low-traffic:* Travel time between two locations depends on the departure time. Consider for instance peak-hour congestions or low-traffic periods between 2 am and 5 am.

3. *Network Conditions:* Specific events may influence the travel time in a network. Examples are accidents on a road, train delays, road or railway construction work and weather conditions, etc.
4. *Uncertainty:* The travel time duration might be uncertain.

The first and second aspect motivates the use of time-dependent edge labels, where the travel time along an edge depends functionally on the arrival time (earliest possible departure time) at the start location of that edge.

The third aspect motivates the use of state-dependent edge labels, where the travel time along an edge depends functionally on the information state at the time of departure (exactly: earliest possible departure time). Therefore travel time durations are probabilistic since developing information states form a random process.

The fourth aspect motivates the use of probabilistic edge labels, where the uncertain travel times are modelled by probabilistic durations.

We start with a definition for non-probabilistic edge labels. An edge label is a state- and time-dependent travel time function which models the travel time along an edge.

Definition 2.16 (Edge Label) *Let (i, j) be an edge. Let S be an information state space. The edge label $d_{ij} : S \times \mathbb{R} \rightarrow \mathbb{R}_+$ models the travel time (a duration) for travelling along edge (i, j). $d_{ij}(s, t)$ is the travel time along edge (i, j) when starting from location i at time point t and in information state s.*

By this, both time- and state-dependent edge labels can be modelled. Constant edge labels, time-dependent edge labels and state dependent edge labels are viewed as special cases of the more general time- and state-dependent edge labels:

Definition 2.17 (Types of Edge Labels) *Let d_{ij} be an edge label.*

- *d_{ij} is a constant edge label, iff $\forall_{s \in S} \forall_{t \in \mathrm{R}}\, d_{ij}(s, t) = d_{ij}$.*
- *d_{ij} is a time-dependent edge label, iff $\forall_{s \in S} \forall_{t \in \mathrm{R}}\, d_{ij}(s, t) = d_{ij}(\bullet, t)$.*
- *d_{ij} is a state-dependent edge label, iff $\forall_{s \in S} \forall_{t \in \mathrm{R}}\, d_{ij}(s, t) = d_{ij}(s, \bullet)$.*
- *d_{ij} is a time- and state-dependent edge label, iff it is neither constant nor time-dependent or state-dependent.*

For convenience, we denote instances of constant edge labels by d_{ij} and instances of time-dependent edge labels by $d_{ij}(t)$ where t is the time of departure from i, thus omitting those parameters that do not carry information for the travel time along that edge.

The computation of the travel time along routes will be discussed for some edge types in the following sections. Travel times along routes will be expressed by route labels. A route label is a state- and time-dependent travel time function which models the travel time along a route.

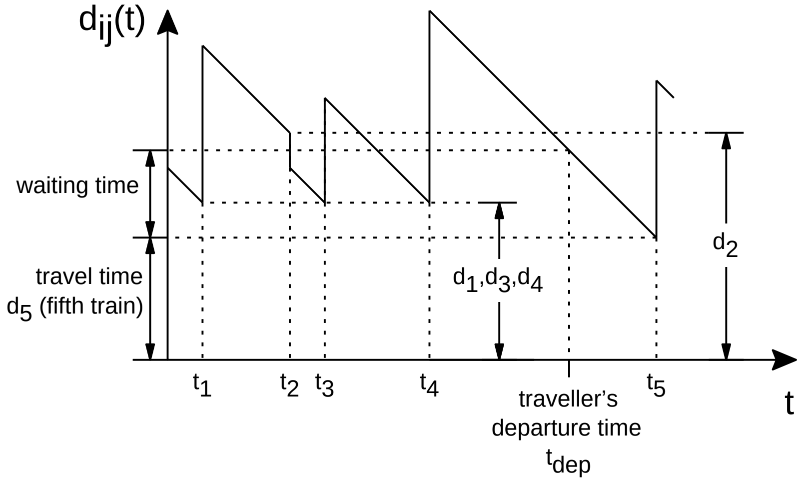


Figure 2.9: Time-Dependent Travel Time $d_{ij}(t)$ for Edge (i, j)

Definition 2.18 (Route Label) *Let r be a route and let S be a domain of the information state. The* <u>*route label*</u> $d_r : S \times \mathbb{R} \to \mathbb{R}$ *models the travel time along route r. $d_r(s, t)$ is the travel time along route r when starting in information state s at time t.*

A route label for route $r = (i_1, \ldots, i_n)$ is denoted by d_r or $d_{(i_1,\ldots,i_n)}$.

2.8.1 Time-Dependent Edge Labels

A time-dependent deterministic edge label models the travel time along an edge when starting at a specific time. This is useful for scheduled public transportation, where transition times from one location to another depend on the waiting time for the next vehicle and on the vehicle's transition time according to the schedule.

For scheduled traffic, time-dependent deterministic edge labels $d_{ij}(t)$ correspond to saw tooth functions. An example is given in Fig. 2.9 for trains. Trains numbered by $1, \ldots, 5$ depart at time points $t_1, \ldots, t_5$ and have travelling times $d_1, \ldots, d_5$ respectively. A traveller wants to depart from location i at time t_{dep}. He must wait until time point t_5 (waiting time is $t_5 - t_{dep}$). Then he can take the fifth train. The journey itself has a duration of d_5, i.e. the total transition time from i to j is $t_5 - t_{dep} + d_5$. Therefore, $d_{ij}(t_{dep}) = t_5 - t_{dep} + d_5$.

With time-dependent edge labels, arrival times at intermediate locations are computed iteratively. For route $r_n = (i_1, \ldots, i_n)$, let t_1 denote the departure time at location i_1 and t_k for $1 < k \leq n$ denote the arrival time at location i_k.

$$t_k = t_{k-1} + d_{i_{k-1}i_k}(t_{k-1}) \qquad \text{for } 1 < k \leq n \tag{2.1}$$

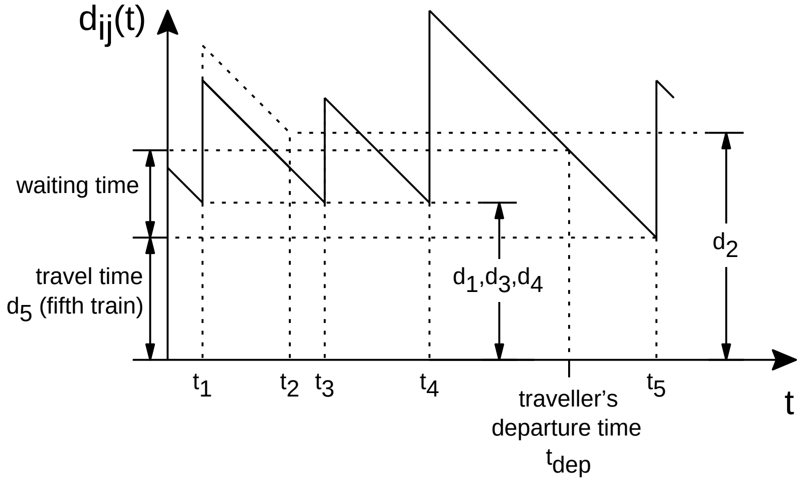


Figure 2.10: Start-Arrival Monotone Travel Time for Edge (i, j)

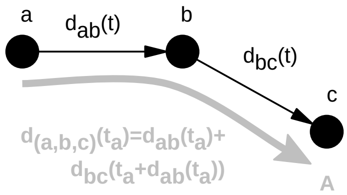


Figure 2.11: Time-Dependent Travel Times

Arrival times which are computed according to this iteration law do not necessarily lead to the earliest possible arrival time for travelling along a route. Consider for instance the edge label depicted in Fig. 2.9. There, the third train departs after but arrives prior to the second train, i.e. $d_{ij}(t_2 - \epsilon) = d_2$ while $d_{ij}(t_2 + \epsilon) = d_3$. A traveller arriving prior to the second trains departure should wait for the third train instead of taking the second train.

For the computation of fastest paths, we require travel times along edges to be start-arrival monotone, thus guaranteeing departing later always results in later arrival.

Definition 2.19 (Start-Arrival Monotonicity) *An edge label d_{ij} is called <u>start-arrival monotone</u>, if*

$$\forall_{t,t' \in T}\, t < t' \Rightarrow d_{ij}(t) < d_{ij}(t') \tag{2.2}$$

To obtain start-arrival monotone edge labels, slow trains have to be hidden away. This is depicted in Fig. 2.10 for the example. Here, train number 2 is hidden.

The computation of time-dependent route labels is illustrated in Fig. 2.11 for route (a, b, c). Here, the travel time duration for the whole route is given at once by the sum of the travel times along the single edges. $d_{ab}(t_a)$ is the travel time along edge

(a, b) when starting at time t_a and $d_{bc}(t_a + d_{ab}(t_a))$ is the travel time along edge (b, c) when starting at time $t_a + d_{ab}(t_a)$.

The travel time duration along the route $(i_k, \ldots, i_n)$ without explicit computation of intermediate arrival times can be computed recursively by

$$d_{(i_k,\ldots,i_n)}(t) = \begin{cases} d_{i_k i_{k+1}}(t) + d_{(i_{k+1},\ldots,i_n)}(t + d_{i_k i_{k+1}}(t)) & \text{for } k < n-1 \\ d_{i_k i_n}(t) & \text{for } k = n-1 \end{cases}$$

We employ time-dependent route labels for the computation of route sets (a preprocessing step for notification planning) (cf. Section 3.1). The start-arrival monotonicity allows for the efficient computation of fastest paths (cf. Kämpke and Schaal [KS98], Schaal and Kämpke [SK00] for a detailed discussion on this topic).

2.8.2 Time-Invariant State-Dependent Edge Labels

The travel time may depend on the information state at the departure time. We study the case of state-dependent edge labels without time-dependency in this section and the case with time-dependency in the next section.

The computation of a route travelling time with state-dependent edge labels requires knowledge about the information states at intermediate locations. Every sequence of information states at intermediate locations may lead to another travel time along the route. Vice versa, the travel times between locations influence the probabilities of information states at intermediate locations. The arrival time at a location after travelling along a specific route can be modelled by a probabilistic time point.

Let $r = (i_1, \ldots, i_n)$ be a route. Let $\mathbf{t}_k$ denote the uncertain time point and $\mathbf{s}_k$ denote the uncertain information state at location i_k.

For a specific sequence of information states $s_1, s_2, \ldots, s_n$ at the locations $i_1, i_2, \ldots, i_n$, we can calculate the time points $t_1 = t_{start}, t_2, \ldots, t_n$ for the departure times at locations $i_1, i_2, \ldots, i_{n-1}$ and the arrival time t_n by:

$$\begin{aligned} t_2 &= t_1 + d_{i_1 i_2}(s_1, t_1) \\ \vdots &= \vdots \\ t_{k+1} &= t_k + d_{i_k i_{k+1}}(s_k, t_k) \\ \vdots &= \vdots \\ t_n &= t_{n-1} + d_{i_{n-1} i_n}(s_{n-1}, t_{n-1}) \end{aligned} \tag{2.3}$$

With time points t_k known for $1 \leq k \leq n$ the probability of the sequence of information states conditioned on the initial state s_1 can be calculated by the product of the respective transition probabilities (assuming the markov property for the sequence of information states):

$$P(s_1, \ldots, s_n \mid s_1) = \prod_{k=1}^{n-1} p_{s_k s_{k+1}}(t_k, t_{k+1}) \tag{2.4}$$

The tree of possible information state sequences along a single route can be modelled by a situation tree (cf. Section 2.7). For a specific route $r = (i_1, \ldots, i_n)$, start time t_{start} and information state $s(t_{start})$ at start time, the situation tree (CN, LN, PC, cp) is declared as follows:

1. Situation $sit = (i_1, r, s(t_{start}), t_{start})$ is the root node, i.e. the route r is chosen as a determined route. If $n > 1$, then sit is inserted in the set of chance nodes CN, otherwise sit is the only node of the tree and inserted in the set of leave nodes LN.
2. If $sit = (i_k, r, s, t) \in N = CN \cup LN$ and $k < n$, then for any $s' \in S$:
   - $n' = (i_{k+1}, r, s', t')$ is a new node with $t' = t + d_{i_k i_{k+1}}(s, t)$. If $n > k + 1$, then sit is inserted in the set of chance nodes CN, otherwise sit is inserted in the set of leave nodes LN.
   - $(n, n') \in PC$
   - $cp((n, n')) = p_{ss'}(t, t')$.
3. Nothing else is in CN, LN and PC.

Example 14 (Stockholm-Munich I) *Consider a journey from Stockholm (s) to Munich (m) via Berlin (b) by car. We look at the route segment from Berlin to Munich first. The information state is assumed to be the weather condition, i.e. sunny (s) or rainy (r). On rainy days, the trip from Berlin to Munich takes 10 hours ($d_{bm}(r) = 10$) instead of 6 hours as it is the case on sunny days ($d_{bm}(s) = 6$). Assume it is sunny on the day before the trip. Let the probability of a change from sunny to rainy weather within a day be 50%. The trip from Stockholm to Berlin takes exactly 24 hours, independent of the weather conditions ($d_{sb}(\dot{}) = 24$).*

This scenario can be modelled as follows (and Fig. 2.12 shows the resulting event tree):

- *Two information states[4]: $S = \{s, r\}$ ($s \equiv$ sunny, $r \equiv$ rainy) with state transition probabilities*

$$p_{sr}(t, t') = 0.5 \tag{2.5}$$
$$p_{ss}(t, t') = 0.5 \tag{2.6}$$

  with $t + 24 = t'$

[4]Only one global state variable for actual weather is used here.

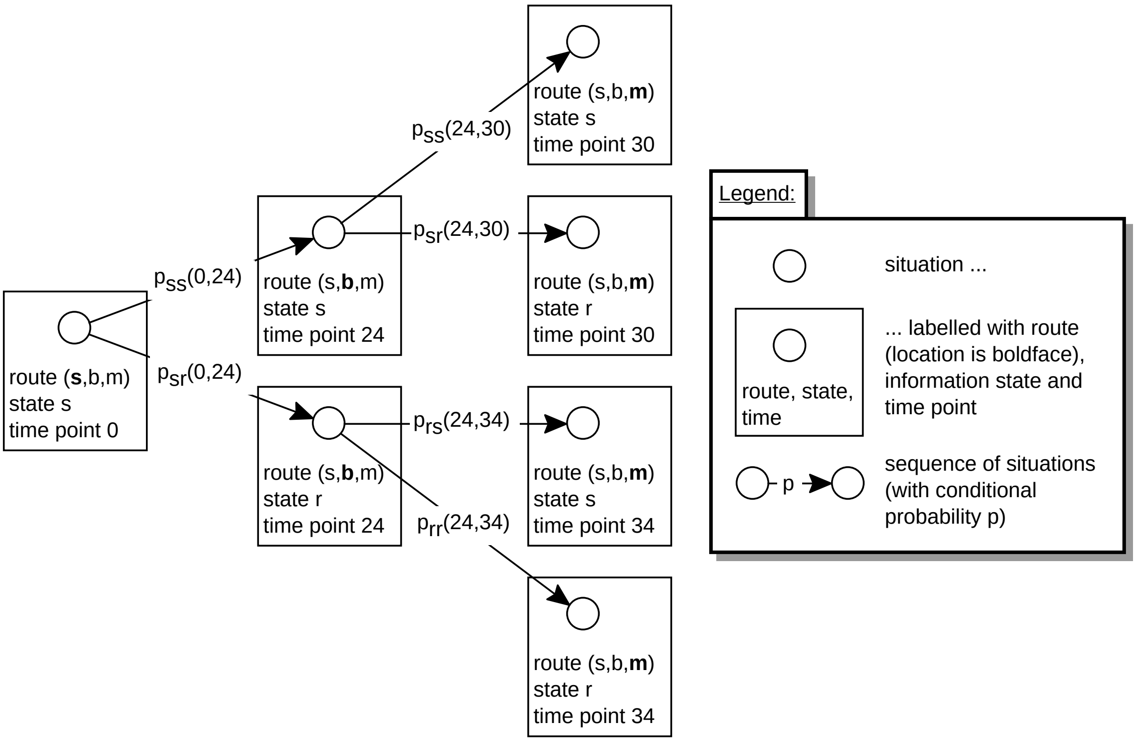


Figure 2.12: Event Tree for Example 14

- *Three locations:* $L = \{s, b, m\}$ *(*$b \equiv$ *Stockholm,* $b \equiv$ *Berlin,* $m \equiv$ *Munich).*
- *The state-dependent edge labels are given by*

$$d_{bm}(s) = 6\,(hours) \tag{2.7}$$
$$d_{bm}(r) = 10\,(hours) \tag{2.8}$$
$$d_{sb}(s) = 24\,(hours) \tag{2.9}$$
$$d_{sb}(r) = 24\,(hours) \tag{2.10}$$

Route $r = (i_1, \ldots, i_n)$, starting time t_1 at location l_1 and the initial information state $s(t_1) = s_1$ is given. Any specific sequence $iss = (s_1, \ldots, s_n)$ of information states at intermediate locations $i_1, \ldots, i_n$ has the conditional probability $P(iss \mid s_1)$ (cf. Eq. 2.4 and the arrival time $t_{dest}(iss)$ (cf. Eq. 2.3). Now, let $S(t) = \{iss \mid t_{dest}(iss) = t\}$ denote the set of all information state sequences, that have the same arrival time t. Then, the probabilistic arrival time $\mathbf{t}_{dest}$ after travelling along route r with starting time t_1 and initial state s_1 is given by $P(\mathbf{t}_{dest} = t) = \sum_{iss \in S(t)} P(iss \mid \mathbf{s}_1 = s_1)$.

Now, $\bar{d}_{(i_k,\ldots,i_n)}(s,t) = E(\mathbf{t}_{dest} - t_1 \mid \mathbf{s}_k = s, \mathbf{t}_k = t)$ (shortcut: $d_k(s,t)$) denotes the expected travel time for travelling along route $(i_k, \ldots, i_n)$ with starting time t and information state $s = s(t)$. The expected travel time $\bar{d}_1(s_1, t_1)$ is recursively given by:

$$\bar{d}_k(s,t) = \begin{cases} d_{i_k i_{k+1}}(s) + \sum_{s' \in S} \bar{d}_{k+1}(s',t') \cdot p_{ss'}(t,t') & \text{for } k < n-1 \\ d_{i_k i_n}(s) & \text{for } k = n-1 \end{cases} \tag{2.11}$$

with $t' = t + d_{i_k i_{k+1}}(s)$.

We continue our Stockholm-Munich example 14 and illustrate the computation of expected travel times with state-dependent edge labels:

Example 15 (Stockholm-Munich II) *The following expected travel time durations can be derived:*

- *The expected travel time duration for travelling from Berlin to Munich on a sunny day (i.e. with information state s at departure time 0):*

  $$\bar{d}_{bm}(s,0) = d_{bm}(s) = 6 \tag{2.12}$$

  .

- *The expected travel time duration for travelling from Stockholm to Munich on a sunny day (information state $s = s$, not $s = r$) and with departure time 0 from Stockholm:*

  $$\begin{aligned} \bar{d}_{(s,b,m)}(s,0) &= d_{sb}(s) + \sum_{s' \in S} p_{ss'}(0,24) \cdot \bar{d}_{(b,m)}(s',24) \\ &= 24 + p_{ss}(0,24) \cdot \bar{d}_{(b,m)}(s,24) + p_{sr}(0,24) \cdot \bar{d}_{(b,m)}(r,24) \\ &= 24 + 0.5 \cdot 6 + 0.5 \cdot 10 = 32 \end{aligned} \tag{2.13}$$

2.8.3 State- and Time-Dependent Edge Labels

There is only a small difference between state-dependent edge labels and time- and state-dependent edge labels. The respective formulas can be derived directly from Section 2.8.2. We state the formulas for expected arrival time here:

$$\bar{d}_k(s,t) = \begin{cases} d_{i_k i_{k+1}}(s,t) + E(\bar{d}_{k+1}(\mathbf{s}(t'),t') \mid s(t) = s) & \text{for } 1 \le k < n-1 \\ d_{i_{n-1} i_n}(s,t) & \text{for } k = n-1 \end{cases}$$

with t' (the departure time from location i_{k+1}) given by $t' = t + d_{i_k i_{k+1}}(s,t)$. The expected travel times are calculated by:

$$E(\bar{d}_{k+1}(\mathbf{s}(t'),t') \mid s(t) = s) = \sum_{s' \in S} P(\mathbf{s}(t') = s' \mid s(t) = s) \cdot \bar{d}_{k+1}(s',t') \tag{2.14}$$

with $(\mathbf{s}(t') = s' \mid s(t) = s) = p_{ss'}(t,t')$.

2.9 Utility of Arrival Time

In order to support a user on the route to his destination, his preferences have to be considered. We have to check evaluated routes and notifications against user's goals. This will be done by a utility model.

Which user's goals are to be considered? Generally, multiple conflicting goals exist in the domain of route guidance, e.g. low price vs. quick reaction. Modelling preferences for multiple goals is not our subject here. Comprehensive studies on modelling preferences with multiple objectives can be found e.g. in [KR76] and [Käm95]. We will focus on the *user's arrival time* only. A utility function $u : T \rightarrow \mathbb{R}_0^+$ is an order preserving mapping of the preferences of the user with respect to arrival time. Let t_1 and t_2 be arrival time options. Then $u(t_1) < u(t_2)$ if and only if the user prefers t_2 over t_1.

The utility of a random arrival time is assumed to be equal to the expectation of the utility for specific arrival times, i.e. $u(\mathbf{t}) = E(u(\mathbf{t})) = \sum_t u(t) \cdot p(\mathbf{t} = t)$ (discrete case). Optimization of expected utility reflects the user's preferences if the utility function fulfills the four axioms of preference as stated by von Neumann and Morgenstern [NM44].

Utilities for arrival times can be defined in many ways, some of which are listed here:

Earlier Arrival Time: An arrival time t_1 is better than arrival time t_2 if $t_1 < t_2$. The utility function can be any order-preserving transformation of the negative arrival time, e.g. $u(t) = -t$.

Deadline Constraint: In this case, a deadline $t_{deadline}$ is given. This leads to

$$u(t) = \begin{cases} 1 & t \leq t_{deadline} \\ 0 & \text{else} \end{cases} \tag{2.15}$$

The term *deadline confidence* is justified by this: The utility of a random variable for arrival time is linked to the probability of meeting the deadline.

The random arrival time is conditioned upon the current information state, current user location, time and default route as well as the choice of notification policy to be followed for current and future notification decisions. The best notification policy will be selected in such a manner as to optimize the utility of the arrival time.

2.10 Route Planning

This section describes (classical) route planning and route selection with developing information states. Notification planning will be discussed later.

2.10.1 Classical Route Planning

There is a huge literature on route planning covering different application domains (e.g. travelling, network routing, logistics, etc.) and different utility models as well as different types of transportation networks.

For mobile route guidance, we are only interested in so-called one-to-one problems, i.e. route planning from a single source (the current location of the traveller) to a single destination (the destination of the traveller).

For constant edge labels, the one-to-one problem is solved by Dijkstra's algorithm (cf. e.g. Lawler [Law76]). With time-dependent edge labels, the one-to-one problem (with starting time) can still be solved by a Dijkstra-like algorithm efficiently (cf. Kämpke and Schaal [KS98]).

With state-dependent edge labels, this is not efficiently possible, since for a given route $r = (i_1, \ldots, i_k, \ldots, i_n)$, starting time t_1 and starting state s_1 at location i_1 any combination of possible time points and information states at location i_k has to be considered even for simple utility functions (e.g. earlier arrival time).

A similar result has been found by Wellman et al. [WFL95] for random travel times in general (i.e. not explicitly caused by state-dependency). They argue, that the sum of expected travel times (modelled by constant edge labels) for a sequence of edges does not model the expected travel time along the whole route which is given by the sequence of edges. The reason is that the travel time duration along an edge is generally conditioned upon the departure time and the information about departure time distribution is lost by employing constant edge labels.

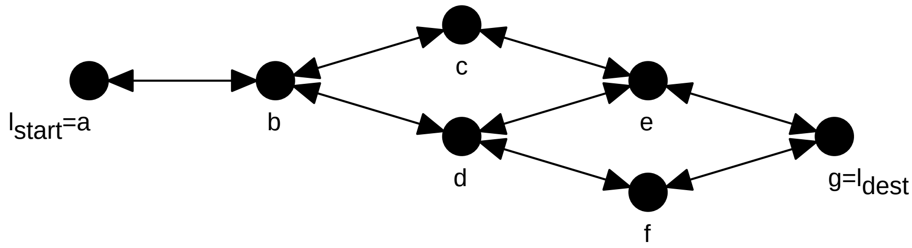
In this thesis, we will solve the following route planning problem.

Problem 1 (Fastest Paths Problem) *Given a transportation network (L, E) with time-dependent edge labels $d_{ij}(t)$ for $i, j \in E$, a start location $l_{start} \in L$, a destination $l_{dest} \in L$, a departure time t_{start}, find a start-destination route set $R(l_{start}, l_{dest})$ with k elements, so that for any pair of routes $r \in R(l_{start}, l_{dest})$, $r' \notin R(l_{start}, l_{dest})$:*

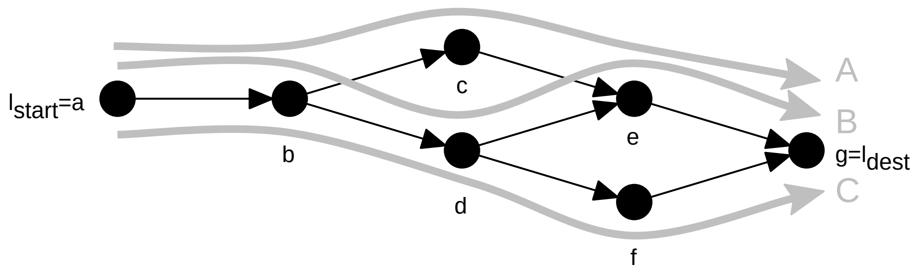
$$d_r(t_{start}) \leq d_{r'}(t_{start}) \tag{2.16}$$

Route labels $d_r(t_{start})$ and $d_{r'}(t_{start})$ are computed according to Eq. 2.1. Equation 2.16 ensures that the set of k routes in $R(l_{start}, l_{dest})$ is the set of k-fastest paths from the start location to the destination.

The result of fastest paths computation is depicted in Fig. 2.13. Figure 2.13(a) shows a transportation network with two locations marked as start location and destination, respectively. The result of fastest paths computation is illustrated in Fig. 2.13(b).



(a) Transportation Network, Start, Destination



(b) A Start-Destination Route Set

Figure 2.13: The Fastest Path Problem

2.10.2 Route Selection with Developing Information States

Up to now, we have looked at route planning, i.e. finding the best route within a network efficiently. For developing information states, we will evaluate and compare single routes with respect to a given utility function. By this, we provide a method for route selection, i.e. finding the best route within a set of routes. Note, that efficient algorithms for the case of an arbitrary utility function don't exist[5].

With probabilistic arrival times, we have to look at the concept of a best route more closely. For route planning, we ignored the traveller's utility function and simply searched for the fastest route. For route selection, we assign utilities to routes.

Definition 2.20 (Expected Utility of a Route) *Let start time t, information state s at start time, start-destination route set $R(l_{start}, l_{dest})$ and the traveller's utility function u be given. The expected utility $\bar{u}(r, s, t)$ of route $r \in R(l_{start}, l_{dest})$ when starting at time point t in information state s is given by*

$$\begin{aligned} \bar{u}(r, s, t) &= E_{\mathbf{d}_r}(u(t + \mathbf{d}_r(s, t)) \mid s, t) \\ &= \sum_d P(\mathbf{d}_r(s, t) = d) \cdot u(t + d) \end{aligned} \tag{2.17}$$

where $\mathbf{d}_r(s, t)$ is the random travel time for route r when starting at time point t in information state s.

[5]For the case of earliest arrival time, Wellman et al. [WFL95] provided an efficient algorithm for dependent travel times employing the concept of *stochastic dominance*.

The expected utility of a route is the expectation for the utility of the random arrival time at the destination when travelling along the route and starting at current time and with the current information state.

We are not interested in single route selection here, where routes are selected at the beginning of a journey. Instead, we look at sequential route selections. We distinguish:

Myopic Route Selection *Myopic Route Selection* is the selection of a route out of a set of n routes with maximum expected utility and without consideration of later route choices.

Predictive Route Selection *Predictive Route Selection* is the selection of a pre-route with maximum expected utility under the assumption that predictive route selection is repeated for the remaining route choices whenever a pre-route ends.

In the following we shall define these concepts.

Definition 2.21 (Myopic Route Selection) *Given a transportation network* (L, E) *with edge labels* $d_{ij}(s, t)$ *for* $i, j \in E$*, a start-destination route set* $R(l_{start}, l_{dest})$*, initial information state* s*, initial time* t *and a utility function* u*, the* <u>*myopic route selection*</u> $r^*_{myopic} \in R(l_{start}, l_{dest})$ *fulfills*

$$\bar{u}(r^*_{myopic}, t, s) = \max_{r \in R(l_{start}, l_{dest})} \bar{u}(r, s, t). \tag{2.18}$$

This definition specifies the myopic route as a selection at the start location for a given start-destination route set. However, the same holds true for subsequent locations of the start-destination route set when information state and time are updated.

Predictive route selection is based on the concept of route selection and revision. At pre-route r, only the next pre-route r' is selected without revision.

Definition 2.22 (Pre-Route Set) *Let* $R(l_{start}, l_{dest})$ *be a start-destination route set. The* pre-route set R_{pre} *of* $R(l_{start}, l_{dest})$ *is defined by:*

$$R_{pre} = \{r \mid \exists_{r' \in R(l_{start}, l_{dest})} r \subset r'\} \tag{2.19}$$

The definition of pre-routes is given by Def. 2.10.

Example 16 (Pre-route Set for Road Scenario) *The pre-route set for the road scenario is depicted in Fig. 2.14. Not all pre-routes are of interest. Consider for example pre-route* (a, b, c)*. This pre-route has only one successor, namely* (a, b, c, e)*. No decision is to be taken after having followed* (a, b, c)*.*

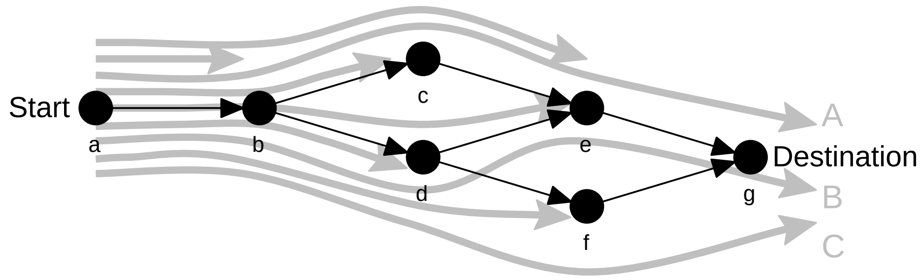


Figure 2.14: Pre-Route Set of the Road Scenario

With respect to a start-destination route set, a pre-route is said to lead to a decision point, if it has more than one successor.

Definition 2.23 (Decision Point) *Let $R(l_{start}, l_{dest})$ be a start-destination route set. Let L be the set of all locations contained in any route in $R(l_{start}, l_{dest})$. Let R_{pre} denote the pre-route set of $R(l_{start}, l_{dest})$. A route $r \in R_{pre}$ leads to a* <u>*decision point*</u> *if there exists at least two locations $l_1, l_2 \in L$, $l_1 \neq l_2$ so that $r \cdot l_1, r \cdot l_2 \in R_{pre}$.*

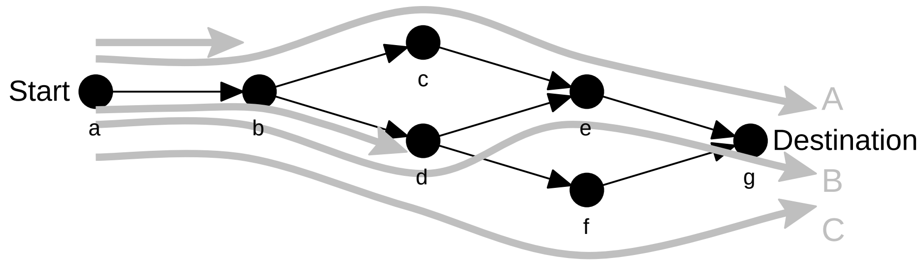
In Example 16, only pre-routes (a, b) and (a, b, d) lead to decision points. We construct the so-called *pre-route tree* from pre-routes leading to decision points and the routes in the start-destination route set.

Definition 2.24 (Pre-route Tree) *Let $R(l_{start}, l_{dest})$ be a start-destination route set. The tree $(R, <_{pre})$ is the* <u>*pre-route tree*</u> *of $R(l_{start}, l_{dest})$ with*

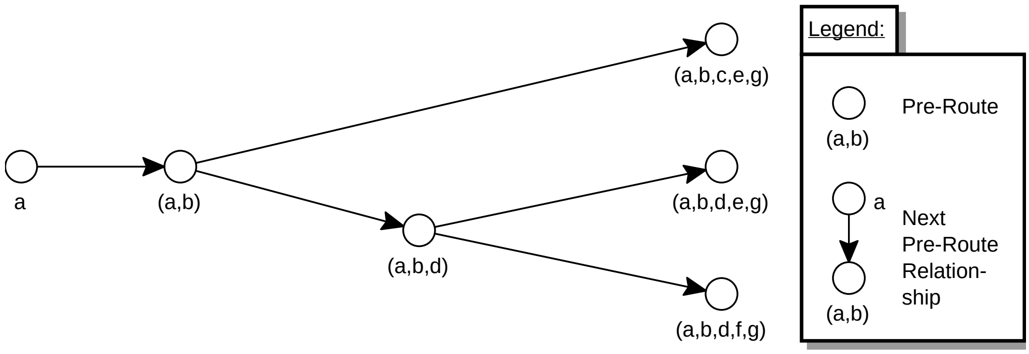
- *node set $R = \{(l_{start})\} \cup R_{dec} \cup R(l_{start}, l_{dest})$ where*
  - *l_{start} is the start location,*
  - *R_{dec} is the set of all pre-routes leading to a decision point,*
  - *$R(l_{start}, l_{dest})$ is the start destination route set.*
- *root node (l_{start}) (a pre-route consisting of the start location only),*
- *$<_{pre} = \{(r, r') \mid r, r' \in R \wedge r \subset r' \wedge \not\exists_{r'' \in R}\, r \subset r'' \subset r'\}$ is the parent-child pre-route relation and models subsequent decision nodes.*

Note, the leaves of the pre-route tree of $R(l_{start}, l_{dest})$ are the routes in $R(l_{start}, l_{dest})$.

Example 17 (Pre-route Tree for Road Scenario)
The construction of the pre-route tree for the road scenario is given in Fig. 2.15. The relevant pre-routes leading to decision points are derived from the previous example and shown in Fig. 2.15(a). The resulting pre-route tree is given in Fig. 2.15(b).



(a) Pre-Routes leading to Decision Points



(b) The Pre-Route Tree

Figure 2.15: Pre-Route Tree Construction

With predictive route selection, the utility for the remaining route when starting from a node in the pre-route tree is not bound to a specific route. Instead, alternative pre-routes can be chosen at any inner node of the pre-route tree and the utility of the remaining route can be improved at any decision point.

The optimal selection of pre-routes can be modelled by a decision tree (cf. Section 2.7). For a pre-route tree $(R, <_{pre})$ with root node r_0, start time t_{start}, information state $s(t_{start})$, and utility function u, the decision tree $DT = (CN, DN, LN, PC, cp, val)$ is constructed according to the following (declarative) definition:

1. Situation $sit = (l_{start}, r_0, s(t_{start}), t_{start})$ is the root node. sit is an element of the set of decision nodes DN unless $R = \{r_0\}$, in which case sit is the only node and contained in the leave node set LN with $val(sit) = u(t_{start})$.

2. If $sit = (l, r, s, t) \in DN$, then for any $r' \in Childs(r)$:

   - $sit' = (l, r', s, t) \in CN$,
   - $(sit, sit') \in PC$.

3. If $sit = (l, r, s, t) \in CN$, take the direct successor l' of l in r and for any $s' \in S$:

- $sit' = (l', r, s', t')$ is a node with $t' = t + d_{l,l'}(s, t)$. sit' is a decision node ($sit' \in DN$) if $(l_1, \ldots, l, l') \in R_{dec}$), sit' is a leave node ($sit' \in LN$) with $val(sit') = u(t')$ if $(l_1, \ldots, l, l') \in R_{l_{start}, l_{dest}}$), otherwise sit' is a chance node ($sit' \in CN$),
- $(sit, sit') \in PC$
- $cp((sit, sit')) = p_{ss'}(t, t')$.

4. Nothing else is in CN, DN, LN and PC.

Example 18 (Decision Tree for Road Scenario) *Fig. 4.14 shows a decision tree for the road scenario. This specific decision tree is constructed for a subtree of the pre-route tree shown in Fig. 2.15(b) rooted at pre-route (a, b). The values for the leave nodes are given and evaluated for all other nodes.*

We have used decision tree evaluation for predictive route selection here. For the evaluation of arbitrary strategies, we will formally define route selection policies and their expected utilities.

Definition 2.25 (Route Selection Policy) *Let $(R, <_{pre})$ be a pre-route tree, S an information state space and $T \subset \mathbb{R}$ a finite set of time points. A route selection policy is a function $\delta : R \times T \times S \rightarrow R$ with $\delta(r, t, s) \in Childs(r)$.*

With pre-route tree $(R, <_{pre})$ given, a *route selection policy* selects one pre-route $\delta(r, t, s) \in Childs(r)$ for any combination of a pre-route $r \in R$, an information state $s \in S$ and a time point $t \in T$.

Definition 2.26 (Expected Utility of a Route Selection Policy) *Let r denote the current pre-route with current location $l = last(r)$. Let s be the current information state and t be the current time. Then, the expected utility $\bar{u}(\delta, r, s, t)$ for route selection policy δ when starting with pre-route r, in information state s and at time t is recursively given by*

$$\bar{u}(\delta, r, s, t) = \begin{cases} u(t) & \text{for } r \in R(l_{start}, l_{dest}) \\ \sum_{s' \in S} p_{ss'}(t, t') \cdot \bar{u}(\delta, r \cdot l', s', t') & \text{else} \end{cases} \tag{2.20}$$

with $l' = last(\delta(r, t, s))$ and $t' = t + d_{ll'}(s, t)$

This will be used for the definition of predictive route selection.

Definition 2.27 (Predictive Route Selection) *Given a pre-route tree* $(R, <_{pre})$*, initial pre-route* r*, initial information state* s*, initial time* t*, and utility function* u*, the best predictive route selection at pre-route* r *is a pre-route* $r^*_{predict} = \delta^*(r, t, s) \in Childs(r)$*, where* δ^* *is a best policy for selecting the next pre-route, i.e.* δ^* *fulfills*

$$\bar{u}(\delta^*, r, s, t) = \max_{\delta \in \Delta} \bar{u}(\delta, r, s, t). \tag{2.21}$$

This definition specifies the predictive route selection as a selection of a pre-route r' according to the optimal policy δ^*.

The expected utility with predictive route selection is greater or equal to the expected utility with myopic route selection. It is equal for the case of independent states at different time points when nothing can be learned from the current information state about future information states.

We cannot control the route selection of the traveller directly. Instead, the traveller is to be notified about the best route to follow. We introduce the concept of planning with notification options (so-called notification planning) in the next section. These are the major differences:

- Instead of deciding about the actual route segment to be used, we decide about a route segment for notification. However, the traveller will be informed about the most likely route even in the case of predictive route selection.
- Instead of considering the time of an arrival at future locations, we consider the time of notification options. In the simplest case, the earliest possible arrival time at a decision point is used as the time for a notification option.

A next section is dedicated to the concept of notification planning.

2.11 Notification Planning with Notification Options

Other than route selection, *notification planning* is concerned with the right time and content of notifications to be sent to the traveller. Several assumptions will be taken in order to keep notification planning comparable with route selection policies. Most of these assumptions can be dropped for true applications, due to the flexibility of the proposed influence diagram models for the implementation of notification planning.

In order to define the concept of *notification planning*, we will have a look at the following questions:

- What is a notification?
- What is the effect of a notification? Are combined effects of notifications to be considered?
- What is the value of a notification?

A *notification* is a message sent from the information system to the user. We assume first, that the message transmission requires no time. This is a possible and convenient simplification. It is *possible*, since the modern traveller is notified quickly via mobile devices and the real time required for message transmission is relatively short. It is *convenient*, since a notification can thus be modelled as an atomic event.

Assumption 1 (Instantaneous Messaging) *The transmission of a notification requires no time, i.e. the traveller can use the content of a notification at exactly the same time point at which the notification has been sent.*

A notification is characterized by time, location, content and presentation, i.e. its information logistics' dimensions, and *push* is implicitly given as mode of transmission (cf. Section 1.2). The presentation of the notification is not considered here. It is assumed, that an appropriate presentation exists when a notification is sent.

Assumption 2 (Appropriate Representation) *A notification can always be sent in an appropriate representation for the traveller, so that the traveller will be able to perceive the notification content.*

The *notification effect* is the impact on user's action. The modelling of the notification effect is a very difficult task since many assumptions from cognitive sciences, artificial intelligence, psychology and perceptions from everyday's life are combined either explicitly or implicitly. We cannot dig into this deeply. Instead, we will use a very simple assumption about the notification effect.

Assumption 3 (Traveller's Obedience) *The traveller follows his* default route *until changed by a notification. Every time the user has to choose between alternative routes, he will follow his default route.*

This assumption is justified since (1) the traveller does usually not have superior knowledge about route alternatives and (2) it is convenient to follow the suggestion offered by the notification.

In addition to assumptions 1, 2 and 3, the following assumption is necessary for the information system to apply route selection policies for the computation of appropriate notifications.

Assumption 4 (Identifiable Location) *The location of the traveller is always known.*

Now, what is the value of a notification? Informally, the *notification value* is the difference between the expected utility achieved *with* notification and the expected utility achieved *without* notification. The notification value depends upon the time of evaluation, since the arrival time is dependent on the developing information state.

It is not always optimal to issue a notification with positive notification value. Consider the following cases:

- *Multiple notifications* with the same notification effect are redundant. Even though all of them may have positive notification value, it is useless to send more than one.
- *Interdependencies between notifications* need to be resolved. An earlier notification may cancel the notification effect of a later notification, e.g. the default route which is implied by the later notification becomes inaccessible due to following the default route which is implied by the earlier notification.

Notifications which are not issued yet but may transmitted in the future will be called *notification options*. A notification option becomes a notification when issued by the information system. The selection of notification options is our main planning problem. For the analysis of multiple notification options future decisions are modelled by a *notification policy*. A notification policy determines which notifications will be issued now and in the future depending on the information state at the respective notification time point.

With the assumptions set up to here, the problem of finding a best notification at current time is very similar to *route selection*, except (1) notifications need only be sent if different from current default route and (2) the set of decision points is constrained by the set of notification options.

The rest of this section is structured as follows. Notification options and some heuristics for reducing the number of notification options will be discussed in Section 2.11.1. Notification planning is discussed in Section 2.11.2 together with a formal specification of the notification problem. Finally, we give an interpretation for the concept of *notification value* in Section 2.11.3 which is based on the concept of notification planning.

2.11.1 Notification Options

A notification option *no* is a future possibility for a notification of the traveller, and it is characterized by time and content where

- the *time* is the time at which the notification occurs, i.e. the time at which the message is sent from the information system and received by the traveller, and
- the *content* represents a route from the start location of the traveller to the destination of the traveller.

Definition 2.28 (Notification Option) *Let $R(l_{start}, l_{dest})$ be a start-destination route set. A notification option $no = (r, t)$ is a 2-tuple with*

- *$no.r \in R(l_{start}, l_{dest})$ is the intended content for notification,*
- *$no.t$ is the intended time of notification.*

Not all combinations of time and content need to be considered. Instead, we reduce the total number of notification options as follows:

1. As the traveller needs to choose between routes at decision points only, a notification has to be considered at decision points only. The time points at decision points can be derived from the decision tree for predictive route selection.
2. For notifications which are issued at a specific decision point, only routes starting with the respective pre-route are to be considered as notification content.
3. For each decision point, only limited time points prior to the decision points time will be considered, e.g. the earliest possible and the average arrival time at the respective decision point.
4. If several candidates for notification time are specified for a single decision point, these are to be considered exclusively.

Example 19 (Placement of Notification Options) *The placement of notification options prior to decision points is visualized in Fig. 2.16 with two examples. The examples have been derived from the scenarios in Section 4.*

Here, notification options are marked as white circles. In the example to the left, notification options have been specified for pre-route (a, b) and (a, b, d), respectively. They coincide with the respective destination of the pre-routes. In the example to the right, two notification options with different time points have been placed prior to train departure along pre-route (a, b).

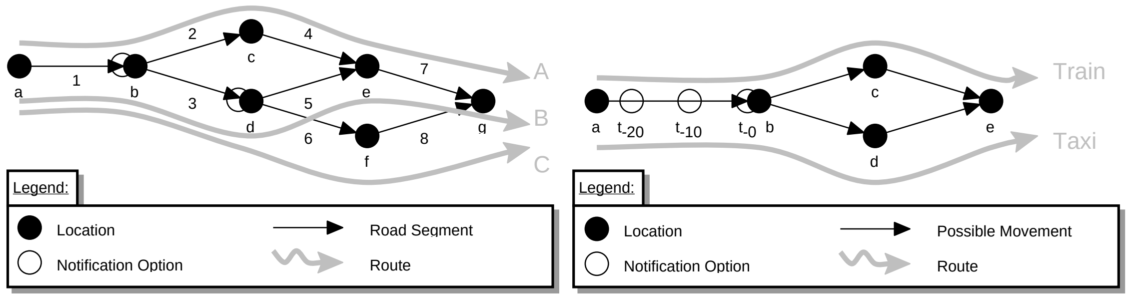


Figure 2.16: Placement of Notification Options

User's default route, notifications sent and pre-routes interact in the following manner:

- Unless notified, the user always follows his default route, i.e. only pre-routes of the default route can be followed-up.
- If a notification is sent to the user in time, the notification content replaces the default route.

Example 20 (Default Route, Notifications, Pre-Routes) *In Example 19, the default route can be selected from* A,B *prior to decision point (pre-route)* (a,b)*. With default route* A*, the next pre-route is* (a,b,c) *(no decision point) and with default route* B *or* C*, the next pre-route is* (a,b,d) *(a decision point).*

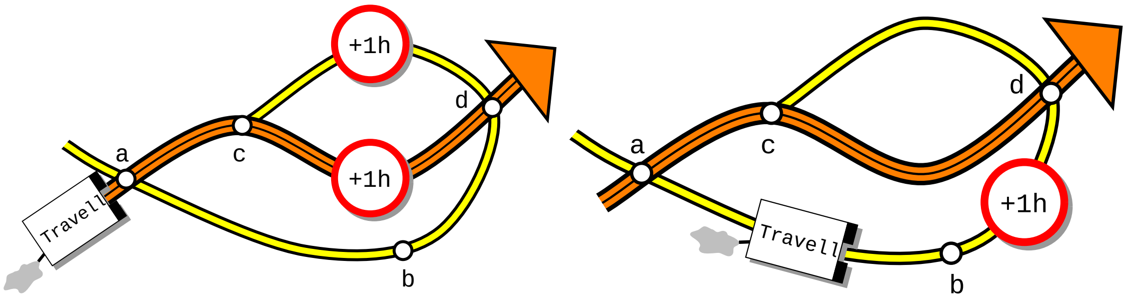
2.11.2 Notification Planning

Notification planning is a method proposed in this thesis for the selection of the best time and content for traveller notification. Notification planning uses future notification options, i.e. the improvement of the expected utility to be achieved by future notifications is considered for the selection of current notifications.

Notification planning faces numerous uncertainties:

- Uncertainty about the information state in the future.
- Uncertainty about the notification effect of current and future notifications.
- Uncertainty about the travel time.

For notification planning, predictive route selection is adapted so that only the notification options are considered for future decisions. With correct notification options notification planning based on predictive route selection is better than notifications based on myopic route selection.



(a) Before Location a (b) One possible situation after Location a

Figure 2.17: Road Scenario

For the demonstration of the core functionality of an *i-Alert* service, pre-trip planning (based on myopic route selection) and predictive route selection (with full control) will be compared by example. Other than pre-trip planning, an *i-Alert* service considers future notification options. Other than predictive route selection, notification planning is bound to these notification options.

Example 21 (Comparison of Route Selection Policies) *A specific situation is depicted in Fig. 2.17(a) and 2.17(b) respectively. Before reaching location a (the first junction from the introductory scenario), two congestions on different routes are known (encoded in the information state), each of them vanishing independently with a probability of 50 %. Another congestion may develop on the third route* (a, b, d) *with probability 30 %. The best route at that time and that information state without consideration of future decisions is* (a, b, d)*. This route is to be selected for pre-trip planning at the time of departure.*

In the case of predictive route selection, the situation is slightly more complicated. Then, the probability of future situations has to be known. One congestion disappears with probability 50 % (less than the probability of 70 % for free flow (a, b, d)*), but one of both disappears with probability 75 %. The situation depicted in Fig. 2.17(b) illustrates such a situation after choosing the route selected by the pre-trip route planning. With predictive route selection, both variants of* (a, c, d) *would have been superior than* (a, b, d)*.*

Loosely speaking, notification planning is superior to pre-trip planning, but inferior to predictive route selection. A formal comparison between different notification policies is given in appendix C. Notification planning can be viewed as predictive route selection where control is performed via notifications to the traveller.

2.11.3 Notification Value

This section is about the concept of notification value and about the avoidance of unnecessary notifications by notification planning.

Notification planning does not depend on the concept of notification value. Instead, notification planning simply optimizes the traveller's utility. But we can give an interpretation of notification planning by looking at the concept of the notification value.

Generally, routes are sent to the user in order to help him in achieving his goals. Thus, we want to measure the effect of notifications on the user's expected utility, cf. Section 2.9).

Now, for a single current notification[6] the notification value can be defined as the difference between the expected utility with and without notification. Consequently, the notification value is

- equal to zero, if the notification content is the same as the current default route,
- less than zero, if the current default route has a greater expected utility than the notification content.
- greater than zero, if the current default route has less expected utility than the notification content.

The implicit assumptions for the cases with and without notification are, that the traveller

- will follow the old default route until he reaches his destination if he is not notified and
- will follow a new default route if he is notified accordingly.

This is not the general case. The default route might be changed over and over again by subsequent notifications. Therefore, the opportunity costs of not notifying at the current time might be less or even negative when future notification options are considered. Notification planning does not only improve the traveller's utility, it may even help to avoid notifications completely be deferring current notifications.

A concept of notification value which is consistent with notification planning must consider future decisions about notifications. Therefore, we will compare the utility of so-called notification policies[7] for the assignment of a notification value to some current notification option:

[6] A *current notification* is a notification which takes place at current time.

[7] The idea for notification policies came up in a private discussion with K.-H. Waldmann (2001), TU Karlsruhe.

Definition 2.29 (Notification Policy) *A notification policy $po : R_{pre} \times T \times S \to R \cup \{\perp\}$ is a mapping from a set of embedded time points $T \subset \mathbb{R}$, information state set S and pre-route set R_{pre} into route set R. For each $t \in T$ the notification policy po assigns a notification content $po(r_{pre}, t, s)$ ($\perp$ is the empty content) to each decision point $r_{pre} \in R_{pre}$ and each state $s \in S$.*

Note, a notification policy is very similar to a route selection policy except (1) only certain time points are considered for notification and (2) the empty notification content $\perp$ instead of a route selection. The evaluation of the expected utility of a notification policy builds upon the developing default route, but is not given formally here.

A notification policy tells us, which notifications will be sent in which situation. Thus, a notification policy does not only determine current notifications but also determines future notifications. Application of a notification policy implies that the traveller will be notified accordingly, i.e. at any point in time $t \in T$ having followed pre-route r_{pre} and with actual state s he will receive a notification with content $po(r_{pre}, t, s)$.

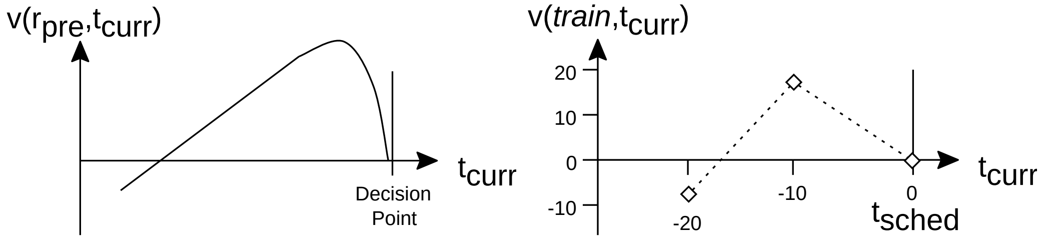
The expected utility of a policy $\bar{u}(po)$ is conditioned on current state, time and pre-route. It is the expectation of the traveller's utility when starting at the current state, time and location and being notified according to notification policy po.

Now we define the notification value (of a notification option). Let no be a notification option with $no.t = t_{curr}$ (a notification option at current time). Then $PO^{true}(no)$ is the set of policies where the traveller is notified at current time about the notification content $no.r$, i.e. $PO^{true}(no) = \{po \mid po(r_{pre}, t_{curr}, s_{curr}) = no.r\}$. Accordingly, $PO^{false}(no) = \{po \mid po(r_{pre}, t_{curr}, s_{curr}) \neq no.r\}$ is the set of policies where the traveller is not notified at the current time about this route. Then, the *notification value* at current time t_{curr} with current state s_{curr} of a notification option no is the difference between the maximum expected utility for a notification policy $po^* \in PO^{true}(no)$ and the maximum expected utility for a notification policy $po^* \in PO^{false}(no)$.

Definition 2.30 (Notification Value) *Let current time t_{curr}, current state s_{curr} and current pre-route r_{pre} given. Let no be a notification option with $no.t = t_{curr}$ (a notification option at current time). The notification value $v(no)$ is given by*

$$v(no) = \max_{po \in PO^{true}(no)} \bar{u}(po) - \max_{po \in PO^{false}(no)} \bar{u}(po) \tag{2.22}$$

A qualitative scenario for the change of notification value for the same notification content (route $no.r$) but with evolving current time t_{curr} is shown in Fig. 2.18(a). The traveller approaches a decision point, the notification content $no.r$ is different from the default route $r_{default}$ and it is assumed to be better than the default route for the whole segment of time shown here.



(a) A qualitative Scenario (b) A quantitative Example

Figure 2.18: Time-Dependent Notification Value

Due to a growing expected utility of route r, the notification value increases while the traveller approaches the junction until it reaches a maximum. But shortly before the traveller reaches the junction, the value will drop and decreases to zero. This behavior is due to the fact that no time will be left for a notification and a reaction to be in time and effective.

Example 22 (Notification Value) *A complete model for the train scenario is given in Section 4.4. With a specific value for the current information state, namely* none *(no delay warning), the notification value for the notification content* train *can be derived for time points* -20, -10 *and* 0. *These are shown in Fig. 2.18(b). At current time* $t = -20$, *the expected utility for immediate notification with notification content* train *is* 72, *while the best notification policy without such a notification has expected utility* 79.9, *cf. Section 4.4.2.1. Therefore, the notification value at this time is* $v(train, -20) = \max_{po \in PO^{true}(no)} \bar{u}(po) - \max_{po \in PO^{false}(no)} \bar{u}(po) = 72 - 79.9 = -7.9$, *cf. Table 4.5. At time* $t = -10$, *the expected utility for immediate notification about* train *is* 88, *while waiting provides an expected value of* 70. *Therefore, the notification value is* $88 - 70 = 18$. *At* $t = 0$, *none of the possible notifications does have an effect on the traveller, therefore, the expected utility of all notification policies is the same. The notification value equals zero.*

The notification value is closely related to information value. Ahituv [Ahi87] proposes a metamodel of information flow and introduces a concept of information value measured with respect to an individual human decision maker. Ahituv observes the fact that the information value is time-dependent. Data sets (similar to notification contents) are transformed into concepts perceived by the decision maker and this transformation can be time-dependent. However, any explicit notion of time-dependence is missing in Ahituvs metamodel.

Research on measuring information value has begun in 1966 when Howard [How66] introduced his concept of economic information value. He defines information value as the gain of expected reward by obtaining the outcome of an information in question. Ever since the concept of information value has received a broad interest both in the fields of economics (cf. [Law99]) and artificial intelligence (cf. [Pea91]).

2.12 Summary

We have introduced *time points*, *durations* and *travel times* in *transportation networks* as a basis for the computation of *deterministic* or *probabilistic state- and time-dependent route labels*.

We distinguish *route planning* and *route selection with developing information states*, the latter being the basis for *myopic route selection* and *predictive route selection*. Note, *route planning* and *notification planning* are completely different. *Route planning* plans for the network topology without consideration of the future. *Notification planning* plans for alternative futures under uncertainty.

These concepts will be used in later chapters:

- Chapter 3 actually treats two topics as prerequisites for notification planning. Fastest path planning and Transition Probability Assessment. The first topic is concerned with the preprocessing step of both myopic and predictive route selection, the computation of routes. The second topic is an elaboration on the estimation of transition probabilities.
- Chapter 4 discusses the compact representation and solution of instances of notification planning with influence diagrams.
- Chapter 5 discusses the implementation of *i-Alert* as an ILOG service.

Chapter 3

Routes and Transition Probabilities

This chapter is concerned with the computation of routes and transition probabilities needed for notification planning.

Given a transportation network, a start location, a destination and time-dependent edge labels, Section 3.1 discusses the computation of a route set $R(l_{start}, l_{dest})$ of fastest paths in case of a data federation.

Based on recorded data, Section 3.2 discusses the computation of the transition probabilities $p_{ss'}(t, t')$ for modelling the information state change.

3.1 Distributed Fastest Paths

The distributed computation of fastest paths has been studied and published previously in Schaal and Kämpke [SK00]. This approach will be used here for the computation of route set $R(l_{start}, l_{dest})$ for given start location l_{start} and destination l_{dest}. The major innovations of distributed computation of fastest paths vs. Dijkstra's algorithm for shortest paths are

- cross-organizational route planning vs. local route planning which involves the construction and exploitation of aggregated information (or meta-data) for shortest (fastest) path computation and
- the shift from *shortest* to *fastest* path with consideration of time tables which requires the concept of start-arrival monotonicity for time-dependent edge labels.

We present a method that computes fast routes within a distributed transportation system. Travel times adhere to fixed schedules. Consequently, they are independent from current network conditions and current request for transportation capacity. This distributed computational problem originated from the German project DELFI ("Durchgängige ELektronische FahrplanInformation"/ continuous electronic time table information, cf. Radermacher et al. [Rad98]).

The objective is to find multiple routes from a specified origin l_{start} to a specified destination l_{dest} utilizing different information sources so that the transition $l_{start} \to l_{dest}$ minimizes the travel time. Information sources represent transportation networks of autonomously acting transportation providers. They are interconnected by sharing certain locations.

The following issues will be covered in this section:

- The design of an appropriate model for fixed schedules.
- The development of an algorithm for the computation of fastest paths.
- The development of a method for aggregation of partial results for the distributed computation of fastest paths.

Optimal solutions can be found in a two step approach: An aggregated structure shrinks the search space in such a way that subsequent computations on the complete structure provide the fastest overall solution without searching the complete structure.

3.1.1 The Fastest Path Problem

A directed graph $G = (V, E)$ is assumed to be given over a finite, non empty set V of vertices and a non-empty set $E \subseteq V \times V$ of directed edges. G is assumed to be connected meaning that for each $v, w \in V$ there is a path of vertices $p(v, w) = (v = i_0, i_1, \ldots, i_k = w)$ so that $(i_0, i_1), \ldots, (i_{k-1}, i_k) \in E$. The label of edge (i, j) is a cost function $d_{ij} : T \to I\!R_{\geq}$ where the index set T denotes time points.

Vertices adhere to locations and paths adhere to routes. A travel along edge (i, j) starting at time t_0 takes time $d_{ij}(t_0)$. Let $p = (i_1, \ldots, i_r)$ with (i_1, i_2), …, (i_{r-1}, i_r) $\in E$ be a path without cycles, i.e. $i_s \neq i_t$ for $s \neq t$. This leads to arrival time t_k at vertex i_k with iteration law

$$t_k = t_{k-1} + d_{i_{k-1} i_k}(t_{k-1}) \text{ for } k = 2, \ldots, r.$$

We assume the function $d_{ij}(t)$ being start-arrival-monotone (cf. Definition 2.19), i.e. $t_a \leq t_b \Rightarrow t_a + d_{ij}(t_a) \leq t_b + d_{ij}(t_b)$. Any path from v to w starting at time t is denoted by $p(v, w; t)$. For $p(v, w; t) = (v = i_0, i_1, \ldots, i_k = w)$ the value of this path $\vartheta(p(v, w; t)) = t_k$ is given by the arrival time at destination w.

Definition 3.1 (Fastest Path) *A fastest path* $p^0(v,w;t_1)$ *from* v *to* w *starting at time* t_1 *is the solution of*

$$\min_{p(v,w;t_1)} \vartheta(p(v,w;t_1)).$$

An obvious lower bound for a fastest path is given by shortest paths, where each edge (i,j) receives a positive real-valued label $dmin_{ij}$ denoting the minimum of all proper transition times along the edge. The length of a path $p(v,w) = (v = i_1,\ldots,i_r = w)$ with respect to the time independent labels is denoted by $\lambda(p(v,w)) = \sum_{j=1}^{r-1} dmin_{i_j i_{j+1}}$ so that $\lambda(p(v,w)) \leq \vartheta(p(v,w;t))$ for all $t \geq 0$ and a shortest path is denoted by $p^0(v,w) = argmin_{p(v,w)}\lambda(p(v,w))$.

3.1.2 Distributed Computations

Distributed computing requires local computations (at sites) as well as a communication overhead (between sites). The local computations adhere to region covers, cf. [KS98]. Other than partitions, regions of a region cover do overlap. In fact, this is necessary since all edges should belong to a region.

Definition 3.2 (region cover) *Let* $G = (V,E)$ *be a connected graph. A collection* $\mathcal{C} = \{C_1,\ldots,C_\mu\}$ *of subsets or classes* C_i *of vertices from* V *is called a region cover of* V *if it satisfies the following conditions:*

1. $\bigcup_{i=1}^{\mu} C_i = V$.
2. $\bigcup_{i=1,\, i\neq k}^{\mu} C_i \neq V \;\forall\, k = 1,\ldots,\mu$.
3. $\bigcup_{i=1}^{\mu} E(C_i) = E$.

Condition 1 means that $\mathcal{C}$ is a cover of V and condition 2 means that each class C_i contains a vertex which is not contained in all other classes united. Condition 3 means that a path's transition from one class to another occurs only at vertices that are common to both classes ($E(C_i) = \{(i,j) \in E \mid i,j \in C_i\}$). Hence from condition 3 alone there follows that a region cover can never be a partition of V except in the trivial case $\mu = 1$.

The classes C_i of a region cover are supposed to be those subgraphs for which fastest paths can be computed locally by a Dijkstra like algorithm. We make use of the so-called trace.

Definition 3.3 ($\mathcal{C}$-trace) *A trace or* $\mathcal{C}$*-trace of a path is a sequence of classes containing the vertices of a path in the order of their appearance where a class is only changed if necessary.*

For example, a path $i_0, \ldots, i_k$ has trace (C_7, C_8, C_3) if $i_0, \ldots, i_{s_1} \in C_7$, $i_{s_1+1}, \ldots, i_{s_2} \in C_8 - C_7$, and $i_{s_2+1}, \ldots, i_k \in C_3 - C_8$. A trace of a path is generally not unique, but it is in a two region cover. Computing the trace of a fastest or reasonably fast path based on aggregated information appears to be the core problem of distributed path computations.

In the sequel, a fastest path will be concatenated from partial paths stemming from different computations steps (over different classes).

3.1.3 Computing Traces from an Aggregated Viewpoint

The intersection graph $G_{\mathcal{IC}}$ of all non-empty intersections of regions is introduced to enable the computation of traces which ideally coincide with traces of fastest paths. The intersection graph allows to calculate candidate traces for fastest paths given estimates of single travel times.

Definition 3.4 (Intersection Graph) *The intersection graph* $G_{\mathcal{IC}} := (V_{\mathcal{IC}}, E_{\mathcal{IC}})$ *is defined by having the vertex set* $V_{\mathcal{IC}} := \{v_{C_i \cap C_j} | \, C_i, C_j \in \mathcal{C} \text{ with } C_i \cap C_j \neq \emptyset\}$ *and the edge set* $E_{\mathcal{IC}} := \{(v_{C_i \cap C_j}, v_{C_k \cap C_l}) | \; \{C_i, C_j\} \cap \{C_k, C_l\} \neq \emptyset \text{ and } \{C_i, C_j\} \neq \{C_k, C_l\}\}$.

The intersections of distinct classes receive different vertices even if the intersections should be equal. The edge set specification contains the condition "$\{C_i, C_j\} \neq \{C_k, C_l\}$" ensuring that the intersection graph contains no loops. Each class of $\mathcal{C}$ is represented in $G_{\mathcal{IC}}$ by the clique of all its intersections with other classes.

$G_{\mathcal{IC}}$ receives constant edge labels by assigning edge $(v_{C_i \cap C_j}, v_{C_k \cap C_l})$ the value $\min_{v_0 \in C_i \cap C_j, w_0 \in C_k \cap C_l} \lambda(p^0(v_0, w_0))$, cf. Section 3.1.1. Thereby time independent labels are given for each edge.

Candidate traces for a fastest path from $v \in C_1$ to $w \in C_\nu$ can be computed as k shortest paths in $G_{\mathcal{IC}}^{(v,w)}$, an extension of the intersection graph resulting from insertion of vertices $v \in C_1$ and $w \in C_\nu$ into $V_{\mathcal{IC}}$ and insertion of all edges $(v, v_{C_1 \cap C_i})$ with $C_1 \cap C_i \neq \emptyset$ and all edges $(v_{C_i \cap C_\nu}, w)$ with $C_i \cap C_\nu \neq \emptyset$ into $E_{\mathcal{IC}}$. An example is given by figure 3.1.

Computation of k shortest paths in the intersection graph $G_{\mathcal{IC}}^{(v,w)}$ and distributed computation of a fastest path for the respective traces can be used to filter for the fastest paths in G, if no region needs to be visited twice, i.e. if for any class C_i there is at least one fastest path from any $v \in C_i$ to any $w \in C_i$ starting from an arbitrary moment onward and consisting only of vertices in C_i (closing condition).

Using the facts, that

1. every fastest path for a given trace gives an *upper bound* for the travel time of the fastest path on the graph G,

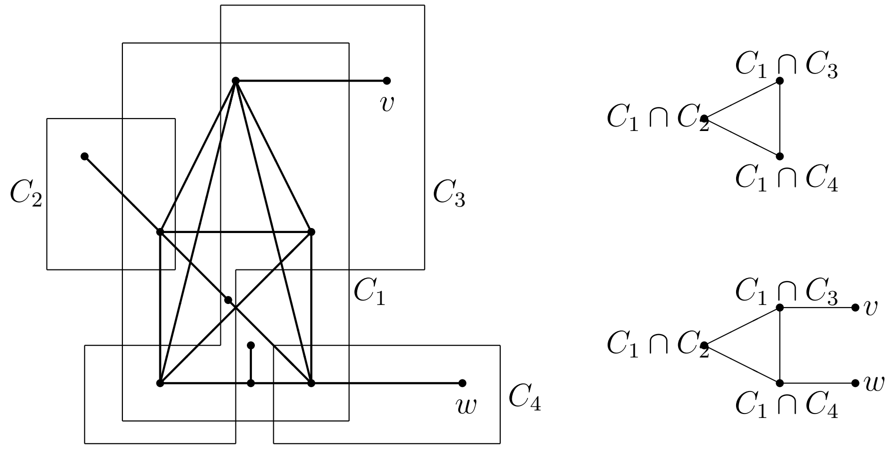


Figure 3.1: Graph with region cover (left), intersection graph $G_{\mathcal{IC}}$ (top right) and extended intersection graph $G_{\mathcal{IC}}^{(v,w)}$ (bottom right)

2. the length of a path on the extended intersection Graph $G_{\mathcal{IC}}^{(v,w)}$ gives a *lower bound* for the travel time of the fastest path using the corresponding trace,

algorithm **DFP** (**D**istributed **F**astest **P**ath) computes $p^0(v, w; t_1)$. While looping through the k shortest traces ($k = 1, 2, \ldots$), λ_{max} (an upper bound for the best travel time) keeps the minimum travel time of the fastest path found with any of the previous traces.

INPUT: graph G, region cover $\mathcal{C}$, start v, destination w, start time t_1
OUTPUT: fastest path $p^0(v, w; t_1)$ in G

1. Initialization: $k = 0$, $\lambda_{max} = \infty$
2. REPEAT
   (a) $k = k + 1$
   (b) Compute k shortest path in $G_{\mathcal{IC}}^{(v,w)}$ and corresponding trace T
   (c) Compute fastest path $p(v, w; t_1)$ from v to w starting at t_1 and with fixed trace T
   (d) Update λ_{max} if newly computed fastest path is faster than λ_{max}, i.e. $\lambda_{max} = min(\lambda_{max}, \vartheta(p(v, w, t_1)) - t_1)$

   UNTIL k shortest path in $G_{\mathcal{IC}}^{(v,w)}$ is greater than λ_{max}
3. RETURN path from v to w starting at time t_1 with length λ_{max}

The core of computation is the trace generation. We show an approach for trace generation by computing k shortest paths in aggregated graphs.

Fastest paths computations can be used to find upper and lower time bounds for all vertices (esp. in intersections) for certain objectives dominated by travel time.

Specifically, algorithm **DFP** can be adapted in order to compute multiple paths from v to w. By this, algorithm **DFP** can be used for the computation of a set of relevant[1] routes from location l_{start} (adhering to v) to location l_{dest} (adhering to w). This route set will be denoted by $R(l_{start}, l_{dest})$.

[1]E.g. all routes requiring up to Δt time more than the best route.

3.2 Transition Probabilities

Notification planning as proposed in this thesis requires a model for state change governed by time-dependent transition probabilities $p_{ij}(t, t')$. A transition probability $p_{ij}(t, t')$ denotes the probability of information state j at time point t' when the information state was i at time point t. The transition probabilities model an information state change and will be used in Section 4 for notification planning.

Transition probabilities can be assigned based on different methods. They can be based on expert's advice, on a mathematical model or on empirical evidence. However, they will be subjective in most cases. Our method for the assignment of transition probabilities is based on empirical data.

We employ a so-called event history which holds all relevant events recorded up to current time. The relative frequencies of similar situations from the past are used as estimates for the transition probabilities. Similar situations are to be described and detected in the event history.

3.2.1 The Event History

The event history lists all events that have occurred up to a time point t. It is denoted by $H(t)$. $H = H(t_{clock}$ denotes the current event history. Events occur at the time they enter the event history:

Definition 3.5 (Event History) *$H(t)$ is an event history, iff*

1. *$\forall_t \forall_{e \in H(t)} : e$ is an event.*
2. *for any duration $d > 0$ and for any time point t*

$$H(t) \subseteq H(t + d) \tag{3.1}$$

In other words, a time-dependent set of events is an event history if events are added in the course of time and never deleted. The occurrence time of an event is the time of its first appearance in the event history and is denoted by $e.t$. Note, events are always unique, distinguished by their occurrence time.

Example 23 (Event History) *Let e denote a congestion warning on a road segment (i, j) in the transportation network. A time-dependent state variable $s_{ij}(t)$ with domain $S_{ij} = \{congested, free\}$ has a state transition from $free$ to congested at time point $e.t$ if the state was $free$ prior to time point $e.t$. The occurrence of e at time $e.t$ is recorded in the event history according to $e \in H(t)$ for $t \geq e.t$ and $e \notin H(t)$ for $t < e.t$.*

3.2.2 Event Classes and Counting

Events are denoted $e_1, e_2, \ldots$. Event classes are denoted $E_1, E_2, \ldots$. If an event e_i is an occurrence of an event class E_j, then we annotate $e_i \in E_j$ as an *element-relationship*. This relationship is non-exclusive, i.e. $e_i \in E_j$ and $e_i \in E_k$ is possible even if $E_j \neq E_k$.

Example 24 (Event Class for Congestion Warning) *Event class E_{warn} denotes all occurrences of congestion warnings wherever they happen in the transportation network, while event class E^{ij}_{warn} denotes occurrences of congestion warnings for the specific road segment (i, j). Therefore, all occurrences of E^{ij}_{warn} are also occurrences of E_{warn} and the first can thus be viewed as a specialization of the latter. A change of state variable $s_{ij}(t)$ (cf. Example 23) from* free *to* congested *is fully represented by event class E^{ij}_{warn}.*

Example 25 (Event Class for Train Departure) *Event class E_{depart} denotes all occurrences of train departures, while event class $E^{train-id}_{depart}$ denotes occurrences for certain train departures only, namely departures with a specific* train-id. *Note, $E^{train-id}_{depart}$ can still have several occurrences if the same* train-id *is used on different days. Other than that, a sub-state of the information state usually refers to a specific departure on a certain day.*

The occurrences of event classes have to be counted.

Definition 3.6 (Event Count) *The event count $C_t(E)$ is the number of occurrences of E in $H(t)$.*

$C(E)$ denotes the event count of E in the current event history $H = H(t_{clock})$. The following *no purging* condition holds:

$$\forall_{t,d \geq 0}\, C_t(E) \leq C_{t+d}(E) \tag{3.2}$$

.

The event count will be used for the computation of relative frequencies of certain situations and thus for the estimation of transition probabilities.

3.2.3 Composite Events and Event Assignment

We distinguish *primitive* and *composite event classes* (e.g. missing the train after the train has left). Occurrences of primitive event classes are the elements of the event history, while composite event classes specify combinations of one or more primitive events.

We need the following operators to specify composite events, cf. Hinze and Voisard [HV02]:

- *Negation:* $\neg E$ occurs whenever E does not occur.
- *Sequence:* $(E_1, E_2)_d$ occurs whenever an occurrence of E_1 is followed by an occurrence of E_2 within duration d. (E_1, E_2) is a shortcut for $(E_1, E_2)_\infty$. Note that instances of $(E_1, \neg E_2)_d$ are occurrence of E_1 not followed by an occurrence of E_2 within duration d.

In order to specify composite event classes, we employ the definition symbol ":=".

Example 26 (Composite Event Specification) *Let a train departure event be specified as $E^{ICE2001}_{depart}$. Its scheduled departure occurs at a specific time according to the time table and is specified as $E^{ICE2001}_{sched}$. Now, the event of a delayed departure is specified as a composite event class by $E^{ICE2001}_{delayed} := ((E^{ICE2001}_{sched}, \neg E^{ICE2001}_{depart})_5, E^{ICE2001}_{depart})$ (for a delay of 5 minutes), i.e. the time event of its scheduled departure followed by no train departure within 5 minutes later, and followed by the train departure.*

The occurrence time of composite events is exactly the time point at which the occurrence can be observed in the event history for the first time.

The selection of primitive event occurrences for a single composite event occurrence can be driven by semantic issues. Other than in example 26, where event classes referred to a single departure only, the event class E_{sched} specifies the time events of scheduled train departures, while event class E_{depart} specifies the actual train departures. Selected occurrences of E_{sched} and E_{depart} refer to the same train departure, e.g. *train-id*$= ICE2001$. For the specification of a composite event class, whose instances are composed of primitive events with matching *train-id*s, we introduce the concept of *event assignment*.

Definition 3.7 (Assigned Events) *Two events e_1, e_2 are called <u>assigned</u> ($e_1 \sim_{att} e_2$), if they have identical values for certain predefined attributes att.*

If E is a composite event class, then $(E)^{att}$ is another composite event class with the additional constraint that all occurrences are combinations of primitive events with identical values for attributes *att*.

Example 27 (Event Assignment) *The composite event E_{warn} of a train delay warning which happens up to* 15 *minutes before a scheduled departure E_{sched} and a train arriving at least 10 minutes late (departure event E_{depart}) is annotated by*

$$E_{composite} := (((E_{warn}; E_{sched})_{15}; \neg E_{depart})_{10}; E_{depart})^{train-id} \tag{3.3}$$

By this, any set of instances of E_{warn}, E_{sched} and E_{depart} which constitutes an occurrence of $E_{composite}$ refers to the same train with a given attribute value for train-id.

3.2.4 Transition Probability Estimation

To estimate information state transition probabilities, earlier occurrences of the same state transition need to be sampled from the event history.

Example 28 (Transition Probability Estimation) *Let* $s_{ICE2001}(t)$ *denote the state variable for a single train departure with domain* $S_{ICE2001} = \{initial, normal, delayed, ontime, late\}$. *Prior to the trains departure time, the state is either* normal *(no delay warning occured) or* delayed*. 5 minutes after the trains departure time, the state is either* ontime *(the train has departed) or* late *(the train has not departed yet). We introduce an artificial time event indicating the start of observation 60 minutes prior to the scheduled departure time. By start of observation, the state changes from* initial *to* normal *and may only thereafter be changed to* delayed *by a delay event.*

Let t *denote the time point 15 minutes prior to departure time and* t' *denote the time point 5 minutes after the train departure. Now, the transition probability* $p_{normal,late}(t, t')$ *is to be computed, i.e. the probability of having a late departure 5 minutes after the train departure when having had no delay warning 15 minutes prior to the train departure.*

We assume prior knowledge about the events which can be compared for empirical analysis. For transition probability $p_{s,s'}(t, t')$, we assume event class E to reflect similar situations for having state s at time t and we assume event class E' to reflect similar situation for having state s' at time t'.

Example 29 (Transition Probability Estimation (cont'd)) *Let* $E_{announce}$ *denote all delay announcements for trains.* E_{start} *denotes the start of observation for any specific train 60 minutes prior to the scheduled train departure.* E_{depart} *denotes actual departures of trains. The* train-id *is an attribute specifying a specific train departure. Then, the following composite event classes can be considered for transition probability* $p_{normal,late}(t, t')$:

- $E := (E_{start}, \neg E_{announce})_{45}^{train-id}$ *denoting situations, where no delay announcement has occurred 60-45=15 minutes prior to the scheduled train departure and*
- $E' := ((E_{start}, \neg E_{announce})_{45}, \neg E_{depart})_{20}^{train-id}$ *denoting situations, where no delay announcement has occurred 15 minutes prior to the scheduled train departure and the train has not departed until 5 minutes after the scheduled train departure.*

In order to compute the relative frequency of E' versus E we need to ensure, that an occurrence of E' implies an occurrence of E. For this, the concept of a partial event class is introduced and based on the concept of partial occurrence.

Composite event occurrences $e \in E$ are annotated by $e = (e_1, \ldots, e_n; t)$ where $e_1, \ldots, e_n \in H$ are the primitive events constituting the composite event and $t = e.t$ is the occurrence time of the composite event e.

Example 30 (Composite Event Occurrence) *An event class $E_{late} := (E_{sched}; \neg E_{depart})_{10}$ denotes the composite event of the train arrival not having happened up to 10 minutes after the scheduled departure time. Now, $e_{sched} \in E_{sched}$ has occurred at t_{sched} and E_{depart} did not occur within 10 minutes after t_{sched}. Thus, $e_{late} = (e_{sched}; t_{sched} + 10)$ is an occurrence of E_{late}, i.e. $e_{late} \in E_{late}$.*

Definition 3.8 (Partial Occurrence) *Let $e_1 = (es_1; t_1)$ and $e_2 = (es_2; t_2)$ denote composite events. $(es_1; t_1)$ is a partial occurrence of $(es_2; t_2)$, iff*

- *$\exists_{es_3}\, es_2 = es_1 \cdot es_3$, i.e. event sequence es_1 is a prefix of event sequence es_2,*
- *$t_1 \leq t_2$, i.e. event e_1 occurs (is observed) prior to event e_2.*

Note, the second (time) condition is needed in Definition 3.8 in order to make this relationship irreflexive. Consider the following example.

Example 31 (Partial Occurrence) *Consider Example 30. Another event class $E'_{late} := (E_{sched}; \neg E_{depart})_5$ denotes a late arrival of 5 minutes. $e'_{late} \in E'_{late}$ is a partial occurrence of e_{late} but not vice versa.*

Now, a partial event class can be defined. Informally, an event class E_1 is a partial event class of event class E_2, if every occurrence of E_2 has started with a partial event occurrence in E_1.

Definition 3.9 (Partial Event Class) *Let E_2 be a composite event class. E_1 is a partial event class of E_2 iff any $e_2 \in E_2$ has a partial occurrence $e_1 \in E_1$.*

Now we define the event completion probability $p_c(E, E')$ for event classes E and E'.

Definition 3.10 (Event Completion Probability) *Let E and E' be composite event classes with E is a partial event class of E'. The event completion probability $p_c(E, E')$ is given by the relative frequency of the completion, i.e.*

$$p_c(E, E') = \frac{C(E')}{C(E)} \tag{3.4}$$

Example 32 (Event Completion) *Consider primitive event classes E_{sched} and E_{depart} for the scheduled departure and real departure of trains. The attribute train$-$id specifies a specific departure. The following composite event classes are considered:*

- $E_{late5} := (E_{sched}, \neg E_{depart})_5^{train-id}$, *the event of 5 minutes delay,*
- $E_{late10} := (E_{sched}, \neg E_{depart})_{10}^{train-id}$, *the event of 10 minutes delay.*

Event class E_{late5} is a partial event class of E_{late10}. We look at the occurrence times of the primitive and composite event classes for different values of $train - id$:

$train - id$	E_{sched}	E_{depart}	E_{late5}	E_{late10}
1	*10:05*	*10:11*	*10:10*	-
2	*10:20*	*10:22*	-	-
3	*11:05*	*11:17*	*11:10*	*11:15*
4	*11:20*	*11:29*	*11:25*	-

The event counts for the composite event classes can be observed from the table: $C(E_{late5}) = 3$ and $C(E_{late10}) = 1$. Thus, the event completion probability is calculated as follows:

$$p_c(E_{late5}, E_{late10}) = \frac{C(E_{late10})}{C(E_{late5})} = \frac{1}{3} \tag{3.5}$$

This value can be used as the estimated probability of the train departure to be 10 minutes late conditioned on the fact, that it is already 5 minutes late.

If E refers to $s(t) = s$ and E' to $s(t') = s'$, then $p_c(E, E')$ is an estimate of the transition probability $p_{ss'}(t, t')$.
This will be demonstrated in the next two sections for both scenarios.

3.2.5 Road Scenario Revisited

Here, we will look at the state variable $s_{ij}(t) \in S_{ij} = \{\text{free}, \text{congested}\}$ for a single route segment (i, j). The state will be influenced by congestion warnings and cancellations, i.e. the following event classes are considered as constituents for the composite events to be studied:

- *Congestion Warning:* Let E_{con} be the event class for congestion warnings.
- *Cancellation:* Let E_{can} be the event class for cancellations of congestions warnings.

Now, we consider $s_{ij}(t) = congested$ and $s_{ij}(t') = free$ with $t < t'$. Let t_{last} denote the last occurrence time of E_{con} prior to t. Then, the composite event class $E = (E_{con}, \neg E_{can})_{t-t_{last}}$ classifies the situation $s_{ij}(t) = congested$ and the composite event class $E' = ((E_{con}), \neg E_{can})_{t-t_{last}}, E_{can})_{t'-t}^{edge}$ classifies a situation $s_{ij}(t') = free$ following $s_{ij}(t) = congested$. The transition probability is estimated by the relative frequency, i.e. $\hat{p}_{congested,free}(t, t') = p_c(E, E') = \frac{C(E')}{C(E)}$.

3.2.6 Train Scenario Revisited

Here we will look at the state $s_x \in S_x = \{\text{initial}, \text{delayed}, \text{normal}, \text{ontime}, \text{late}\}$ for a specific train departure, i.e. the index x specifies one concrete departure including the train number, day and time of departure on route segment (i, j). The state value *initial* is reserved for a situation prior to the start of observation (which will be signaled by an event).

The following primitive event classes are considered as constituents for the train scenario:

Delay Warning: E_{warn} represents a delay warning occurring prior to the train departure.

Scheduled Departure: E_{sched} represents the scheduled departure of a train.

Actual Departure: E_{depart} represents the actual departure of a train.

Start Observation: E_{start} represents an event occurring at a fixed interval prior to a scheduled departure.

Note that these event classes are defined for any train departure. Timely and late departure are defined as composite event classes with attribute $train-id$ for the assignment of events belonging to the same departure.

Timely Departure: $E_{ontime} := (E_{sched}; E_{depart})_5^{train-id}$ represents the event of a train departure being less than or equal to five minutes after the scheduled departure time point.

Late Departure: $E_{late} := (E_{sched}; \neg E_{depart})_5^{train-id}$ represents the event of a train departure being more than five minutes after the scheduled departure time point.

Now, we consider $s_x(t) = normal$ and $s_x(t') = late$ with $t < t'$. Let t_{sched} denote the scheduled departure time of the train and t_{start} denote the time point when the observation starts. Then, the composite event class $E = (E_{start}, \neg E_{warn})_{t-t_{start}}^{train-id}$ classifies a situation $s_x(t) = normal$ and composite event class $E' = ((E_{start}, \neg E_{warn})_{t-t_{start}}, E_{late})_{t'-t}^{train-id}$ classifies a situation $s_x(t') = late$ following $s_x(t) = normal$. The transition probability is estimated by the relative frequency, i.e. $\hat{p}_{normal,late}(t, t') = p_c(E, E') = \frac{C(E')}{C(E)}$.

3.2.7 Summary

We gave a definition of the conditional probability of an event completion. Transition probabilities $p_{ss'}(t, t')$ are estimated by the specification of composite events modelling similar situations and subsequent assessment of the event completion probability. This corresponds to the counting of relative frequencies for the occurrence of state s' at time t' after occurrence of state s at time t.

Transition probabilities are subsequently employed for the solution of notification problems (cf. Section 4).

Chapter 4

Influence Diagrams for Notification Planning

In this chapter, the decision models for planning of notifications with options are studied and represented as influence diagrams. Influence diagrams are a powerful technique for the combined representation of

- dependencies between random variables (known as Bayesian networks) and
- dynamic decision problems, where decisions may depend upon given evidence.

Influence diagrams where originally proposed by Howard and Matheson [HM81] as a compact representation for decision trees. A few years later, Shachter [Sha86] proposed a method for direct evaluation of influence diagrams without explicit decision tree analysis. Shachter's approach is restricted to directed networks with sequentially ordered decisions. But this is sufficient for our application. Therefore, we have not searched for other approaches.

4.1 Influence Diagrams

We want to use influence diagrams for the representation of *current decisions* for notification planning, i.e. notification problem, current time, location and information state are given. A short but comprehensive introduction to influence diagrams will be given here.

Influence diagrams are directed graphs with three types of nodes (cf. Howard [HM81], Shachter [Sha87] and Pearl [Pea91]). Chance nodes (shown as ovals) represent random variables, decision nodes (shown as rectangles) represent decision options and value nodes (shown as diamonds) represent rewards or costs dependent upon decisions and outcomes of uncertain quantities. Directed edges leading to chance nodes denote conditional dependencies, directed edges leading to value nodes denote functional dependencies and directed edges leading to decision nodes are informational, i.e. the values of the corresponding chance nodes are known before a decision is made.

Before an influence diagram can be used, the following information has to be provided:

- *Chance nodes:* Chance nodes without incoming edges have a so-called marginal distribution. This is a unconditioned probability distribution. For chance nodes with incoming edges, a conditional probability distribution must be applied, i.e. a probability distribution conditioned on any interaction of values of preceeding decision nodes or chance nodes, respectively.

- *Decision nodes:* Only the possible outcomes of the decision needs to be specified, i.e. the domain of the decision variable.

- *Utility nodes:* For any combination of values from incoming decision and chance nodes, the respective utility has to be modelled.

Evaluation of influence diagram implements Bayesian inference for uncertain quantities and exploits efficient algorithms for local computations from Lauritzen and Spiegelhalter [LS88]. Values of decision nodes are determined (usually by dynamic programming) in such a way as to optimize the total expected value over all value nodes.

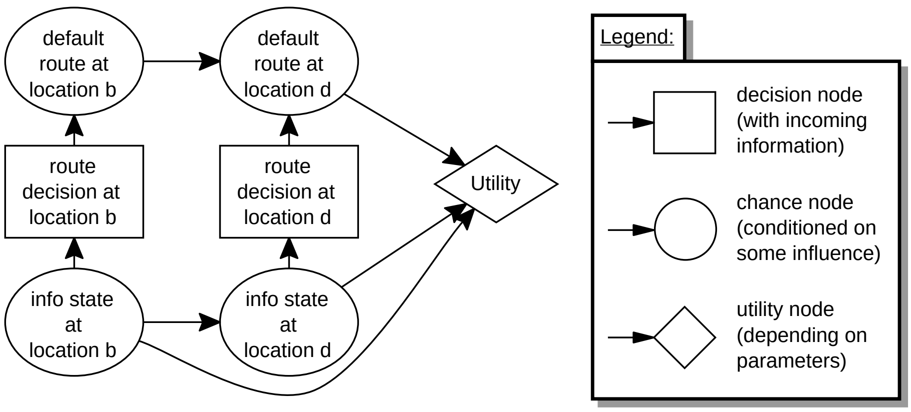


Figure 4.1: Influence Diagram Example

The graphical notation of influence diagrams is demonstrated by example in Fig. 4.1. The example refers to the road scenario. Consider for instance the route decision at location d. This decision is taken with respect to the information state at location d (denoted by the incoming edge from the respective information state node) and it influences the traveller's default route. Note, that the graph of an influence diagram is only part of an influence diagram model. Marginal and conditional probability distributions for the chance nodes are to be added.

4.2 Notification Planning

The concept of notification planning has been introduced in Section 2.11.2. Now we define the single steps for the transformation of a concrete problem instance into an influence diagram.

A notification problem is given by the traveller's start location l_{start}, destination l_{dest} and utility function u. A start-destination route set $R(l_{start}, l_{dest})$ is computed in a pre-processing step by an algorithm for fastest paths, e.g. the Distributed Fastest Path (DFP)-algorithm developed for the case of distributed data sources (cf. Section 3.1).

A notification decision occurs at certain time points and is characterized by a current pre-route r_{curr}, a current time point t_{curr} and a current information state $s_{curr} = s(t_{curr})$. A set NO of notification options is provided, cf. Def. 2.28.

Therefore, the following input is given for a notification decision:

- The start-destination route set $R(l_{start}, l_{dest})$,
- the traveller's utility function u,
- the current pre-route r_{curr}, time t_{curr}, information state s_{curr},
- the set of notification options NO.

The following data structures have to be computed prior to the translation into an influence diagram:

- The pre-route tree $(R, <_{pre})$ of $R(l_{start}, l_{dest})$, cf. Definition 2.24.
- The so-called extended[1] current pre-route tree, i.e. the subtree of $(R, <_{pre})$ which is rooted at the current pre-route r_{curr}.
- For each pre-route $r \in R$ the set $T_{no}(r)$ of time points where at least one notification option can be effective[2], i.e. $T_{no}(r) = \{no.t \mid no \in NO \wedge r \subseteq no.r\}$.

Example 33 (Pre-Processing of Influence Diagrams) *Fig. 4.2 shows an example of a transportation network with 6 locations. The notification problem is given by start location $l_{start} = a$ and destination $l_{dest} = e$. The current pre-route is given by $r_{curr} = (a, b, b')$. The start-destination route set is given by $R(l_{start}, l_{dest}) = \{A, B, C\}$ with $A = (a, b, b', d, c, e)$, $B = (a, b, c, d, e)$ and $C = (a, b, b', d, e)$.*

[1]Not every current route is a decision point. Therefore, the current pre-route needs to be inserted.

[2]A notification option can be effective only if the current pre-route is a sub-route of the notification content.

The pre-route tree for this example is shown in Fig. 4.3(a) with indication of the next pre-route (after the current pre-route). The resulting extended current pre-route tree is shown in Fig. 4.3(b).

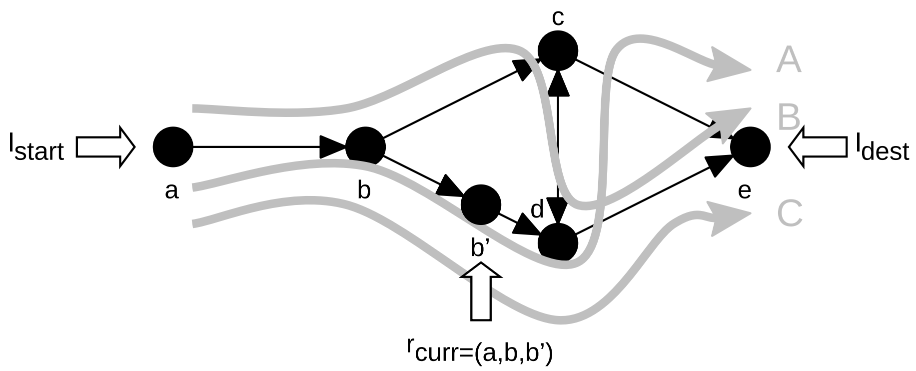
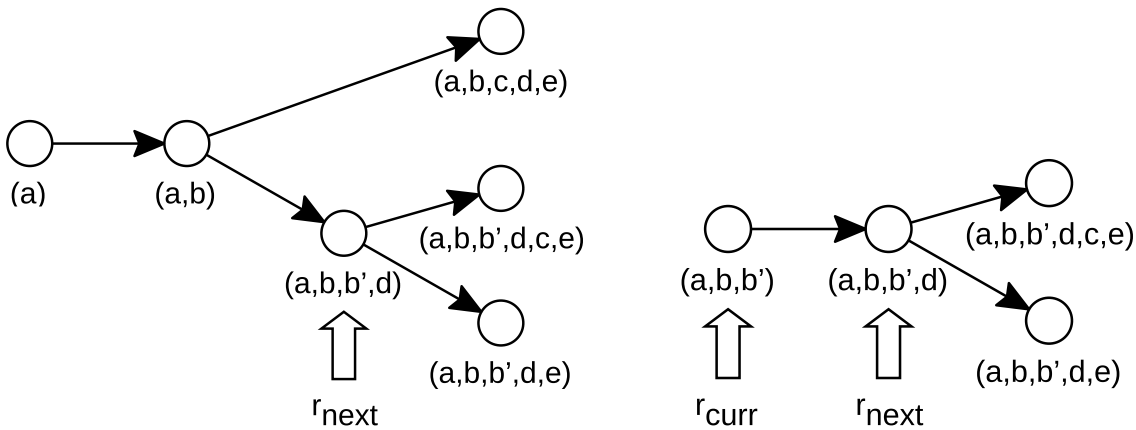


Figure 4.2: Route Set for Example 33



(a) Pre-Route Tree

(b) Extended Current Pre-Route Tree

Figure 4.3: Pre-Route Trees for Example 33

We give construction principles for influence diagram models in Section 4.3. The method of notification planning with influence diagrams is demonstrated for both running examples, i.e. the *train scenario* and the *road scenario* in Sections 4.4 and 4.5.

4.3 The Influence Diagram Model

We model and solve decision problems arising from notification planning by using influence diagrams.

For each node in the extended current pre-route tree, chance nodes (in the influence diagram) are used for modelling the information state, default route and arrival time. In the case of decision points the choice of alternative notification contents is modelled by a decision node (in the influence diagram). If several notification time points are to be considered, an extra decision node is used for selecting the right notification time point. Chance nodes and a decision node per pre-route form a so-called building block of an influence diagram.

These building blocks are to be connected according to the pre-route tree structure. The marginal and the conditional probabilities are to be assigned appropriately.

These issues are covered in the following sections.

4.3.1 The Nodes of the Pre-Route Tree

Each node (or pre-route) of the extended current pre-route tree is modelled by the uncertain *default route*, the uncertain *information state* and the uncertain *arrival time* after having travelled along this route. A single time point is associated to each pre-route and has one of the following three types:

- the current time t_{curr}, if pre-route r is the root node,
- a notification time point $t_{no}(i) \in T_{no}(r)$, if pre-route r is an inner node and represents a decision point i,
- the estimated arrival time, if the other types don't apply.

The influence diagram model of a pre-route tree node is given below for different cases.

4.3.2 Decision Points with a Single Notification Time

A decision point i is given by a pre-route r which leads to i. We consider the case that there is exactly one notification time point $t_{no}(i) \in T_{no}(r)$.

For the decision point i, the following quantities and one decision are to be modelled as nodes:

- $\mathbf{r}_i$ is the uncertain *default route* at decision point i, modelled by a chance node.
- $\mathbf{s}_i$ is the uncertain *information state* at decision point i, modelled by a chance node.

- $\mathbf{t}_i$ is the uncertain *arrival time point* at decision point i, modelled by a chance node.
- δ_i is the *route decision* at decision point i, modelled by a decision node.

This influence diagram building block is depicted in Fig. 4.4.

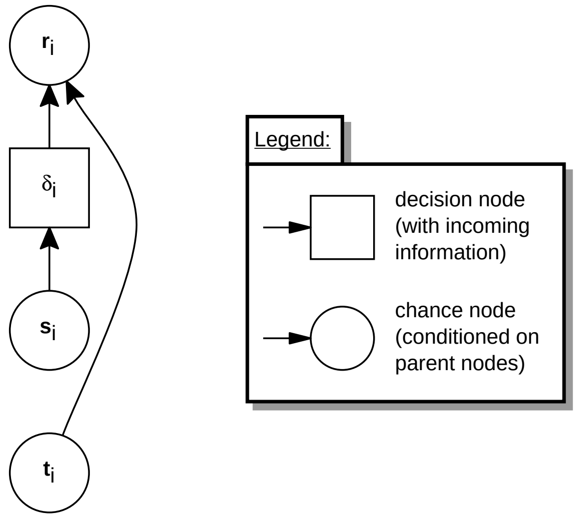


Figure 4.4: An Influence Diagram Building Block for a Decision Point i

The domains of the uncertain quantities and the decision are given by:

- $dom(\mathbf{r}_i) = R(l_{start}, l_{dest})$. The set of all routes leading from start to destination.
- $dom(\mathbf{s}_i) = S$. The information state space for the transportation network.
- $dom(\mathbf{t}_i) = T_i$. A finite set of time points given by the possible arrival times.
- $dom(\delta_i) = R_{no}(r)$. The set of all routes in $R(l_{start}, l_{dest})$ having pre-route r.

The edges within this building block have the following semantics:

- $\mathbf{s}_i \rightarrow \delta_i$: An informational edge, the value of $\mathbf{s}_i$ is known, when the decision δ_i has to be made.
- $\delta_i/\mathbf{t}_i \rightarrow \mathbf{r}_i$: The value of $\mathbf{r}_i$ is influenced by the decision δ_i if the notification time point $t_{no}(i)$ is earlier (smaller) than the arrival time point $\mathbf{t}_i$ and if the previous default route also had pre-route r. The default route remains unchanged otherwise (cf. Section 4.3.5).

Note, there is no informational edge from time $\mathbf{t}_i$ to route decision δ_i, i.e. the time of arrival is unknown even when the traveller is near to the respective decision point. This is justified since the information system does not have information about the location of the traveller in our model.

4.3.3 Decision Points with Multiple Notification Times

In this section, several notification time points $t_{no}(i)$ are to be considered for the same decision point i. Only one notification time point is to be selected for notification. Therefore, an extra decision node will be introduced for the decision about the best notification time:

- select $t_{no}(i)$ is the decision about the notification time for decision point i.
- $dom(\text{select } t_{no}(i)) = T_{no}(r)$. The set of notification time points considered for pre-route r leading to decision point i.

The extension of the influence diagram in Fig. 4.4 is shown in Fig. 4.5. The additional edges have the following semantics:

- select $t_{no}(i) \rightarrow \mathbf{r}_i$: The default route can only be changed according to the route decision, if the selected notification time point $t_{no}(i) < \mathbf{t}_i$. Otherwise, the default route remains unchanged.
- select $t_{no}(i) \rightarrow \mathbf{s}_i$: The information state s_i depends on the notification time $t_{no}(i)$.

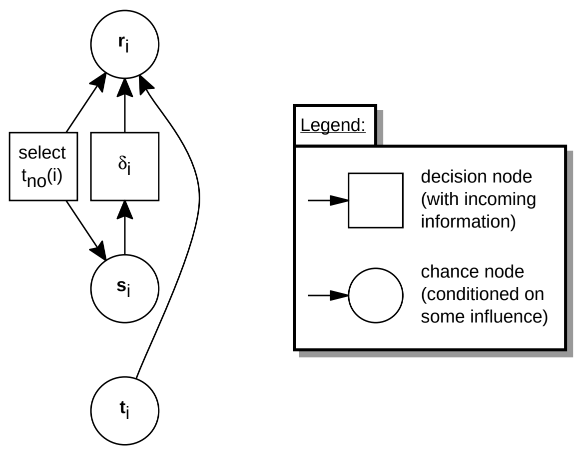


Figure 4.5: Decision Point with Decision about the Notification Time

4.3.4 Other Nodes of the Pre-Route Tree

In principle, all other nodes of the pre-route tree, i.e. leave nodes, a root node $r_{curr} \neq r_{next}$, and inner nodes without notification time points, can be modelled in a similar way (without a decision about the notification time point). The following specialities have to be observed:

- Leave nodes have no choice for the notification content, which is the current pre-route. Therefore, decision nodes can be omitted in the influence diagram building block for a leave node. The same holds true for root nodes without choices, i.e. for the case of $r_{curr} \neq r_{next}$, cf. Fig. 4.3(b).

- Inner nodes r without notification time points, i.e. with $T_{no}(r) = \emptyset$ can be dropped.

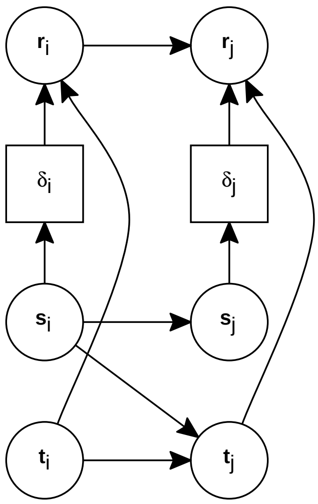
- If there is no decision node, then there is no need for an edge $\mathbf{t}_i \rightarrow \mathbf{r}_i$, the previous default route is simply copied to the current default route.

- Leave nodes are only needed for utility assessment. Therefore, the chance node for the uncertain information state can be omitted.

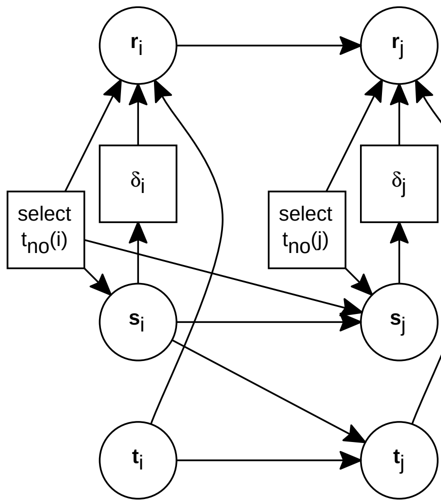
These specialities reduce the complexity of the resulting influence diagram.

4.3.5 The Edges of a Pre-Route Tree

For any edge in the pre-route tree, edges in the influence diagram are to be created between the building blocks according to the templates shown in Fig. 4.6(a) and Fig. 4.6(b).



(a) Single Notification Time Point



(b) Multiple Notification Time Points

Figure 4.6: Transformation of a Pre-Route Tree

The edges have the following semantics:

- $\mathbf{r}_i \rightarrow \mathbf{r}_j$: The default route remains unchanged if the notification time point $t_{no}(j)$ is later (greater) than the time point $\mathbf{t}_j$ or if the previous default route was not feasible[3].

- $\mathbf{s}_i \rightarrow \mathbf{s}_j$: The probability of information state $\mathbf{s}_j$ at notification time point $t_{no}(j)$ is conditioned on information state $\mathbf{s}_i$ at notification time point $t_{no}(i)$.

- $\mathbf{t}_i/\mathbf{s}_i \rightarrow \mathbf{t}_j$: The time point $\mathbf{t}_j$ of arrival at decision point j is conditioned on the time point $\mathbf{t}_i$ of arrival and on the information state at decision point i (cf. Section 2.8.3).

- $\mathsf{select}\ t_{no}(i) \rightarrow \mathbf{s}_j$: The notification time point selection $\mathsf{select}\ t_{no}(i)$ influences the uncertain information state $\mathbf{s}_j$ by replacing the first time point parameter of the respective transition probabilities.

Some of these edges will be omitted, if the respective chance or decision nodes are omitted.

4.3.6 Utility Assignment

Each leave node of the pre-route tree gets an utility assigned according to Fig. 4.7. Note, a positive utility $u(t)$ for a certain arrival time t at the destination can only be assigned if the default route $\mathbf{r}_i$ matches the pre-route represented by the respective leave of the pre-route tree. Therefore, there is an incoming edge from the default route $\mathbf{r}_i$.

Example 34 (The Influence Diagram Model) *We construct the influence diagram for Example 33. Indices* 1, 2, 3 *and* 4 *correspond to pre-routes* (a, b, b'), (a, b, b', d), (a, b, b', d, c, e) *and* (a, b, b', d, e) *respectively. This is depicted in Fig. 4.8(a). Notification time point selection is considered for decision point* 2, *but no notification time points are considered for all other nodes. The resulting influence diagram is shown in Fig. 4.8(b).*

Shadowed boxes are used for highlighting the building blocks of the influence diagram created per node of the pre-route tree. These boxes are not a part of the influence diagram model.

The influence diagram is used by assignment of evidence to the respective node of the pre-route tree. Usually, this is the root node which represents the actual time point.

[3]A default route is not feasible, if the pre-route leading to the decision point is not a pre-route of the default route. In this case, the traveller does not arrive at the respective decision point.

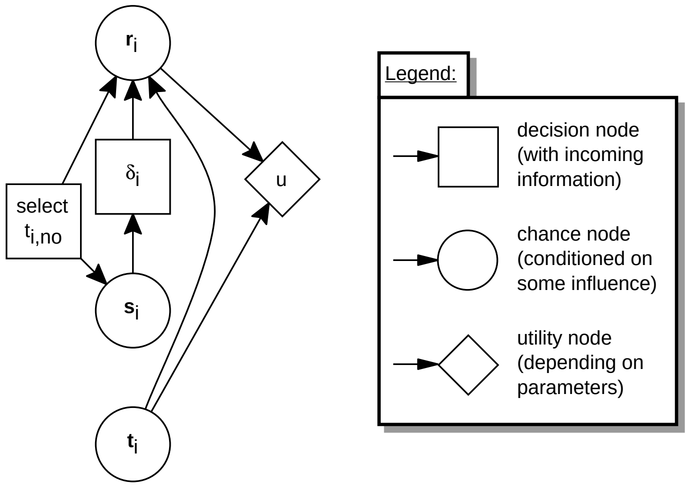


Figure 4.7: Adding Utility for Leave Node i

With little loss of accuracy, the influence diagram can also be used for subsequent decisions. In this case, the following values are assigned as evidences to the respective decision point i:

- *The information state $\mathbf{s}_i$ is the current information state s_{curr}.*
- *The default route $\mathbf{r}_i$ is a completion of the current pre-route r_{curr}.*
- *The arrival time $\mathbf{t}_i$ is predicted from information state s_{curr}, time t_{curr} and from information about the travellers location, if available.*

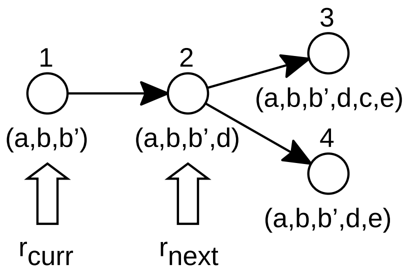
4.3.7 Related Work on Influence Diagrams

Reasoning with imperfect information has led to the theory of POMDP (Partially Observable Markov Decision Processes, cf. Hauskrecht [Hau97]). POMDP's are an extension of Markov Decision Processes, cf. e.g. [Bat00].

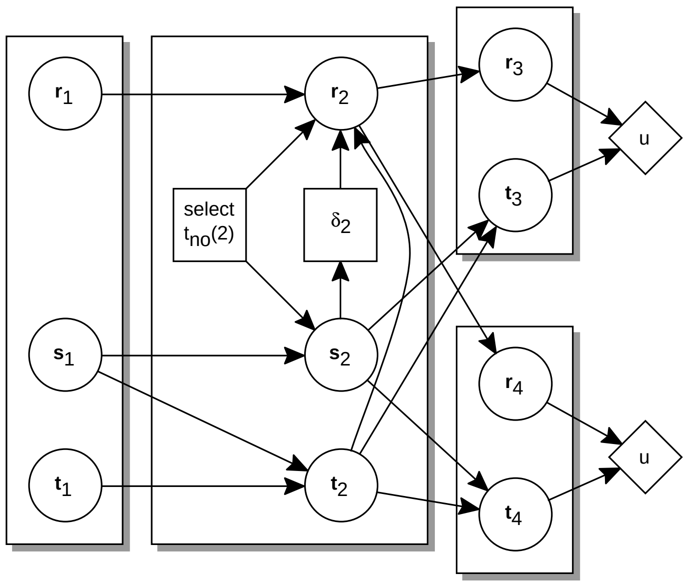
Time representation in Bayesian networks has led to various approaches. Berzuini [Ber90] introduced a network of dates in order to reason about the probabilistic nature of event occurrence times for medical applications. Temporal random variables and continuous time is used in this work.

Kjaerulff [Kja95] proposes a methodology and a tool for the analysis of dynamic time-sliced Bayesian networks.

Dean and Kanazawa [DKK92] propose random variables for duration as a means to represent semi-Markov processes in probabilistic networks. Tawfik and Neufeld [TN94] employ Temporal Bayesian Networks (TBN) for the representation of probabilities as functions of time. Arroyo-Figueroa and Sucar [AFS99] model



(a) Indexed Pre-Route Tree



(b) Resulting Influence Diagram

Figure 4.8: Constructing the Influence Diagram for Example 33

event occurrence times as nodes with respect to time intervals as developed by Allen [All83]. This last approach is actually very similar to Berzuini's approach, but it is restricted to a finite number of intervals for the occurrence time of events.

We employ temporal variables similar to [Ber90] and [AFS99]. However, we employ relative points in time in order to represent the influence of temporal distances on the conditional probabilities.

4.4 Train Scenario Revisited

The specific features of our model for the train scenario (cf. Scenario 1.4.2) are as follows:

1. **Initial Route Ignorance:** At current time, the traveller selects any route with an equal probability. Therefore, the default route is uncertain even at current time. In the influence diagram, this will be modelled by not having any preference or evidence of a current default route.

2. **Selection of Notification Time:** Only one notification is allowed here.

3. **Traveller Acceptance:** The traveller's acceptance of a notification declines while the decision time is coming nearer. This effect causes a continuous loss of options, since the notification issued by the information system looses control over the travellers decision as time passes by.

Note, these features provide a relaxation of the traveller's obedience assumption, cf. Section 2.11.

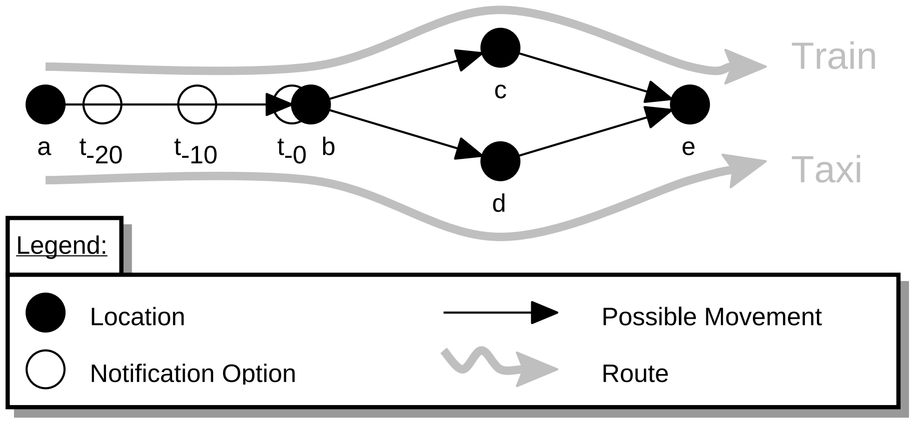


Figure 4.9: Routes for the Train Scenario (with Notification Options)

The route set for this scenario is depicted in Fig. 4.9 and consists of two routes branching at location b. Pre-route (a, b) leads to the only decision point. The notification options are temporally located 0, 10 and 20 minutes prior to the scheduled departure time of the train from location b.

For ease of understanding, time points are defined relative to scheduled time. The following discrete variables for time are used:

- Current time $t_{curr} \in \{-20, -10, 0\}$, i.e. 20, 10 or 0 time units prior to the scheduled departure of the train at b.

- Notification time point $t_{no} \in \{-20, -10, 0\}$, i.e. 20, 10 or 0 time units prior to the scheduled departure of the train.

The following decisions need to be taken:

- At current time, the best notification time point has to be selected.
- If the current time equals the best notification time, then the best route is selected and the traveller is notified immediately.

4.4.1 The Influence Diagram Model

The influence diagram for our example is shown in Fig. 4.10. A description of the nodes is given next. We use the simplified annotation train $= r_{train} = (a, b, c, e)$ and taxi $= r_{taxi} = (a, b, d, e)$.

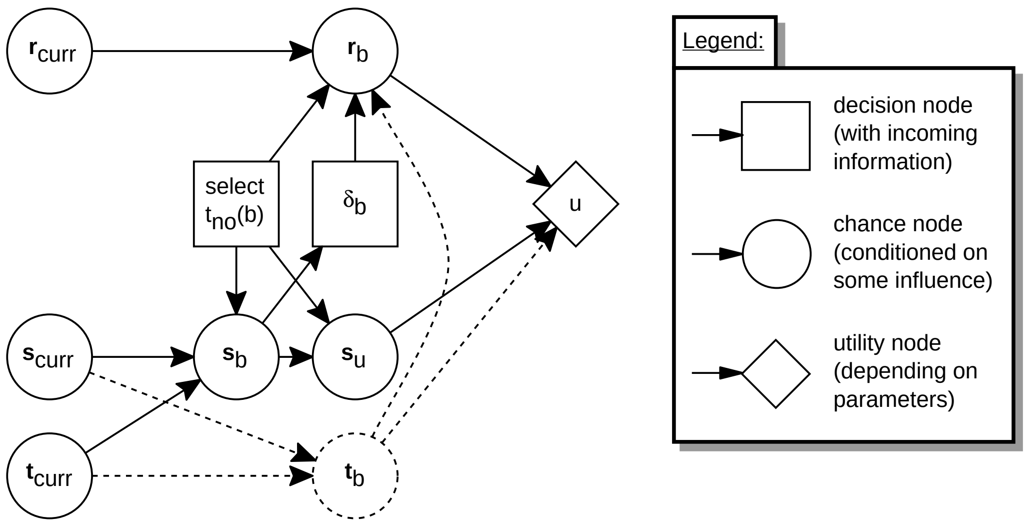


Figure 4.10: Influence Diagram for the Train Scenario

Chance Nodes:

$\mathbf{r}_{curr} \in$ {train,taxi}, the default route at current time (prior to notification time);

$\mathbf{r}_b \in$ {train,taxi}, the selected default route at pre-route (a, b) (with or without notification);

$\mathbf{s}_{curr} \in \{delay, none\}$, the current information state;

$\mathbf{s}_b \in \{delay, none\}$, the information state at notification time $t_{no}(b)$;

$\mathbf{s}_u \in \{delay, none\}$, the information state at the scheduled departure time $t_{sched} = 0$;

$t_{curr} \in \{$*-20/-10/0*$\}$, the current clock time;

$t_b = 0$, the time arrival for pre-route (a, b) is shown here for illustration, but dropped in the model for simplification;

Decision Nodes:

select $t_{no}(b) \in \{$*-20/-10/0*$\}$, decision about the notification time point, given relative to the scheduled departure time;

$\delta_b \in \{\text{train,taxi}\}$, the route selection at pre-route (a, b).

Value Nodes:

u, the utility node.

The current time $\mathbf{t}_{curr}$ and the current information state $\mathbf{s}_{curr}$ are control variables, so that the influence diagram can be used for replanning at different points in time prior to reaching location b. Note, that the notification time $\mathbf{t}_{no}(b)$ has an explicit influence on the default route $\mathbf{r}_b$ after notification.

Any available evidence for random variables such as $\mathbf{t}_{curr}$ and $\mathbf{s}_{curr}$ can be propagated through the network and the updated joint distribution will be computed. An efficient algorithm for this propagation usually known as *junction tree algorithm* has been introduced by Lauritzen and Spiegelhalter [LS88].

4.4.1.1 Utility u

The utility is given in Table 4.1 as conditional probability for an arrival in time. For $\mathbf{r}_b$=taxi, the utility is independent of the information state. $u(\mathbf{r}_b{=}\mathsf{taxi} \mid \mathbf{s}_u) = 50$ is the utility for taking the taxi. The utility given here is the probability of timely arrival at some destination (in percent). The outcome can be interpreted as follows:

- The traveller will reach his destination by taxi in time with a probability of 50%.
- Using a train, the traveller will reach his destination in time with a probability of 100%, if the train is not announced to be late ($\mathbf{s}_u = none$), and with 0%, if the train is delayed ($\mathbf{s}_u = delayed$).

Route $\mathbf{r}_b$	*train*		*taxi*	
State $\mathbf{s}_u$	*none*	*delayed*	*none*	*delayed*
Utility u	100	0	50	50

Table 4.1: *Utility Function u*

4.4.1.2 Conditional Probabilities for Information States

The conditional probability distributions (CPTs) for information states are based on transition probabilities $p_{ss'}(t, t')$ and shown in Table 4.2 and 4.3. We consider three time points for information states here, current time $\mathbf{t}_{curr}$, selected notification time $t_{no}(b)$ and arrival time at location b, i.e. $t_b = 0$. Therefore, $\mathbf{t}_{curr}$ and $t_{no}(b)$ both influence $\mathbf{s}_b$. Information state $\mathbf{s}_u$, which denotes the information state at arrival time $t_b = 0$, is used for utility assignment.

The transition probabilities shown are artificially created with respect to the following assumptions:

- The information state changes from *none* to *delayed* but never changes back.
- Transition probabilities are conditioned on the temporal distance until scheduled departure time. Here, the probability of the information state changing from *none* to *delayed* is *20%* between time points *-20* and *-10* vs. *10%* between time points *-10* and *0*.

Since $t_{curr} > t_{no}(b)$ is not possible, the respective conditional probabilities are irrelevant. The respective values are omitted and annotated by "-" in the table below. Identical columns are shown in one column using the $*$-symbol as a placeholder for any value.

	$\mathbf{s}_{curr}$	*none*									*delayed*
	$t_{no}(b)$	*-20*			*-10*			*0*			*
	t_{curr}	*-20*	*-10*	*0*	*-20*	*-10*	*0*	*-20*	*-10*	*0*	*
$\mathbf{s}_x$	*none*	1	-	-	.8	1	-	.72	.9	1	0
	delayed	0	-	-	.2	0	-	.28	.1	0	1

Table 4.2: Conditional Information State distribution of $\mathbf{s}_x$

4.4.1.3 Conditional Probabilities for Route Selection

Late notification means to inform the traveller not in time. This is directly by the influence of notification time $t_{no}(b)$ on the traveller's route selection $\mathbf{r}_b$ at that time. Table 4.4 shows the respective conditional distribution.

	$\mathbf{s}_b$	*none*			*delayed*		
	$t_{no}(b)$	*-20*	*-10*	*0*	*-20*	*-10*	*0*
$\mathbf{s}_u$	*none*	.72	.9	1	0	0	0
	delayed	.28	.1	0	1	1	1

Table 4.3: Conditional Information State distribution of $\mathbf{s}_u$

Early notification with $t_{no}(b)$ =*-20* always results in timely notification, i.e. the traveller's route selection is equal to the information system's route decision in these cases. A late notification time point of $t_{no}(b)$ =*0* prevents the notification to become effective, i.e. the traveller's route selection is equal to his current route r_{curr} in such cases. An intermediate notification time point ($t_{no}(b)$ =*-10*) will be timely with a probability of 90%.

	$t_{no}(b)$	*-20*				*-10*				*0*			
	$\mathbf{r}_{curr}$	*train*		*taxi*		*train*		*taxi*		*train*		*taxi*	
	δ_b	*tr.*	*taxi*	*tr.*	*taxi*	*tr.*	*taxi*	*tr.*	*taxi*	*tr.*	*taxi*	*tr.*	*taxi*
$\mathbf{r_b}_x$	*train*	1	0	1	0	1	.1	.9	0	1	1	0	0
	taxi	0	1	0	1	0	.9	.1	1	0	0	1	1

Table 4.4: Traveller's Route Selection $\mathbf{r}_x$

4.4.2 What-if Scenarios

After modelling the influence diagram, variations of the train scenario can be computed. We will look at two situations occurring at different time points.

4.4.2.1 Scenario I - Early Notification: t_{curr}=*-20*

Clock time t_{curr} is *-20* and current information state $\mathbf{s}_{curr}$ is *none*. The resulting expected utilities for different selected notification time points $t_{no}(b)$ are shown in Table 4.5.

$t_{no}(b)$	$E(u \mid t_{no}(b))$
-20	72.00
-10	79.90
0	61.00

Table 4.5: Expected Utilities for Scenario I

The best notification option is the one at time $t_{no}(b)$=*-10*. Therefore, notification will be deferred. The optimal route decision $\mathbf{r}_b$ cannot be determined, until notification time is reached and thus $\mathbf{s}_b$ is known by evidence.

4.4.2.2 Scenario II - Mid-Time Notification: t_{curr}=*-10*

Now, current time t_{curr} is *-10* and current information state $\mathbf{s}_{curr}$ is still *none*. Again, *-10* is the time of the optimal notification option (Table 4.6 (left)). For immediate notification, the resulting utilities for different route decisions δ_b can be seen in Table 4.6 (right).

$t_{no}(b)$	$E(u \mid t_{no}(b))$
-10	88.00
0	70.00

δ_b	$E(u \mid t_{no}(b) = \text{-}10, \delta_b)$
train	88.00
taxi	52.00

Table 4.6: Expected Utilities for Scenario II

Obviously, the notification option at current time should be realized by notifying the traveller with route *train*. Note, the second best notification policy is to wait for time point $t_{curr} = 0$ (expected utility 70) rather than sending the notification content *taxi* at current time (expected utility 52).

4.4.2.3 Scenario III - Late Notification: t_{curr}=*0*

Now, current time t_{curr} is *0* and current information state $\mathbf{s}_{curr}$ is still *none*. Now, *0* is the only time left for notification with expected utility $E(u \mid t_{no}(b) = 0) = 75$. However, the expected utility is the same for any notification content, because the notification cannot be effective, cf. Table 4.7.

δ_b	$E(u \mid t_{no}(b) = 0, \delta_b)$
train	75.00
taxi	75.00

Table 4.7: Expected Utilities for Scenario III

Obviously, no notification should be sent at all.

4.5 Road Scenario Revisited

The specific features of the road scenario (cf. Scenario 1.4.1) are as follows:

1. **Initial Route Determined:** The travellers initial route is determined, i.e. notification is only necessary if the IS' route decision differs from the initial default route.

2. **Two Decision Points:** Two consecutive decision points are considered here. However, the first decision may result in a loss of the second decision.

Fig. 4.11 shows three different routes from location a to location g (destination). The routes are marked as A, B and C respectively. Locations are labelled with lower case letters, routes are labelled with upper case letters and route segments with numbers.

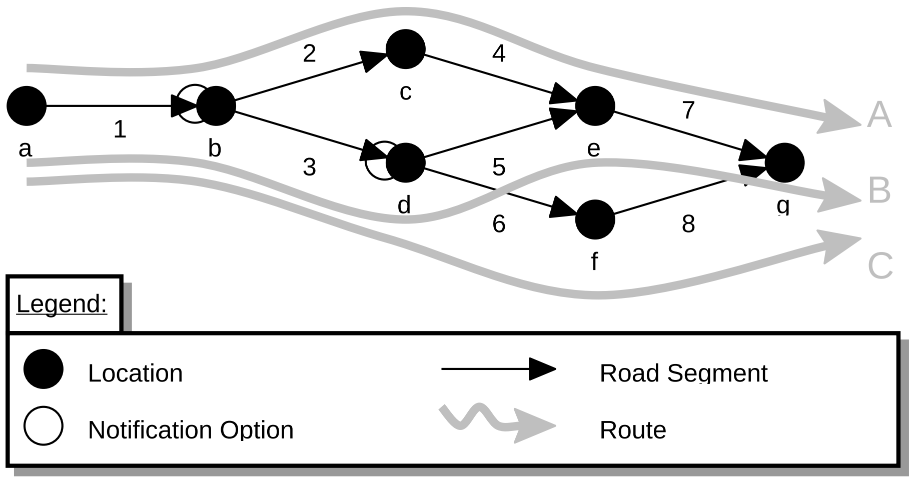


Figure 4.11: Routes for Road Scenario

The decision for route A has to be taken at location b while the choice between routes B and C may wait until reaching location d. Arrows represent the direction of travelling. Travelling along edges consumes time. The problem to be solved here is that of intelligent online guidance selecting the current best route.

Lets assume, that route B is the initial default route of the traveller. In this case, the question of whether to notify at location b or to wait until later, can be described as trade-off between information quality gain and loss of options, where

- *information quality gain* is obtained by information state changes during the traversal of route segment 3 while

- *loss of options* is experienced by loosing the opportunity of notifying about route A after location b has been passed by the traveller.

In Fig. 4.11, the information state consists of eight sub-states, one per route segment i, i.e. $s = (s_1, \ldots, s_8)$. The state space $S = S_1 \times \cdots \times S_8$ is the cross product of the local state spaces $S_i = \{c, f\}$ ($c \equiv$ "route segment is congested", $f \equiv$ "free flow on route segment"). For ease of representation, we look at route segments (c, e), (d, e) and (d, f) only, all other route segments are assumed to be always free.

4.5.1 The Influence Diagram Model

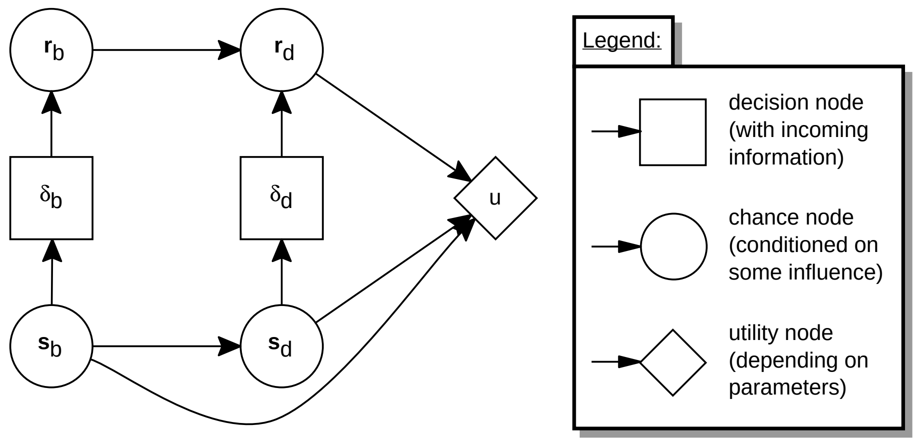


Figure 4.12: Influence Diagram for the Road Scenario

With pre-routes (a, b) and (a, b, d), two decision points are to be considered for notification. Node labels for an influence diagram have indices b or d indicating the respective pre-route:

Chance Nodes:

$\mathbf{r}_b, \mathbf{r}_d \in \{A, B, C\}$, the actual route decision of the traveller for notification at pre-routes (a, b) and (a, b, d), respectively;

$\mathbf{s}_b, \mathbf{s}_d \in S = \{c, f\}^8$, the information state at pre-route (a, b) and pre-route (a, b, d), respectively per route segment $i = 1, 2, \ldots, 8$;

Decision Nodes:

$\delta_b, \delta_d \in \{A, B, C\}$, the route selection at pre-route (a, b) and pre-route (a, b, d), respectively;

Value Nodes:

u, the utility node.

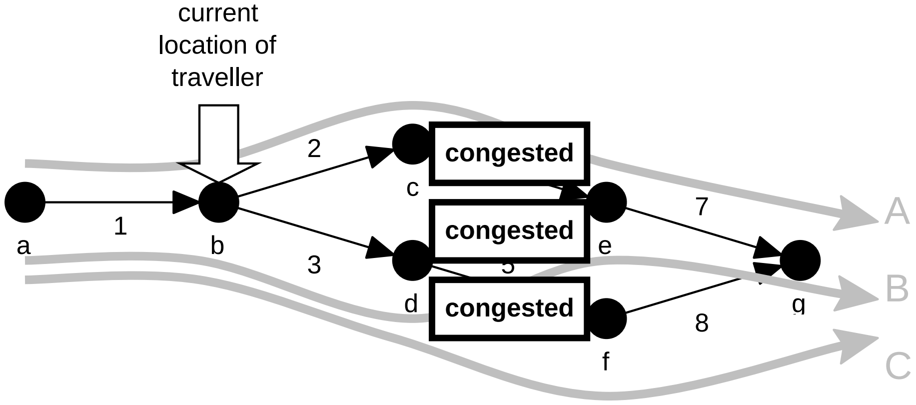


Figure 4.13: Current Situation on Route Segments

The simplified influence diagram is shown in figure 4.12. Explicit modelling of intermediate arrival times is replaced here by edges leading from any intermediate information state node directly to the utility node.

4.5.2 Evaluation

For the road scenario, the current situation is given by congestions on all three routes, cf. Fig. 4.13. A white arrow points to the current location of the traveller. For ease of representation, we write the relevant sub-states only, i.e. the current information state is $\mathbf{s}_{curr} = \mathbf{s}_b = (\ldots, c, c, c, \ldots)$.

In a second step, we have to consider the different possible updated states at the lookahead time. Instead of evaluating the influence diagram directly, we show the corresponding decision tree in Fig. 4.14. The evaluation of decision trees is described in Section 2.7.

At current time, the information state is simply the one given in Fig. 4.13. For future time point t_d, four different situations are of interest. These are the respective combinations of free and/or congested route segments 5 and 6. The information state for route segment 4 may have *any* value (denoted by a $*$).

Probability values are computed here based upon the assumption that a congestion will vanish independently per route segment with probability 0.8 within one time unit and no further congestion will occur. Thus, the conditional probabilities for the information state at decision point (a, b, d) with time $t_d = 1$ are:

- both congestions vanished: $p = 0.64 = 0.8^2$,
- one specific congestion vanished: $p = 0.16 = 0.8 \cdot 0.2$,
- no congestion vanished: $p = 0.04 = 0.2 \cdot 0.2$.

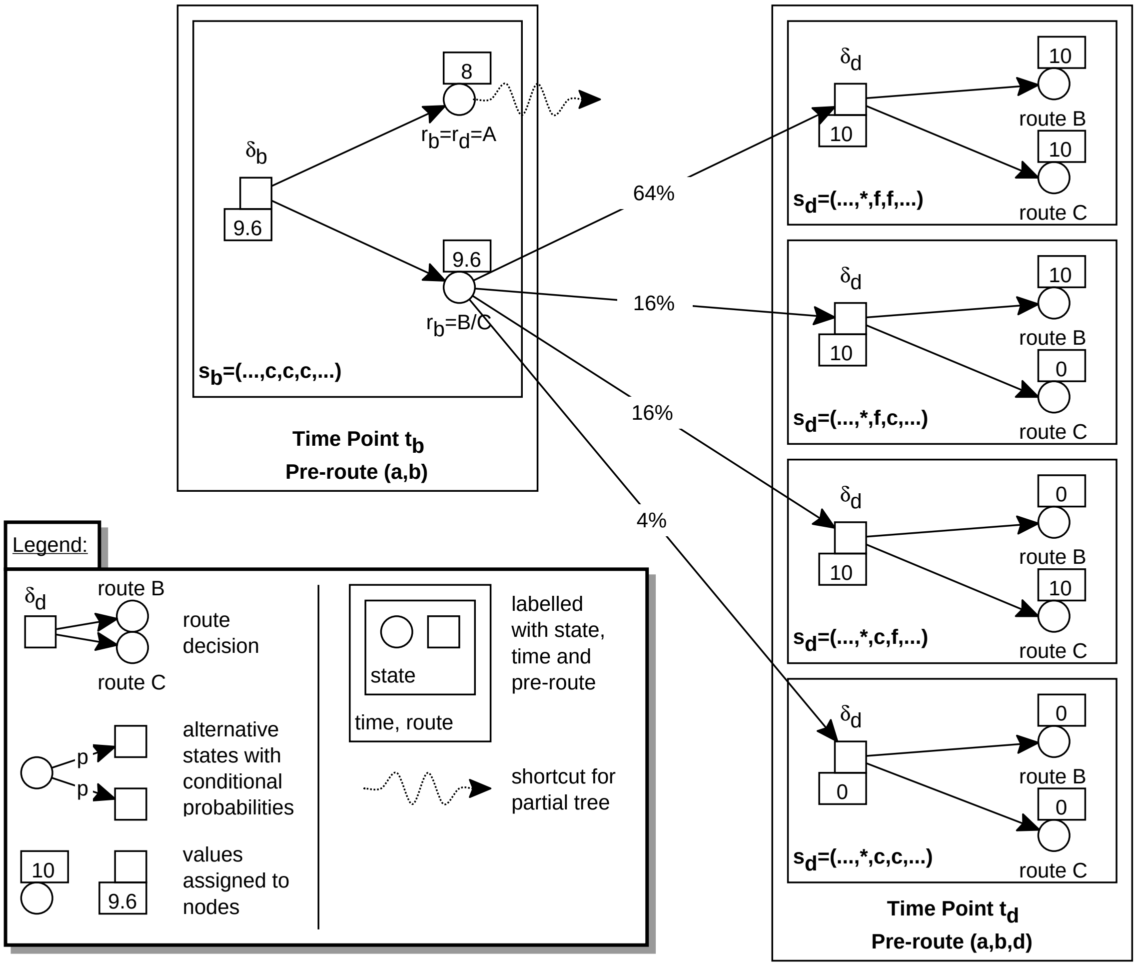


Figure 4.14: Evaluation of Influence Diagram

We assume current time $t_{curr} = 0$. The traveller's utility is equal to 10 for arrival at destination g prior to $t_{deadline} = 3.5$ and 0 for arrival after $t_{deadline}$.

$$u(t) = \begin{cases} 10 & t \leq t_{deadline} \\ 0 & t > t_{deadline} \end{cases} \tag{4.1}$$

The travel time is assumed to be one time unit for free route segments and two time units for congested route segments. Thus, the utility for any leave node in the decision tree is determined by whether or not a congested segment has been passed on the respective route. The decision tree has been truncated for route decision $r_b = A$. The expected utility for this decision is 8 since the congestion on route segment (c, e) vanishes with probability 0.8 until reached.

The expected utility of waiting at current time t_b in state $s_b = (\ldots, c, c, c, \ldots)$ with default route B or C is $E(u(t)) = 9.6$, cf. Fig. 4.14. This value is higher than for notification about route A at current time $(E(u(t)) = 8)$. The result is intuitively clear, since waiting provides us with better information for choosing between route B and route C, while taking route A gives exactly the reward corresponding to the probability of one congestion to disappear.

Chapter 5

Realizing *i-Alert* within the ILOG framework

This chapter discusses the realization of the *i-Alert* service and the requirements for its implementation within an ILOG framework developed, cf. Deiters and Lienemann [DL01]. The following issues will be covered:

- What are the properties of the *i-Alert* service?
- What are the required extensions of the ILOG -framework?

We start with a brief presentation of the *ILOG reference architecture* in Section 5.1. Then, the core functionality of the *i-Alert* service will be introduced in Section 5.2.

Issues of embedding the *i-Alert* service into the ILOG reference architecture are discussed in Section 5.3. Proposed adaptions and extensions of the ILOG reference architecture are presented in Section 5.4.

5.1 The ILOG Reference Architecture

In order to implement ILOG services as described in Section 1.2, a reference architecture for the implementation of information logistics services has been developed (cf. Deiters and Lienemann [DL01]).

An excerpt of this reference architecture is sketched in Fig. 5.1. On the top level, two system parts can be distinguished:

- The *execution system* is the core engine for the realization of any ILOG service. It provides support for scheduling, location- and event-based activation and for the delivery of messages and notifications.

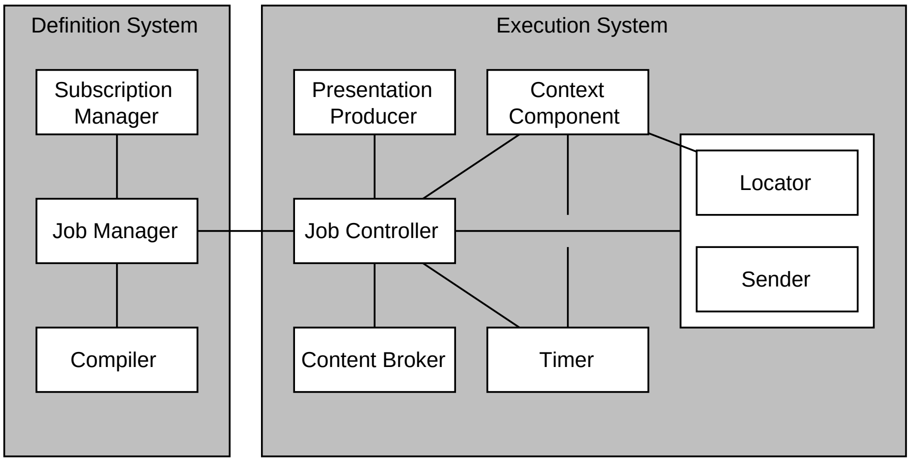


Figure 5.1: Reference Architecture for Information Logistics Applications (excerpt)

- The *definition system* is the service-specific mediator between the information need of the user and the execution plan for the core engine. Mediation is performed by the management of user subscriptions and by the service-specific compilation of subscriptions into execution plans.

The central component is the *Job Controller*. Jobs can be registered at the job controller and will be executed according to the following cases:

- *Immediate Execution:* Jobs intended for immediate execution will be executed immediately.
- *Scheduled Execution:* A job intended for scheduled execution will be scheduled by registration of the execution time point with the *Timer*. The timer sends a message to the job controller whenever a registered time point is reached.
- *Context-based Execution:* Jobs intended for context-based execution will be executed when a certain context occurs. The context will be registered by the *Context Component* which sends a message back to the job controller whenever the registered context occurs.

A job execution usually includes the transfer of a piece (or bunch) of information to the user. The job controller is connected to the *Content Broker*, the Presentation Producer and to the *Locator/Sender* components. The content broker is used for the content creation, the presentation producer is used for message creation and the locator/sender is used for user localization and message transmission.

The maintenance of jobs is performed by the *Job Manager*. Application-specific subscriptions stemming from the user are stored by the *Subscription Manager* and forwarded to the Job Manager. The Job Manager uses the *Compiler* for translating the subscriptions into jobs and registers them at the *Job Controller* for execution. After the last execution of a subscription, the job manager informs the subscription manager about subscription completion.

The interaction of the different components within the reference architecture is exemplified for a content service by a collaboration diagram in Fig. 5.2 and described below. The content service simply supplies a certain content at a predefined deadline. The context component is not used with this service.

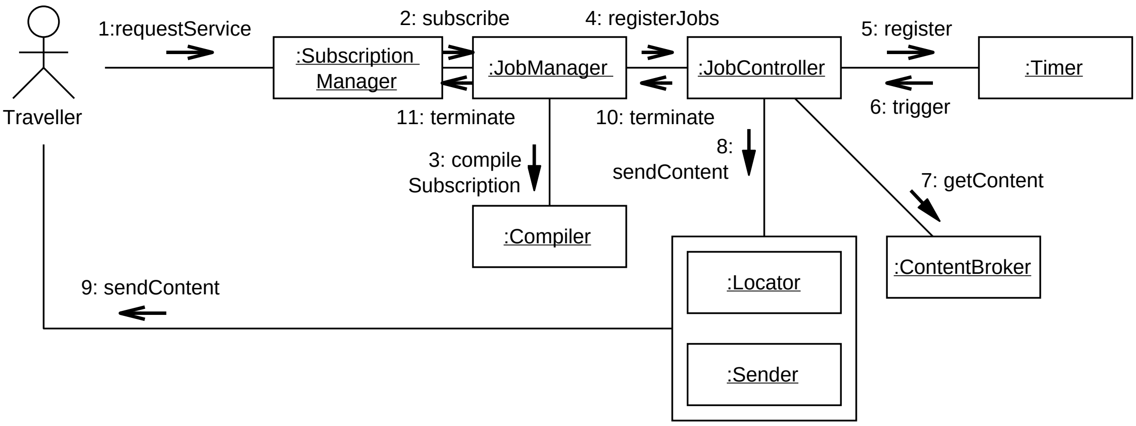


Figure 5.2: Collaboration Diagram for a Content Delivery

1. The Traveller *requests* an ILOG service.
2. The Subscription Manager stores the request and *subscribes* at the Job Manager.
3. The Job Manager *compiles* the subscription by the Compiler.
4. The Job Manager *registers* the job at the Job Controller.
5. The Job Controller *registers* the deadline at the Timer.
6. After reaching the deadline, the Timer *triggers* the Job Controller.
7. The Job Controller *gets* the content from the Content Broker
8. The Job Controller *notifies* the Traveller via Locator/ Sender and ...
9. ... the *Locator/ Sender* sends the notification to the Traveller.
10. The Job Controller *terminates*.
11. The Job Manager *terminates*.

Note, the *i-Alert* service cannot be realized directly within this reference architecture. This is due to the fact, that the *i-Alert* service requires replanning of jobs at certain time points and/or after certain event occurrences. This requires both a mechanism for triggering the replanning and the replanning itself. Both together is currently not supported by the ILOG framework:

- The job controller has a mechanism for triggering jobs, it cannot create and destroy jobs and thus cannot replan.
- The job manager can implement replanning by creation and destruction of jobs, but has no mechanism for triggering.

Thus, the job definition language for the execution system has to be extended or the job manager needs to be triggered by the execution system.

5.2 The *i-Alert* Service

> The *i-Alert* service provides the traveller with on-trip notifications about the best route to follow. The service is based on notification planning with notification options. The core functionality of the *i-Alert* service consists of triggers for replanning and notification of the traveller.

Given a notification problem with start-destination route set, computation is based on the current pre-route, current time and current information state. Because computation does not reduce to notification planning alone, temporal, content- and event-based triggers for revision and deferred notifications need to be defined as well.

Lets have a look at the activity diagram of *i-Alert* in Fig. 5.3. This activity diagram is invoked after the traveller has requested and specified his information demand, i.e. the start and destination of his route and the departure time. After computation of the start-destination route set, the best current route is selected as default route. After that, triggers for events and non-events are set which may influence the route selection. If the selected route is a new route, then the traveller is either notified immediately or otherwise a notification is scheduled for a later time. An event either signals a (scheduled) notification, a replanning or the finish.

What are the functional requirements for the realization of an *i-Alert* service within the ILOG architecture (cf. Section 5.1)? *Notify Traveller* and *Waiting for Events* can be performed by the execution system. However, the following functionalities have to be added:

Route Selection: Select a route or route segment at the current location, with current time and current information state that maximizes the expected utility.

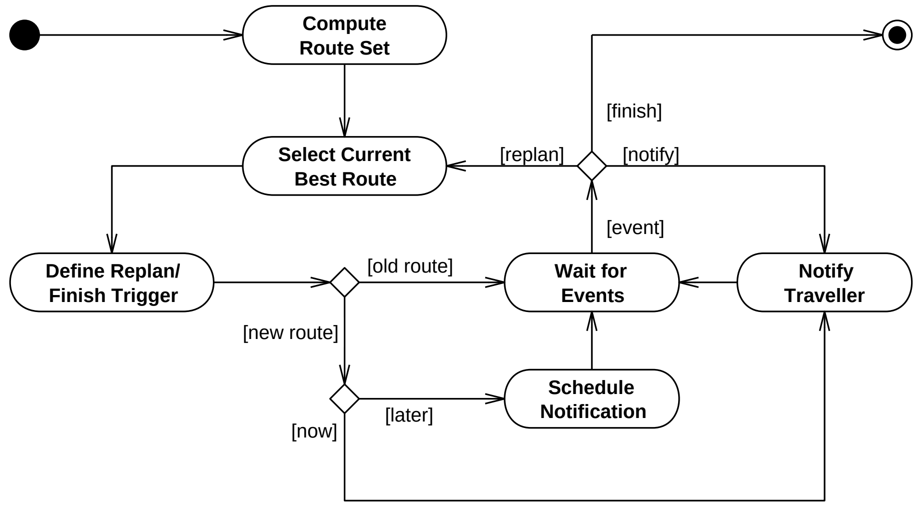


Figure 5.3: Activity Diagram for the *i-Alert* service

Either *myopic route selection* or *predictive route selection* might be appropriate, cf. Section 2.10.

Define Trigger: The next activation has to be specified with respect to events and conditions leading to replanning, or finish. This is usually specified by event classes for relevant events such as congestion warnings or train delay announcements and additionally by a time event (or deadline) set prior to the next decision point of the traveller.

Fig. 5.4 shows the class *iAlert* which provides the interface for both notification planning and the definition of triggers. The class *iAlert* is associated with supporting classes *PathPlanning* and *InfoStateMonitoring*. *PathPlanning* is responsible for default route planning and replanning, while the current information state is kept up-to-date by *InfoStateMonitoring*.

The following functions are provided by the *iAlert* class:

- *requestIntelliAlert(TimePoint,Location,Location,Goal):Handle*. This function takes a start time (of type *TimePoint*), a start location and a destination (both of type *Location*) and a utility function (of type *Goal*) and returns a handle (of type *Handle*) for further reference to this notification problem. The given parameters are internally stored with respect to the handle.

- *getInitialRoute(Handle):Route*. This function takes a handle (of type *Handle*) and returns a default route (of type *Route*). Internally, path planning is used for the computation of the route set R and the best route $r \in R$. The information state is monitored for all routes $r \in R$ by a call of method *startMaintenance(Routes)*.

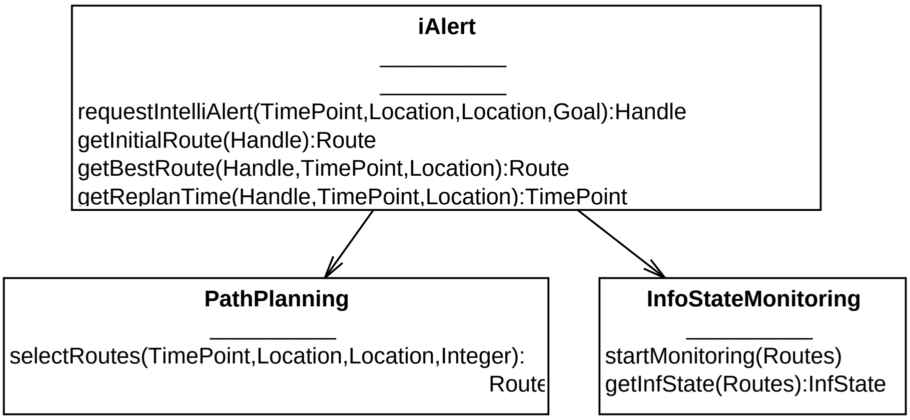


Figure 5.4: Class Diagram showing the *i-Alert* functions

- *getBestRoute(Handle,TimePoint,Location):Route.* This function takes a handle (of type *Handle*), the current time (of type *TimePoint*), the current location (of type *Location*) of the traveller and returns the best route (of type *Route*) for notification. The current information state is got by calling method *getInfState(Routes)* from class *StateMaintenance.*

- *getReplanTime(Handle,TimePoint,Location):TimePoint.* This function takes a Handle (of type *Handle*), the time (of type *TimePoint*) and the location (of type *Location*) of a traveller and returns the time of the next replanning (of type *TimePoint*). Internally, the earliest time for the traveller to reach the next decision point is computed.

The *path planning* is concerned with the computation of the k-fastest paths (or routes) for a given (distributed) transportation network (cf. Section 3.1). The travel times used for path planning are estimates of the unknown travel times and depend on departure time only. The path planning functionality is implemented by the class *PathPlanning.*

The *notification planning* computes the best route for notification with further notification options. The route decision function is implemented by class *iAlert.* The influence diagram decision models developed in Chapter 4 are used for computation.

The *InfoStateMonitoring* class maintains the current information state for the routes in the start-arrival route set $R(l_{start}, l_{dest})$. "Relevant" state variables are to be selected.

5.3 Replanning within the ILOG Reference Architecture

The *i-Alert* service involves that the plan for issuing of notifications is revised regularly. Within the ILOG reference architecture, this involves the creation or cancellation of jobs due to replanning. Conditions for revisions can be detected by the job controller, but the actual information state is kept by the application. Therefore, replanning can be implemented either on the job and or on the subscription level:

- On the *job level*, alternative plans for execution can be specified as part of a job. If a certain condition holds, an alternative plan is actually selected after execution has been started. That is, the job execution language provides a *case-statement* for conditional execution. Job level adaption has been proposed in the original framework for information logistics services from the ISST and is further studied in a master thesis on execution time prediction by Goldbeck[Gol02].
- On the *subscription level*, replanning can be implemented by recompilation of the subscription. In this case, the job manager interprets job termination as signal for replanning and deletes old jobs and creates new jobs according to the result of the replanning.

The adaption of use case execution both on the job and on the subscription level and the respective implications for an ILOG system will be discussed in the sequel.

5.3.1 Job Level Adaption

The execution of a use case within the ILOG reference architecture cannot be encoded into a single job in most cases. This is due to the fact, that a job can be viewed as an Event-Condition-Action (ECA)-rule, a well-known concept from active databases, cf. Paton and Diaz [PD99]. These ECA-rules are managed by the job controller, i.e. the event/condition part of the job is detected by the timer (for time events) or by the context component (for other events). The action part of the ECArule is a sequence of standardized actions to be performed using the functionalities of the content broker, presentation producer and locator/sender (for notification delivery). The time point of performing the action part is the *execution time* of a job. A job may have multiple executions times since the event/condition part may fire more than once.

For job level adaption, different variants of execution need to be specified and the selection of a variant is done at execution time. In order to do this, the job definition language needs to support job level adaption. Note, that the selection of a variant can either be performed by the job controller itself or by an additional decision component.

5.3.2 Subscription Level Adaption

On the subscription level, execution plan adaption is done by cancellation, creation, and change of jobs by the job manager. However, a mechanism is necessary for triggering the execution plan adaption. With the ILOG reference architecture, it is straightforward to use a simple job for scheduling the replanning. Thus, termination of a timed job without an explicit action part is the signal for the job manager to perform replanning.

This is illustrated by a sequence chart in Fig. 5.5.

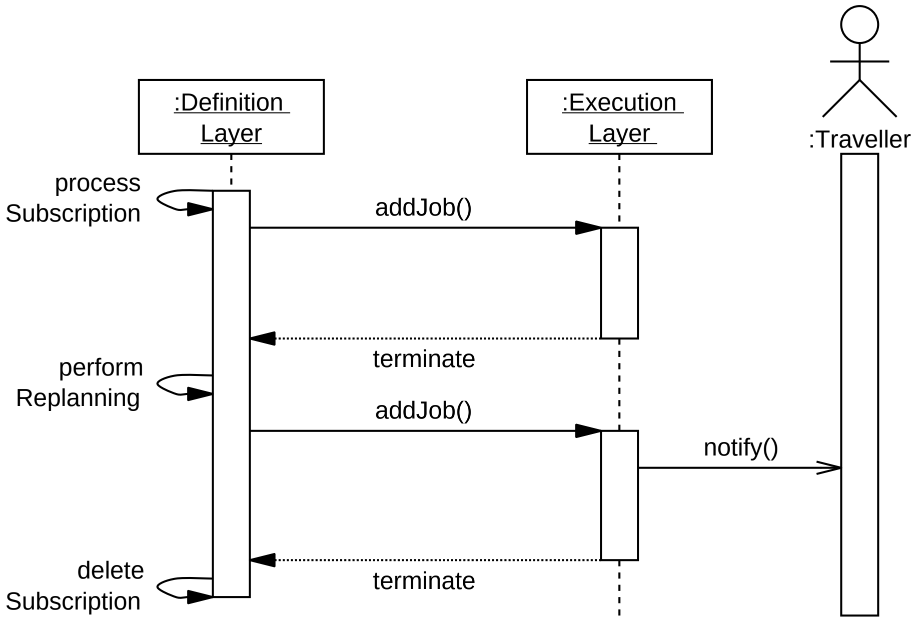


Figure 5.5: Execution Plan Adaption by Replanning on the Subscription Level

In this example the definition system registers a job at the job controller. This first job is used as a timer for replanning. When the definitions systems learns about the job termination, then it performs replanning and creates new jobs. In the example, the new job sends a notification to the traveller and terminates afterwards. Finally, the subscription gets deleted. Note, that the exact localization of the replanning has been left open at this level of abstraction.

5.3.3 Job Level Adaption vs. Subscription Level Adaption

Since we are not digging into implementation details, only two major differences between execution adaption on the job and on the subscription level can be found:

Localization of Functionality: The functionality of replanning is either located in the execution layer (for job level adaption) or in the definition layer (for subscription level adaption).

Extent of Intrusion: With subscription level adaption, the execution layer can stay unchanged, while for job level adaption both execution and definition layer need to be changed.

An implementation of the *i-Alert* service is not the focus of this thesis and no language extension of the job controller is developed. Therefore, the decision for either approach depends on the further development of the ILOG architecture itself and cannot be taken here. However, in the following section a feasible extension of the presented ILOG architecture is sketched for the case of subscription level adaption.

5.4 An Extended ILOG Architecture

Now, the components of the *i-Alert* service are inserted into the reference architecture for information logistics. The following additional components are needed:

- *Route Planner:* This component handles the negotiation with the user which leads to a *i-Alert* subscription.
- *IntelliAlert Compiler:* This component is used by the compiler for the translation of *i-Alert* subscriptions into jobs.

The resulting extended information logistics architecture is depicted in Fig. 5.6. Changes with respect to the reference architecture (Fig. 5.1) are highlighted by bold lines.

A typical interaction pattern starting from the initial route planning and containing one scheduled replanning and one notification is shown as a collaboration diagram in Fig. 5.7 and described below. Only some of the components are shown and communication steps specific to the extended architecture are highlighted.

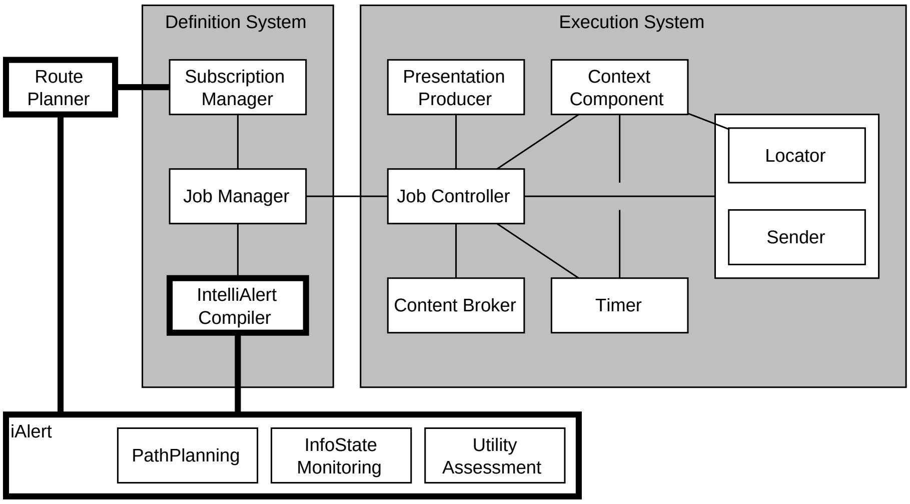


Figure 5.6: The Extended Architecture

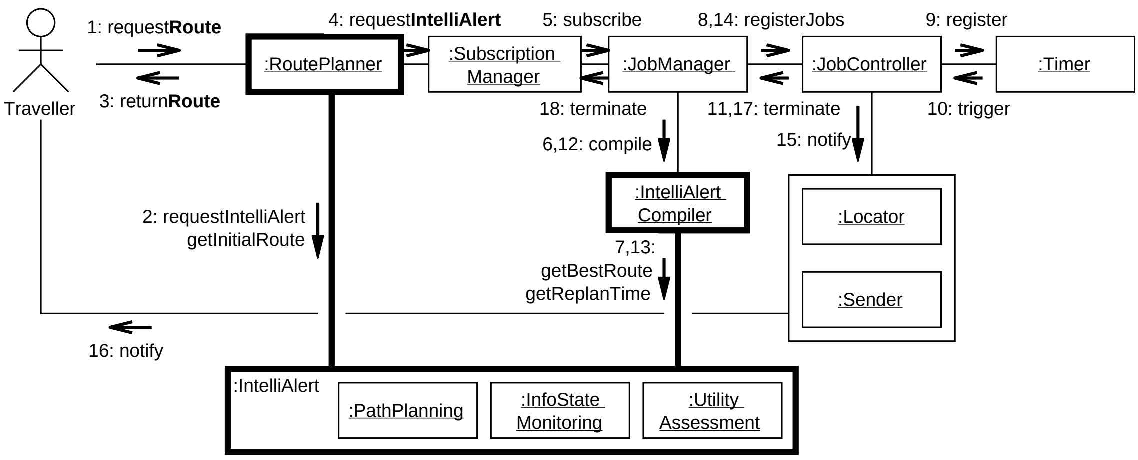


Figure 5.7: Collaboration Diagram for an *i-Alert* Notification Delivery

1. The traveller *requests* **a route** (and implicitly an ILOG service) at the **Route Planner**.
2. **The Route Planner *requests* an** *i-Alert* **function and *gets* the initial default route for the traveller.**
3. **The Route Planner *returns* the initial default route to the traveller.**
4. The **Route Planner** *requests* an *i-Alert* service for mobile route guidance.
5. The Subscription Manager stores the request and *subscribes* at the Job Manager.

6. The Job Manager *compiles* the subscription by the *i-Alert* Compiler.
7. **The Compiler *gets* the best route and next time for replanning.**
8. The Job Manager *registers* the jobs at the Job Controller.
9. The Job Controller *registers* the *time for replanning* at the Timer.
10. When time of replanning reached, the Timer *triggers* the Job Controller.
11. The Job Controller *terminates*.
12. The Job Manager ***recompiles*** the subscription by using the Compiler.
13. **The Compiler *gets* the best route and next time for replanning.**
14. The Job Manager *registers* the jobs at the Job Controller.
15. The Job Controller *notifies* Locator/ Sender **about the best route**.
16. The *Locator/ Sender* sends the notification to the traveller accordingly.
17. The Job Controller *terminates*.
18. The Job Manager *terminates*.

Chapter 6

Conclusion

This thesis provides a study of a synthesis between mobile route guidance and information logistics. Other than route planning with static travel times, mobile route guidance provides the possibility to adapt the current route of a traveller to a developing information state. The prediction of future information states is studied in order to meet the requirements of information logistics, i.e. finding the right time and content for notifying a traveller.

The idea of a trade-off between *information quality gain* and *loss of options* by route selection is a fundamental concept and a mathematical model is developed in this thesis. Options are lost by choosing a specific route and information quality gain may occur when time passes by. We propose *predictive route selection* which considers these effects and provide a quantitative model for *trading information quality gain vs. loss of options* for mobile route guidance. This is the major contribution of this thesis.

Predictive route selection is based on planning under uncertainty, a methodology from artificial intelligence/ operations research and route planning, a methodology from operations research.

The approach for *predictive route selection* is not considered elsewhere and exists in its own right. Therefore, we will only briefly discuss its properties and limitations in comparison with existing information systems for route planning:

- predictive route selection considers state- and time-dependent travel times. This leads to deviations of scheduled and real arrival times due to developing information states and uncertainty.
- predictive route selection is reduced to a limited enumeration of routes, i.e. a route is selected from a predefined route set rather than from all possible routes of a transportation network.
- predictive route selection is superior to myopic route selection if notification options can be used.

In order to simplify the implementation of notification planning with predictive route selection, we provide both a mapping of a problem into an influence diagram and a proposal of a design of a mobile route guidance system within an information logistics architecture.

Current notifications can be directly derived from the influence diagram representation. In order to reduce the problem complexity and to allow for efficient computation, we use the following heuristic for the influence diagram representation: Transition probabilities for information states at different time points are substituted by their estimates.

We hope, that *notification planning with predictive route selection* becomes a key technology for upcoming ILOG services for mobile route guidance. For this, the availability of real-time data from federated transportation information sources is a prerequisite. Currently, this prerequisite is missing.

Bibliography

[AFS99] Gustavo Arroyo-Figueroa and L. Enrique Sucar. A Temporal Bayesian Network for Diagnosis and Prediction. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–99)*, pages 13–20, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

[Ahi87] Niv Ahituv. A metamodel of information flow: a tool to support information systems theory. *Communications of the ACM*, 30(9):781–791, September 1987.

[All83] James F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26:832–843, 1983.

[Bat00] John Bather. *Decision Theory: An Introduction to Dynamic Programming and Sequential Decisions.* John Wiley and Sons, 2000.

[Ber90] C. Berzuini. Representing Time in Causal Probabilistic Networks. In M. Henrion, R. D. Shachter, L. N. Kanal, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 5*, volume 10 of *Machine Intelligence and Pattern Recognition*, pages 15–28. North-Holland, Amsterdam, 1990.

[BK00] S. Busse and R.-D. Kutsche. *Modellierung informationslogistischer Anwendungen - Projektbericht.* Number 00 - 09 in Forschungsberichte des Fachbereichs Informatik. November 2000. Draft.

[BKS97] S Busse, R. Kutsche, and Carsten Schöning. *Metainformation im Bereich Umwelt- und Geoinformationssysteme - eine Modellierungsstudie.* Number 97 - 25 in Forschungsberichte des Fachbereichs Informatik. November 1997.

[CKAK94] Sharma Chakravarthy, V. Krishnaprasad, Eman Anwar, and S.-K. Kim. Composite events for active databases: Semantics, contexts and detection. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *20th International Conference on Very Large Data Bases, September 12–15, 1994, Santiago, Chile proceedings*, pages 606–617, Los Altos, CA 94022, USA, 1994. Morgan Kaufmann Publishers.

[DKK92] Thomas Dean, Jak Kirman, and Keiji Kanazawa. Probabilistic Network Representations fo Continuous-Time Stochastic Processes for Applications in Planning and Control. In James Hendler, editor, *Artificial Intelligence Planning Systems: Proceedings of the First International Conference (AIPS 92)*, pages 273–274, College Park, Maryland, USA, June 1992. Morgan Kaufmann.

[DL01] W. Deiters and C. Lienemann, editors. *Report Informationslogistik - Informationen just-in-time*. Symposion Verlag, 2001.

[DLP03] W. Deiters, T. Löffeler, and S. Pfennigschmidt. *The information logistics approach toward user demand-driven information supply*, pages 37–48. In Spinellis [Spi03], 2003.

[FAM96] George Ferguson, James Allen, and Brad Miller. Towards a Mixed Initiative Planning Assistant. In B. Drabble, editor, *Proceedings of the 3rd International Conference on Artificial Intelligence Planning Systems (AIPS-96)*, pages 70–77. AAAI Press, 1996.

[Gol02] Gernot Goldbeck. Zeitdauer-Prognosen von Arbeitsabläufen in informationslogistischen Systemen. Master's thesis, Computergestützte InformationsSysteme (CIS), Technische Universität Berlin, 2002.

[Har87] D. Harel. Statecharts: A visual formalism for complex system. *Science of Computer Programming*, 8(3):231–274, 1987.

[Has01] Sandra Haseloff. *Ortsunabhängige Informationsversorgung in informationslogistischen Anwendungen*, pages 277–290. In Deiters and Lienemann [DL01], 2001.

[Hau97] Milos Hauskrecht. *Planning and Control in Stochastic Domains with Imperfect Information.* PhD thesis, MIT, 1997.

[HM81] R. A. Howard and J. E. Matheson. Influence diagrams. In R. A. Howard and J. E. Matheson, editors, *Readings on the Principles and Applications of Decision Analysis*, volume II, pages 721–762. Strategic Decisions Group, Menlo Park, CA, 1981.

[Hor99] Eric Horvitz. Principles of Mixed-Initiative User Interfaces. In *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, volume 1 of *Characters and Agents*, pages 159–166, 1999.

[How66] Ronald A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, SSC-2:22–26, 1966.

[HV02] A. Hinze and A. Voisard. Composite Events in Notification Services with Application to Logistics Support. Technical Report tr-B-02-10, Freie Universitaet Berlin, 2002.

[ISO95] ISO/IEC JTC 1. *Open Distributed Processing - Reference Model - Part 2: Foundations. International Standard ISO/IEC IS 10746-2. ITU-T Recommendation X.902.*, 1995.

[Käm95] Thomas Kämpke. *Bestimmung von Präferenzfunktionen mittels linearer und nichtlinearer Optimierungsverfahren.* Berichte aus der Mathematik. Verlag Shaker, 1995.

[Kja95] U. Kjaerulff. dHugin: A computational system for dynamic time-sliced Bayesian networks. *International Journal of Forecasting*, 11:89–111, 1995. Special Issue on Probability Forecasting.

[KR76] Ralph L. Keeney and Howard Raiffa. *Decisions with Multiple Objectives.* John Wiley & Sons, New York, 1976.

[KS98] T. Kämpke and M. Schaal. Distributed Generation of Fastest Paths. *Proceedings of the 10th IASTED International Conference on Parallel and Distributed Computing and Systems*, pages 172–177, 1998.

[KSWLM95] R. Kutsche, Carsten Schöning, Susanne Waßerroth, and Projektteam LUIS-Modellierung. *Informationsmodellierung im Rahmen eines Umweltinformationssystems.* Number 97 - 25 in Forschungsberichte des Fachbereichs Informatik. April 1995.

[Law76] E. Lawler. *Combinatorial optimization - networks and matroids.* Holt, Rinehart and Winston, New York, 1976.

[Law99] David B. Lawrence. *The Economic Value of Information.* Springer-Verlag, New York, 1999.

[Leo98] U. Leonhardt. *Supporting Location-Awareness in Open Distributed Systems.* PhD thesis, Department of Computing, Imperial College, London, May 1998.

[Lew00] Jana Lewerenz. *Dialog Management in Distributed Systems.* PhD thesis, BTU Cottbus, 2000.

[LS88] S. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society series B*, 50:157–224, 1988.

[NM44] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behaviour*. Princeton University Press, 1944.

[OS95] Gultekin Özsoyoğlu and Richard T. Snodgrass. Temporal and real-time databases: A survey. *Knowledge and Data Engineering*, 7(4):513–532, 1995.

[PD99] N. W. Paton and O. Diaz. Active Database Systems. *ACM Computing Surveys*, 31(1):63–103, 1999.

[Pea91] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1991. (Revised 2nd Edition).

[Rad98] Radermacher, Franz-Joseph and Schaal, Markus and Schnittger, Stephan. DELFI - Durchgängige ELektronische FahrplanInformation. Projektträger Bodengebundene Verkehrstechnologien im Auftrag des BMV, Referat A 24, und des BMBF, Referat 621, Köln, Germany, 1998.

[Rho00] Bradley J. Rhodes. *Just-In-Time Information Retrieval*. PhD thesis, MIT Media Lab, 2000.

[SGS00] Glenn Shafer, Peter R. Gillet, and Richard Scherl. The Logic of Events. *ANNALSMAI: Annals of Mathematics and Artificial Intelligence*, 28:315–389, 2000.

[Sha86] Ross D. Shachter. Evaluating Influence Diagrams. *Operations Research*, 33(6), 1986.

[Sha87] Ross D. Shachter. Probabilistic Inference and Influence Diagrams. *Operations Research*, 36(4):589–604, 1987.

[Sha97] Glen Shafer. *The Art of Causal Conjecture*. The MIT Press, Cambridge, Massachusetts, 1997.

[SK00] Markus Schaal and Thomas Kämpke. Cross-organizational computation of fastest paths. In *Proceedings of ODYSSEUS 2000 - First International Workshop on Freight Transportation and Logistics*, Chania, Crete, May 2000.

[SL01] Markus Schaal and Hans-Joachim Lenz. Best Time and Content for Delay Notification. In *Proceedings of the TIME01 International Symposium*, Cividale, Italy, 2001.

[SL03] Markus Schaal and Hans-Joachim Lenz. *Notification Planning with Developing Information States*, pages 63–70. Number 472 in CISM Courses and Lectures. Springer Wien New York, Udine, 2003.

[Spi03] D. Spinellis, editor. *Cross-Media Service Delivery.* Kluwer Academic Publishers, Boston [et al.], 2003.

[TN94] Ahmed Y. Tawfik and Eric Neufeld. Temporal Bayesian Networks. In *Proceedings of First International Workshop on Temporal Representation and Reasoning (TIME)*, Pensacola, Florida, 1994. FLAIRS.

[UML01] Object Modeling Group. *Unified Modelling Language Specification, version 1.4*, September 2001.

[WFL95] Michael P. Wellman, Matthew Ford, and Kenneth Larson. Path Planning under Time-Dependent Uncertainty. In Besnard, Philippe and Steve Hanks, editors, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI'95)*, pages 532–539, San Francisco, CA, USA, August 1995. Morgan Kaufmann Publishers.

Appendix A

Glossary

C

Current Notification A *current notification* is a notification which takes place at current time.

D

Decision Point A *decision point* is the location in time and space at which the user has to select one of several alternative routes. A decision point is modelled here as end-point of a pre-route leading from the start to the location of the decision.

Decision Tree A *decision tree* is a tree of situations where a situation together with its children either forms an information state transition or a route decision. The leave nodes of a decision tree have a certain value of reward.

Default Route The *default route* is the route which will be followed by the user unless otherwise notified. The *default route* is an extension of the determined route up to the destination. Only the extension can be changed at some later time.

Determined Route The *determined route* is the determined part of the travellers default route. The *determined route* has two parts describing past and future movements. Past movements end at the situations location, future movements start from the situations location. A *determined route* can only be extended, it is never changed.

Duration A *duration* (notation: d) has a value and a unit measure. The value of a *duration* is typed by the real numbers $\mathbb{R}$. The unsigned difference between any two time points is a duration. Within a context, the unit measure is omitted and durations are identified with their value, e.g. duration $d = 3$.

E

Edge An *edge* in a transportation network is a primitive directed connection between two (ordered) locations. Edges are primitive in the sense, that no other location will be passed while moving along an edge.

Edge Label An *edge label* is attached to an edge in a transportation network and models the travel time along that edge. The travel time may depend on the departure time from the start location of the edge (for time-dependent edge labels) or on the information state at the time of departure (for state-dependent edge labels) or on both for state- and time-dependent edge labels.

Event An *event* is an instantaneous occurrence of interest at a point in time.

Event Class An *event class* is an abstraction of a single event.

Event History An *event history* is a set of all events within a context happened up to a certain time. $H(t)$ denotes the event history up to time point t.

Execution Time The *execution time* is the time of performing the action part of a job. A job may have multiple executions times since the event/condition part may fire more than once.

I

i-Alert The *i-Alert* service is an active and individual information service providing on-trip notifications about routes to follow in order to reach the traveller's goals.

ILOG Service An *ILOG service* provides individual information to the user that fulfills the following criteria: right mode of transmission, right content, right time and place and appropriate presentation.

ILOG System An *ILOG system* is an information system, that provides ILOG services.

Information Quality Gain The *information quality gain* is the improved quality of a future information state for taking certain decisions.

Information State The *information state* is a representation of the information systems knowledge about the current situation on the transportation network at a specific time. The information state is used for travel time prediction.

Information State Change An *information state change* is an update of the information state. An *information state change* is caused by the receipt of a signal. The receipt of the signal together with the information state change is viewed as a single atomic event. Information state changes are local events.

Information State Transition Starting from a specific information state at a time point, an *information state transition* is the random choice of one out of several different information states at some later time point.

L

Local Event A *local event* occurs at the place of observation and is observed at the time of occurrence.

Location A *location* is a point in space or a connected subset of the space, where a traveller can be located. Moving locations are not considered in this thesis.

Loss of Options The *loss of options* is the loss of route alternatives by passing junctions without notification.

M

Movement Event A *movement event* is either the departure from or the arrival at a geographic location of a traveller or a vehicle. Movement events are remote events.

Myopic Route Selection *Myopic Route Selection* is the selection of a route out of a set of n routes with maximum expected utility and without consideration of later route choices.

N

Notification A *notification* is a message sent to the user in order to inform about some situation or suggestion which is relevant for her or him in a specific context. In this thesis, the context is usually the timely arrival at some destination and notification is to be received by the user while being on the route towards that destination.

Notification Effect The notification effect is the user's change of future action caused by a notification. For *i-Alert* , the notification effect is modelled by a replacement of the default route.

Notification Option A *notification option* is a future option for notification. A notification option is specified by a content for notification and by a time point for notification.

Notification Planning *Notification planning* is a method proposed in this thesis for the selection of the best time and content for traveller notification. Notification planning uses future notification options, i.e. the improvement of the expected utility to be achieved by future notifications is considered for the selection of current notifications.

Notification Value The *notification value* of a current notification is defined as the difference between the best notification policy's expected utility with that notification and the best notification policy's expected utility without that notification.

P

Pre-Route A *pre-route* is a prefix of another route.

Predictive Route Selection *Predictive Route Selection* is the selection of a pre-route with maximum expected utility under the assumption that predictive route selection is repeated for the remaining route choices whenever a pre-route ends.

Probabilistic Duration A *probabilistic duration* (notation: boldface $\mathbf{d}$) is given by a probability space $(D, \mathcal{D}, P)$ where $D \subseteq \mathbb{R}$ is a finite set of durations, σ-algebra $\mathcal{D}$ is the power set of D and the probability measure $P : \mathcal{D} \to [0, 1]$.

Probabilistic Time Point A *probabilistic time point* (notation: boldface $\mathbf{t}$) is given by a probability space $(T, \mathcal{T}, P)$ where $T \subseteq \mathbb{R}$ is a finite set of time points, a σ-algebra $\mathcal{T}$ is the power set of T and the probability measure $P : \mathcal{T} \to [0, 1]$.

R

Remote Event A *remote event* does not occur at the place of observation.

Route A *route* is a sequence of locations without repetitions. A route consisting of two locations only is equivalent to an edge.

Route Decision A *route decision* is the selection of a extension for the determined route out of several possible route extensions.

Route Selection Policy With pre-route tree $(R, <_{pre})$ given, a *route selection policy* selects one pre-route $\delta(r, t, s) \in Childs(r)$ for any combination of a pre-route $r \in R$, an information state $s \in S$ and a time point $t \in T$.

S

Situation An *situation* of the traveller is modelled by the time point, the location of the traveller, the information state and the determined route.

Situation Tree A *situation tree* is a decision tree without decision nodes and without values for the leave nodes.

Start-Destination Route Set A *start-destination route set* is the set of routes starting from a single start location and ending at a single destination.

Subscription A *subscription* is the registration of an individual information need with a specific information service. A *subscription* is a model of a user's interest in receiving notifications.

Symbolic Location A *symbolic location* is an address, a public transport station or any other named location, e.g. a platform within a station.

T

Time Event A *time event* is the event of a clock reaching a certain time point.

Time Interval The ordered set of the values of the time points in a closed *time interval* (notation $[t_1, t_2]$) is an interval of real numbers with t_1 being the value of the beginning time point and t_2 being the value of the end time point.

Time Point A *time point* (notation: t) has a value, a time origin and a unit measure. The value of a *time point* is typed by the real numbers $\mathbb{R}$. A *time point* is related to a real-world time point by specification of time origin and unit measure. The value of the time point is the measurement of the difference between the time origin and the real-world time point to be modelled. Within a context, time origin and unit measure are omitted and time points are identified by their value, e.g. time point $t = 23$.

Transition Probability A transition probability is a conditional probability for a certain state at a future time point conditioned on a certain state at some earlier time point.

Transportation Network A *transportation network* is a connected graph consisting of locations and directed edges between locations.

Traveller The role name *traveller* and the role name user are used interchangeable in this thesis, both referring to the human user of the *i-Alert* service who is also a traveller.

U

User The role name *user* and the role name traveller are used interchangeable in this thesis, both referring to the human user of the *i-Alert* service who is also a traveller.

Utility Function An *utility function* u is an order preserving mapping of the preferences of the user. Let x and y be options. Then $u(x) > u(y)$ if and only if the user prefers x over y.

Appendix B

Acronyms

AI Artificial Intelligence

CIS Computergestützte Informationssysteme, TU Berlin

DAG Directed Acyclic Graph

DFP Distributed Fastest Path

ECA Event-Condition-Action

GPS Global Positioning System

GSM Global System for Mobile communication

ILOG Information LOGistics

IS Information System

ISST Fraunhofer Institut für Software- und Systemtechnik

PDA Personal Digital Assistant

RM-ODP Reference Model of Open Distributed Processing

SMS Short Message Service

UML Unified Modelling Language

Appendix C

Comparison of Notification Policies

A notification policy determines the route to follow at any decision point. Specifically, we consider three different notification policies for comparison:

Pre-Trip Notification The traveller gets notified about the best route only at the beginning of his route.

Myopic Notification The traveller gets notified about the best remaining route at current time.

Predictive Notification The traveller gets notified with future notification options considered.

These notification policies have different utilities for the traveller. This will be discussed here:

Pre-Trip Notification In this case, the utility of the notification policy is equal to the expected utility of the best route at the start location. With start time t and information state s the utility $u_{pretrip}(s,t)$ for pre-trip notification is given by

$$u_{pretrip}(s,t) = \max_{r \in R(l_{start}, l_{dest})} \bar{u}(r,s,t) \tag{C.1}$$

Myopic Notification In this case, the utility can be improved in front of decision points by changing the route during the journey. Let $r = r_{pre}$ be the current pre-route (ending with $l = l_{curr}$, the current location), $t = t_{curr}$ be the current time and $s = s(t_{curr})$ be the current information state. Let $R(r_{pre}) = \{r \in R(l_{start}, l_{dest}) \mid r_{pre} = head(r)\}$ be the set of feasible[1] routes from the current location l_{curr} to the destination l_{dest}.

[1] along some route in $R(l_{start}, l_{dest})$

The best current route is given by

$$r^* = \underset{r \in R(r_{pre})}{\text{argmax}} \; \bar{u}(r, s, t) \tag{C.2}$$

Let $r'_{pre} = r_{pre} \cdot l'$ be the next pre-route after r_{pre} (in the pre-route tree) when following the best current route r^*. Then, the utility $u^{myopic}(r_{pre}, s, t)$ for myopic notification is given by

$$u_{myopic}(r_{pre}, s, t) = \begin{cases} u(t) & \text{for } r_{pre} \in R(l_{start}, l_{dest}) \\ E(u_{myopic}(r'_{pre}, t', \mathbf{s}') \mid s, t, t') & \text{else} \end{cases} \tag{C.3}$$

with $E(u_{myopic}(r'_{pre}, t', \mathbf{s}') \mid s, t, t') = \sum_{s' \in S} p_{ss'}(t, t') \cdot u_{myopic}(r'_{pre}, t', s')$ and $t' = t + d_{ll'}(s, t)$.

Predictive Notification In this case, the best route at each future decision point is computed. Let $r = r_{pre}$ be the current pre-route, $t = t_{curr}$ be the current time and $s = s(t_{curr})$ be the current information state.

The current location is denoted by $l = l_{curr}$. Let $Succ(l)$ denote the set of possible direct successor nodes of l in the route set. Then, the utility $u^{predict}(r, s, t)$ for predictive notification is given by

$$u^{predict}(r, s, t) = \begin{cases} u(t) & \text{for } r \in R(l_{start}, l_{dest}) \\ \underset{l' \in Succ(l)}{\max} \; \sum_{s' \in S} p_{ss'}(t, t') \cdot \\ \quad u^{predict}(r \cdot l', s', t') & \text{else} \end{cases} \tag{C.4}$$

with $t' = t + d_{ll'}(s, t)$

Theorem 1 *For any notification problem given by transportation network (L, E), state- and time-dependent travel time durations d_{ij}, start location l_{start}, destination l_{dest}, utility function u, start time t and information state s at start time, myopic notification is better than pre-trip notification, i.e.* $u^{myopic}((l_{start}), s, t) \geq u^{pretrip}(s, t)$ *with* $r_{pre} = (l_{start})$.

Proof:

We introduce a family of notification policies representing stages between myopic notification and pre-trip notification. With the k-th notification policy of the family, myopic notification will be applied to pre-routes up to length k, $k = 1, 2, \ldots, n$. The expected utility of the k-th notification policy when starting with pre-route r_{pre} at time t and in information state s is denoted by $u_k^{myopic}(r_{pre}, s, t)$ and given by:

$$u_k^{myopic}(r_{pre}, s, t) = \begin{cases} \underset{r \in R(r_{pre})}{\text{argmax}} \ \bar{u}(r, s, t) & \text{for } length(r_{pre}) \geq k \\ \\ \sum_{s' \in S} p_{ss'}(t, t') \cdot u_k^{myopic}(r'_{pre}, s', t') & \text{else} \end{cases} \tag{C.5}$$

with $r'_{pre} = r_{pre} \cdot l'$ is best next pre-route and
with $t' = t + d_{ll'}(s, t)$.
We show $u^{pretrip}(s, t) \leq u^{myopic}((l_{start}), s, t)$ by

1. $u_1^{myopic}((l_{start}), s, t) = u^{pretrip}(s, t)$ by definition.

2. $u_n^{myopic}((l_{start}), s, t) = u^{myopic}((l_{start}), s, t)$ by definition.

3. $u_k^{myopic}((l_{start}), s, t) \leq u_{k+1}^{myopic}((l_{start}), s, t)$ since every additional change of the default route is only done in order to improve the utility.

q.e.d.

Theorem 2 *For any notification problem given by the transportation network (L, E), travel time durations $d_{ij}(s, t)$, start location l_{start}, destination l_{dest}, utility function $u(t)$, start time t_{start} and information state $s(t_{start})$ at start time, predictive notification is better than myopic notification, i.e. $u^{predict}(r_{pre}, s(t_{start}), t_{start}) \geq u^{myopic}(r_{pre}, s(t_{start}), t_{start})$.*

Proof:
Let $r_{myopic} = (l_1, \ldots, l_n)$ be the route followed by a myopic notification policy. We prove theorem 2 by induction over the length k of the pre-route. We use r_k as a shortcut for $(l_k, \ldots, l_n)$ for $1 \leq k \leq n$, i.e. $r_{myopic} = r_n$.

- $k = n$:
$$u^{myopic}(r_{myopic}, s_n, t_n) = u^{predict}(r_{myopic}, s_n, t_n)$$

- $1 < k \leq n$: If for any $s_k \in S$
$$u^{myopic}(r_k, s_k, t_k) \leq u^{predict}(r_k, s_k, t_k) \tag{C.6}$$
then
$$u^{myopic}(r_{k-1}, s_{k-1}, t_{k-1}) = \sum_{s_k \in S} p_{s_{k-1}s_k}(t_{k-1}, t_k) \cdot u^{myopic}(r_k, s_k, t_k)$$

and with Eq. C.6

$$u^{myopic}(r_{k-1}, s_{k-1}, t_{k-1}) \leq \sum_{s_k \in S} p_{s_{k-1}s_k}(t_{k-1}, t_k) \cdot u^{predict}(r_k, s_k, t_k)$$

and since the utility of predictive notification is a maximum

$$u^{myopic}(r_{k-1}, s_{k-1}, t_{k-1}) \leq u^{predict}(r_{k-1}, s_{k-1}, t_{k-1})$$

q.e.d.

Appendix D

Modelling Techniques

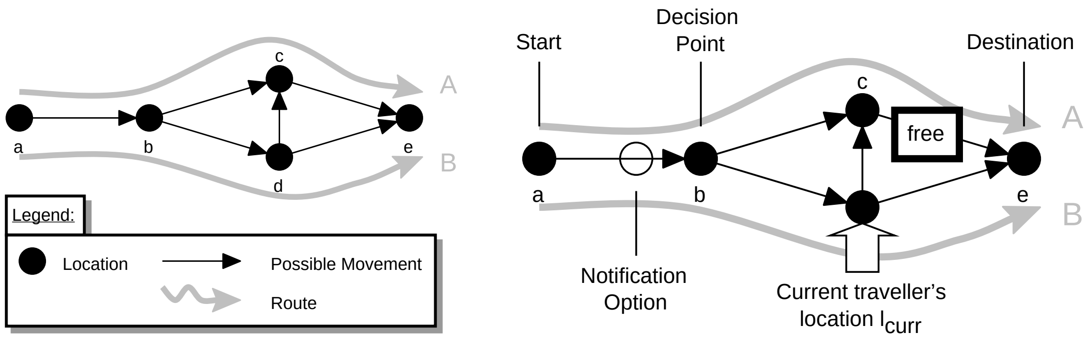
D.1 Transportation Networks

Transportation networks together with a set of routes are illustrated by using the following conventions:

- Locations are depicted by filled circles. An identifier for a location is optionally placed next to the location.
- Edges are depicted by directed arrows. The direction of the arrow indicates the order of the locations for this edge.
- Routes are depicted by thick curved lines in light-gray color following the sequence of locations and ended by an arrowhead. An identifier for a route is placed next to the arrowhead.

An example is given in Figure D.1(a) depicting location set $L = \{a, \ldots, e\}$, edge set $E = \{(a, b), (b, c), (b, d), (c, e), (d, c), (d, e)\}$ and route set $R = \{A, B\}$.
The following additional elements for improving the expressive power of the illustration are used (cf. Fig. D.1(b)):

- Text fragments connected with lines to certain locations (cf. Fig. 1.6 and Fig. 1.7) indicate where start location, destination location, or decision points are located.
- A white circle illustrates the position of a notification option prior to a certain decision point. Next to the white circle, a textual specification for the intended notification time can be placed (cf. Fig. 2.16, Fig. 4.9 and Fig. 4.11).
- A white box arrow indicates the current location of the traveller (cf. Fig. 4.2 and Fig. 4.13).



(a) Plain (b) With Optional Elements

Figure D.1: Transportation Network with Route Set

- The current situation on (information state for) an edge is depicted by a box containing the value of the respective information state which is drawn directly on the respective arrow (cf. Fig. 4.13).

D.2 Decision Diagrams

Decision diagrams for planning under uncertainty visualize uncertain quantities, decisions, temporal order between events and relations between uncertain quantities.

Two kinds of variables are modelled, namely decisions and random variables. Decisions variables have a name and a set (domain) of choices called action space. Random quantities have a name and a set (domain) of outcomes. Uncertain quantities may be conditioned on other variables. A value can be assigned to each choice or outcome bei applying a utility function.

We use two different kind of decision diagrams: *Influence diagrams* and *decision trees*.

D.2.1 Influence Diagrams

Influence diagrams are directed graphs with three types of nodes (cf. Howard [HM81], Pearl [Pea91] and Shachter [Sha87]). Chance nodes (shown as ovals) represent uncertain quantities, decision nodes (shown as rectangles) represent decision options and value nodes (shown as diamonds) represent rewards or costs both for decisions taken and outcomes of uncertain quantities. Directed links leading to chance nodes denote conditional dependency, directed links leading to value nodes denote functional dependency and directed links leading to decision nodes are informational, i.e. the respective quantity is known before the decision has to be made.

The following graphical notations are used:

- chance nodes are shown as circles
- decision nodes are shown as rectangles
- value nodes are shown as diamonds
- the name of a node is shown next to it

The influence diagram notation is demonstrated in Fig. 4.1 with different types of edges explicitly annotated. Edge labels are omitted in the sequel.

This is illustrated in Fig. D.2

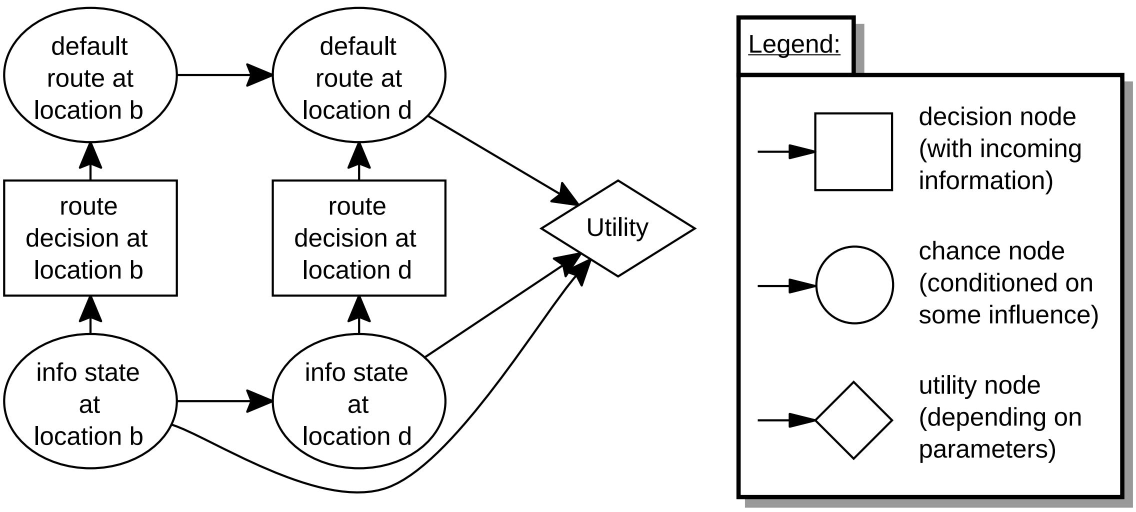


Figure D.2: Influence Diagram Example

D.2.2 Decision Trees

Starting from a root node, each successor node is either a decision or a chance node. In the case of a decision, the node is shown as rectangle, in the case of a chance, the ancestor is shown as a circle and each child occurs with a certain conditional probability which is shown as a label for the connecting arrow.

Decision trees can be evaluated. The value of a node is shown next to the node in a small rectangle. Evaluation goes backward from the leave nodes. A leave node is a value node. It assigns a value to a specific path in the decision tree. Optionally, the leave nodes are shown as diamonds.

The following graphical notations are used:

- chance nodes are shown as circles
- decision nodes are shown as rectangles
- value nodes are shown as diamonds
- one or several nodes are enclosed by a rectangle which indicates the time, location and information state for this or these nodes
- the name of a node is shown next to it
- the value of a node is shown next to it (both for value nodes and back-propagation)

This is illustrated in Fig. D.3

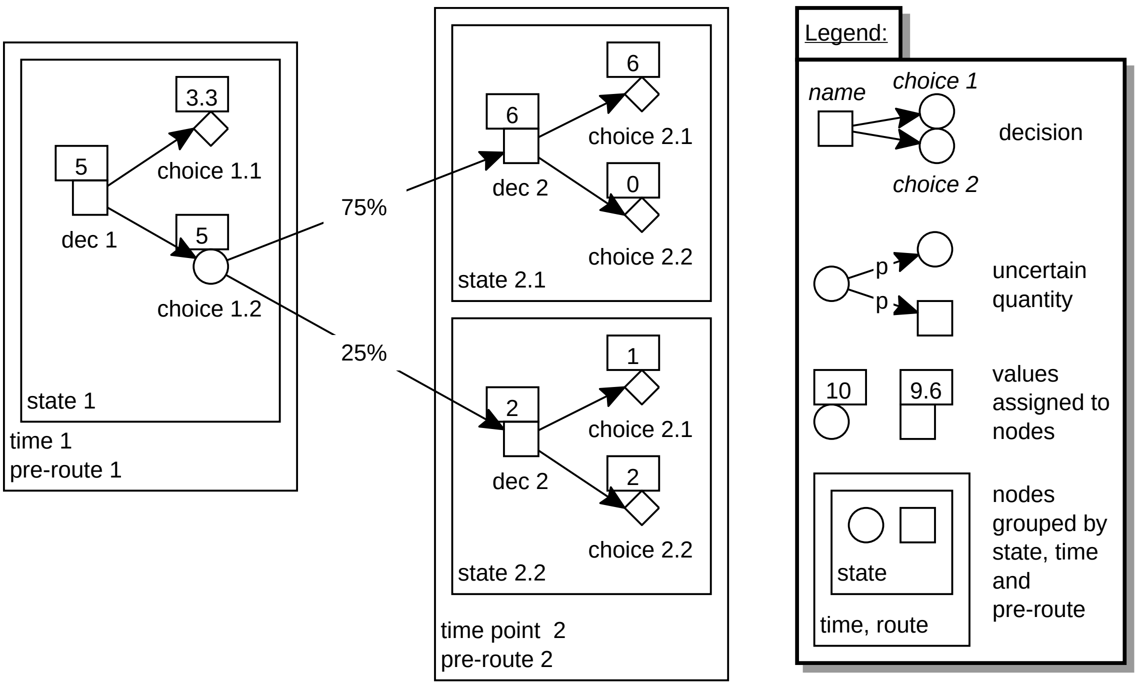


Figure D.3: Decision Tree Example