

---

# Individuelle Stadtführungen mit einem Dynamischen Tour Guide<sup>1</sup>

Marko MODSCHING, Ronny KRAMER und Klaus TEN HAGEN

## Zusammenfassung

Besucher einer fremden Stadt sind meist daran interessiert diese kennen zu lernen. Erfahrungsgemäß ist das Finden von Sehenswürdigkeiten per Karte und Schildern ein schwieriges Unterfangen, sodass viele interessante Attraktionen nicht entdeckt werden können. Abhilfe schafft hier der prototypisch realisierte Dynamische Tour Guide (DTG) - eine Softwareentwicklung für ein mobiles Gerät, der die Rolle eines Stadtführers übernimmt und auf die individuellen Bedürfnisse des einzelnen eingeht. Beim Erreichen des Reiseziels sucht der DTG automatisch nach vorhandenen Attraktionen (z.B. Sehenswürdigkeiten, Restaurants) in der unmittelbaren Umgebung. Diese werden auf aktuellen Informationen wie Öffnungszeiten überprüft. Anhand der persönlichen Interessen des Touristen werden die besten Attraktionen zu einer Tour kombiniert. Durch Navigationshinweise und Informationen in multimedialer Form wird der Tourist auf dieser Tour aktiv vom DTG unterstützt.

## 1 Konzept

Bisher war es kaum realisierbar, eine persönliche Besichtigungstour verschiedener Sehenswürdigkeiten in einer fremden Stadt kurzfristig durchzuführen. Ohne Vorbereitung, das heißt ohne sich über Standpunkte interessanter Sehenswürdigkeiten und deren Öffnungszeiten zu informieren, ist eine Suche mühselig und endet häufig vor geschlossenen Türen. Ein Großteil der Touristen schließt sich daher lieber einer Reisegruppe an und gelangt auf den touristischen Hauptwegen zu Sehenswürdigkeiten die möglicherweise nicht von unmittelbarem Interesse für ihn sind.

Viele Touristen würden sich jedoch gern spontan für eine individuelle Sightseeingtour entscheiden. Mit dem DTG ist solch eine spontane Tour möglich, da er unter Einbeziehung der persönlichen Interessen des Touristen und der Daten der Sehenswürdigkeiten eine Tour empfiehlt. Die Interessen des Touristen, sowie dessen Position werden auf einem mobilen Gerät erfasst. Durch Eingabe des gewünschten Zielpunktes und der verfügbaren Zeit wird sein persönlicher Kontext komplettiert. Dieser kann nun auf den lokalen Kontext der Umgebung abgebildet werden. Ziel ist es, anhand der gewonnenen Informationen eine Tour zu berechnen und diese zu optimieren. Die Tour enthält für die vorgegebene Dauer möglichst viele Attraktionen, die den Interessen des Touristen entsprechen.

Es besteht die Möglichkeit für den Touristen die vorgeschlagene Tour zu modifizieren, indem einige Attraktionen entfernt bzw. hinzugefügt werden können. Durch die Überwachung der Tour im Hintergrund kann der DTG auf eventuelle Vorkommnisse mit einer

---

<sup>1</sup> „Symposium und Fachmesse für Angewandte GeoInformatik – AGIT 2005“; 06.-08.Juli.2005; Salzburg; <http://www.agit.at>

Anpassung der Tour reagieren. So führen z.B. eine zu langsame Laufgeschwindigkeit des Touristen oder zusätzliche Pausen dazu, dass einige Sehenswürdigkeiten gestrichen werden müssen. So kann sichergestellt werden, dass der Touristen pünktlich sein Ziel erreicht.

## 2 Exemplarischer Ablauf

Folgender exemplarischer Ablauf gibt einen Überblick über die Funktionalität des DTG:

Ein Tourist, der sich einen Mobile Digital Assistant (MDA) mit installierten DTG z.B. bei einer Touristeninformation ausleiht muss zunächst seine Interessen spezifizieren. Für den Fall, dass jemand den DTG auf seinem persönlichen Gerät installiert, kann das entstehende Profil lokal gespeichert werden, um diesen Schritt bei einem späteren Start überspringen zu können. Der DTG wählt danach automatisch als Startpunkt die aktuelle Position aus. Diese kann über einen GPS Empfänger metergenau bestimmt werden. Die letzte Information, die der DTG für eine Tour noch benötigt, ist die Angabe über die verfügbare Zeit, die der Tourist bereit ist mit einer Tour zu verbringen. Optional kann der Tourist noch einige Sehenswürdigkeiten der Stadt als obligatorisch kennzeichnen, so dass diese in jedem Fall Bestandteil der Tour sind. Nun findet der DTG alle Sehenswürdigkeit in der Umgebung des Touristen und bewertet dieses gemäß den Interessen des Touristen und den obligatorischen Bausteinen. Das Resultat dieser Berechnung sind mehrere Touren, von denen der Tourist eine auswählen und starten kann.

Nach dem Start der Tour beginnt der DTG damit, den Touristen mit audio-visuellen Navigationsanweisungen zu den einzelnen Attraktionen zu führen. Nähert sich der Tourist einer Attraktion wird die Navigation unterbrochen und es wird damit begonnen, Informationen über die Sehenswürdigkeit zu präsentieren. Diese Informationen werden per Audio an den über Bluetooth angebandenen Kopfhörer des Touristen übertragen. Auf Wunsch kann der Tourist das Gerät noch aus der Tasche ziehen und sich Bilder oder Texte zu der Attraktion ansehen. Wünscht der Tourist keine weiteren Informationen mehr, geht er einfach weiter. Dies wird vom DTG erkannt und die Informationspräsentation automatisch gestoppt. Gleichzeitig wird der Navigator gestartet, um den Touristen zur nächsten Attraktion zu leiten.

Während der Tourist läuft, überwacht der DTG automatisch im Hintergrund den Fortgang der Tour. Dies gilt vor allem für die Laufgeschwindigkeit des Touristen. Gibt es hier eine gravierende Abweichung zu der vorher bestimmten (oder im Profil gespeicherten) Geschwindigkeit, kann dies eine Neuberechnung der Resttour zur Folge haben, wenn der Tourist z.B. nicht mehr alle Attraktionen in der vorgegebenen Zeit schaffen kann. Ähnliches kann auch passieren, wenn der Tourist von der geplanten Tour abweicht. Tritt ein solches Ereignis ein, prüft der DTG zunächst das Umfeld des Touristen. Befindet sich eine Attraktion oder ein Restaurant in seiner Nähe, dass den Interessen des Touristen entspricht, aber nicht in die Tour mit eingeplant wurde, erfolgt automatisch eine Anpassung der Tour durch Einfügen dieser neuen Attraktion. Der Navigator wird entsprechend umgestellt, so dass die neuen Navigationsanweisungen direkt zu dieser neuen Attraktion führen.

Dem Touristen kann so garantiert werden, dass seine Tour immer rechtzeitig zu Ende ist und er an seinem Zielpunkt ankommt.

Die folgenden Abbildungen zeigen Ausschnitte der Benutzeroberfläche:



Abb. 1: Screenshots

### 3 Stand der Technik

Schon seit längerem befassen sich Forschungsprojekte mit Tour Guides. Die folgenden Projekte erfassen Teile der Schwerpunkte des DTG zur Erstellung einer Tour.

- Das Crumpet Projekt (Schmidt-Belz, 2003b; Crumpet, 2004) ermöglicht es einem Mobil-  
Agenten verschiedene Sehenswürdigkeiten zu finden und auf einer Karte anzuzeigen  
bzw. eine Tour zu dieser zu berechnen.
- ⇒ Sehenswürdigkeiten werden nach deren Position angezeigt. Der Nutzer muss selbst  
entscheiden, welche Attraktionen interessant für ihn sind und ob er genug Zeit hat diese  
zu besuchen.
- Eine von Enarro entwickelte Software (Enarro, 2004) bietet dem Nutzer vordefinierte  
Touren mit den wichtigsten Attraktionen einer Stadt an. Solche Touren sind für einige  
große Städte bereits definiert worden. Der Tourist benötigt einen PDA, auf dem sich ein  
spezieller Player und die ausgewählte Tour befinden. Der Tourist benötigt zusätzlich  
noch eine spezielle Navigationssoftware, die ihn zu den einzelnen Plätzen der Tour  
führt. Die erreichten Attraktionen werden den Touristen dann audio-visuell vorgestellt.
- ⇒ Der Nutzer kann nur zwischen existierende Touren wählen. Die einzelnen Sehenswür-  
digkeiten sind von den Betreibern von Enarro so zusammengestellt, dass den Interessen  
einer breiten Massen von Touristen entsprechen.
- Das AgentCities-Projekt ermöglicht es einem Touristen Restaurants zu finden. Das Sys-  
tem enthält eine Restaurant-Ontologie und einen Restaurant-Guide Webservice. Die In-  
formationen sind statischer Natur und von bekannten Restaurantführern entnommen  
worden.
- ⇒ Da die Daten statisch sind, besteht keine Möglichkeit auf Veränderungen z.B. Öff-  
nungszeiten, Reservierungen usw. zu reagieren.

- In Verbindung mit dem AgentCities Framework entwickelte das „Fujitsu Laboratories of America“ (Fujitsu, 2004) einen Veranstaltungsplaner. Basierend auf den Interessen eines Touristen, wählt der Planer mittels einer Ontologie Restaurants aus und führt auf Wunsch Reservierungen durch.
- ⇒ Dies ist ein erster Schritt in Richtung context-awareness, denn die Suche nach einem Restaurant ist dynamisch und abhängig von den Interessen des Touristen.
- Vordefinierte Touren sind nicht der Schwerpunkt des DTG, da er beabsichtigt, individuelle Touren in Echtzeit zusammenzustellen. Durch Einbindung aktueller Daten von den Webservices der Betreiber, können lokale Gegebenheiten wie Öffnungszeiten beachtet und einbezogen werden. Die Vereinigung einer breiten Masse von kontext-sensitiven Daten, ermöglicht es dem DTG sehr flexibel auf externe Einflüsse zu reagieren.

## 4 Architektur

Schon heute besitzen viele Menschen mobile Endgeräte, einige schon einen MDA als Integration von PDA und Telefon und in wenigen Jahren wird mit DGPS eine Lokalisation von bis zu einem Meter möglich sein. Aufgrund dieser technischen Voraussetzungen wurde die Architektur des DTG geplant:

- **Mobiles Gerät**

Der Tourist besitzt ein mobiles Gerät z.B. PDA, MDA oder Smartphone, welches über einen GPS Empfänger die eigene Position ermitteln kann. Über GPRS oder UMTS besteht eine Verbindung zum Internet. Auf dem mobilen Gerät wird der DTG Agent und eine Standard Navigationssoftware wie Navigon ausgeführt.

- **Sehenswürdigkeiten**

Jede Attraktion der Stadt, als potentieller Bestandteil der Tour, wird durch einen Betreiber mit Hilfe eines AuthoringTools modelliert. Relevante Informationen werden durch Webservices (web services) nach außen verfügbar gemacht. Die Webservices werden in einem UDDI Verzeichnis registriert, wo sie anschließend abgefragt werden können.

- **DTG Server**

Ein zentraler Server ist für die Touranfragen zuständig. Zunächst werden alle Sehenswürdigkeiten der Umgebung des Nutzers ermittelt. Diese werden entsprechend den Interessen des Nutzers bewertet. Dies geschieht durch einen Semantic Match Algorithmus. Anschließend werden die bestbewerteten Attraktionen zu einer Tour zusammengestellt.

Während der Durchführung der Tour wird die Navigationssoftware eingesetzt, um Karten inklusive der Routen anzuzeigen und Weganweisungen über Kopfhörer auszugeben. So wird der Tourist von einer Sehenswürdigkeit zur nächsten geleitet. Im Hintergrund verfolgt der DTG den Ablauf der Tour und reagiert z.B. auf eventuelle Zeitdiskrepanzen mit einer Umgestaltung der Tour, sodass der Tourist pünktlich an seinem Zielort angelangt. Es wird eine Ontologie verwendet, um die Interessen-gebiete des Touristen

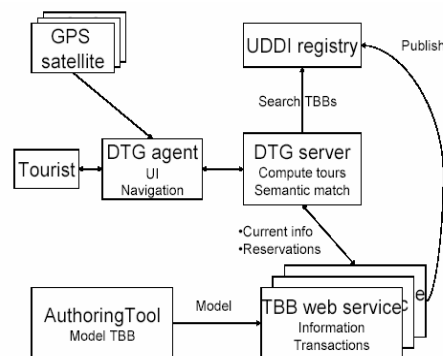
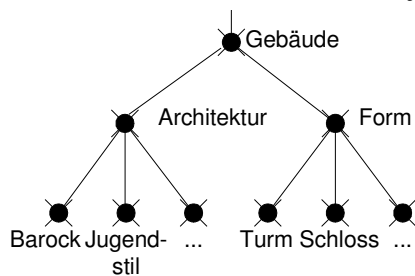


Abb. 2: Architektur des DTG

zu modellieren. Andererseits dient die Ontologie der Zuordnung von Sehenswürdigkeiten zu bestimmten Interessenkategorien. Auf Grundlage der Ontologie wird außerdem das Semantic matching der jeweiligen Profile durchgeführt.

## 5 Semantic matching

Die zentrale Herausforderung des DTG's ist die Bewertung und anschließende Auswahl der Sehenswürdigkeiten anhand der Nutzerinteressen. Dies erfordert eine Art künstliche Intelligenz, die eine Entscheidung trifft ob und inwieweit eine bestimmte Sehenswürdigkeit für einen Touristen von Interesse ist. Dies ist deshalb notwendig, weil in Städten wie Görlitz, allein im Zentrum über hundert Attraktionen vorhanden sind. Daher sind für unterschiedlichen Startpositionen und verfügbaren Zeiten auch viele unterschiedliche Touren möglich. Während ein erfahrener Stadtführer solche Empfehlungen für verschiedene Touristen geben könnte, stellt dies eine besondere Herausforderung für ein Computerprogramm dar. Es ist deshalb wichtig, dass die Software die Bedeutung von Daten wie Interessen interpretieren kann. Dazu ist eine gemeinsame Begriffsbasis notwendig, die alle möglichen Interessengebiete beschreibt und Beziehungen definiert – eine Ontologie. Dabei handelt es sich um ein Modell eines Ausschnitts der Realität. Jedes Konzept welches in der Realität existiert wird durch eine Klasse dargestellt. Die Beziehungen zwischen den einzelnen Klassen ergeben eine hierarchische Struktur, in der jede Klasse Eltern(übergeordnet)- und Kindklassen(untergeordnet) haben kann. Attribute beschreiben Klassen durch Angabe spezifischer Eigenschaften genauer.



Um eine solche Ontologie zu erstellen, wurden alle vorhandenen Sehenswürdigkeiten der Stadt Görlitz (Görlitz, 2003) ermittelt und in Kategorien eingeordnet. Daraus entstanden Bauwerke, Natur, Geschichte, Kunst & Kultur und Wissenschaft & Technik. Jede Kategorie ist noch weiter unterteilt. So können z.B. die Interessen aus dem Bereich Bauwerke nach Architekturstilen wie Barock oder Jugendstil und Bauwerksarten wie Schlösser oder Hallenhäuser unterschieden werden.

Benachbarte Klassen werden als semantisch ähnlich angesehen. Der Grad der Ähnlichkeit hängt von der gemeinsamen Superklasse ab. Wenn die Interessen eines Touristen nicht hundertprozentig zum vorhandenen Angebot an Sehenswürdigkeiten passen, dann sind Subklassen der Ontologie auch potentiell interessant für ihn. So wird z.B. für jemanden der an Tieren interessiert ist, wahrscheinlich auch Natur im Allgemeinen interessant sein. Daher werden auch Landschaften von gewissem Interesse sein.

Die Sehenswürdigkeiten werden mit Hilfe eines AuthoringTolls den Kategorien dieser Ontologie zugeordnet. Dies geschieht durch den Betreiber, wobei ein Modell der Sehenswürdigkeit mit allen verfügbaren Informationen (Interessengebiete, Öffnungszeiten usw.) angelegt wird. Die meisten Sehenswürdigkeiten sind mehreren Kategorien zugeordnet, z.B. wird eine Kirche unter Religion, sowie unter gotischer Architektur zu finden sein.

Touristen geben ihre Interessen ebenfalls aufgrund der hierarchischen Darstellung der Ontologie an, indem sie verschiedene Äste des Baumes auswählen. Die getroffene Auswahl wird

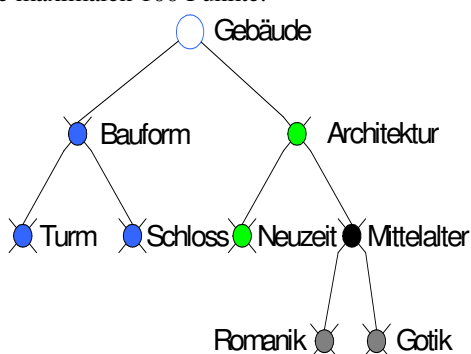
in einem Interessenprofil auf dem mobilen Gerät gespeichert, so dass es für weitere Verwendungen zur Verfügung steht. So kann dasselbe Profil auch in einer anderen Stadt, ohne erneute Eingabe der Interessen, genutzt werden.

Für eine Touranfrage wird die Ontologie aufgrund des Interessenprofils evaluiert. Die hierarchische Anordnung der Klassen einer Ontologie kann auch als gerichteter Graph dargestellt werden. Dieser wird durch zwei Funktionen, von den im Interessenprofil enthaltenen Klassen ausgehend, bewertet. Jeder Knoten erhält somit eine Punktzahl, die den Grad der Ähnlichkeit zum ausgewählten Interessengebiet darstellt.

Das Bewertungsprinzip wird in folgendem Beispiel demonstriert:

Angenommen ein Tourist interessiert sich für die Epoche des Mittelalters.

Ein Gebäude aus dem Bereich Mittelalter trifft dieses Interesse vollständig und erhält daher die maximalen 100 Punkte.



Ein Gebäude welches dem gotischen Stil entstammt erhält ebenfalls 100 Punkte, da Gotik eine Spezialisierung von Mittelalter ist.

Ein Gebäude aus der Neuzeit erhält die Hälfte der Punkte, da sie dieselbe Superklasse haben.

→ Eine Übereinstimmung der Superklassen zweiten Grades ergeben somit 25 Punkte.

Alle Sehenswürdigkeiten die kein Gebäude darstellen erhalten 0 Punkte.

Konkret bewertet der Algorithmus die

Ontologie als gerichteten Graphen anhand des Interessenprofils. Ausgangspunkt ist der Knoten, welcher das ausgewählte Interessengebiet darstellt. Dieser wird mit 1 bewertet. 2 Funktionen sorgen für die Bewertung des Restbaumes, wobei jeder Knoten nur einen Elternknoten haben darf.

Für jeden Knoten höherer Ebene werden die Punkte gemäß der Formel (1) halbiert:

$$y \leftarrow f^u(x) = \frac{1}{2}x \tag{1}$$

Jeder Knoten niederer Ebene erhält durch Formel (2) dieselben Punkte wie sein Elternknoten:

$$y \leftarrow f^d(x) = x \tag{2}$$

Mit Knoten B als Startpunkt sieht der bewertete Graph wie folgt aus:

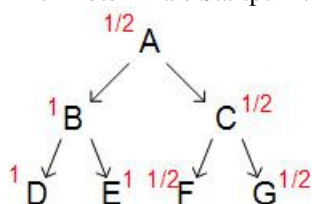


Abb. 3: Bewerteter Graph

Für Knoten D und E greift die Funktion  $f^d(x)$ , sodass diese ebenfalls mit 1 bewertet werden. Der Knoten A erhält  $\frac{1}{2}$  durch die Funktion  $f^a(x)$ . Alle Unterknoten erhalten wiederum  $\frac{1}{2}$  durch die Funktion  $f^d(x)$ .

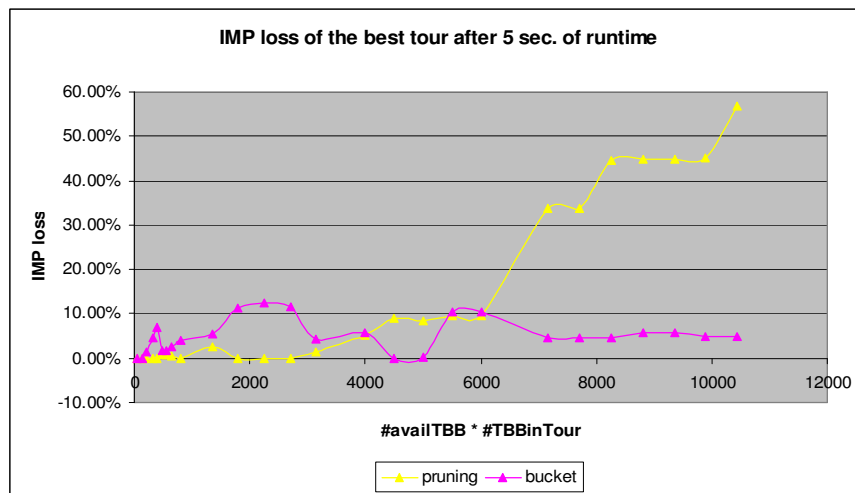
Jedem dieser Knoten sind bestimmte Sehenswürdigkeiten zugeordnet. Die Punkte der einzelnen Sehenswürdigkeiten ergeben sich aus der Summe der Punkte ihrer zugehörigen Knoten. Die Sehenswürdigkeiten mit den meisten Punkten werden bevorzugt in die Tour aufgenommen, sofern sie in der Restzeit der Tour erreichbar sind.

## 6 Tourberechnung

Nachdem der Semantic Match Algorithmus die einzelnen Sehenswürdigkeiten mit Punkten versehen hat, wird eine Tour berechnet. Eine Tour ist eine Sequenz von Sehenswürdigkeiten. Die Berechnung dieser Tour stellt ein Optimierungsproblem dar. Das heißt, es ist eine Route mit möglichst vielen Sehenswürdigkeiten zu finden, die in der angegebenen Zeit abgelaufen werden kann. Da 20 Sehenswürdigkeiten mit gleichem Anfangs- und Zielpunkt zu  $(20-1)!/2 = 6 \cdot 10^{16}$  verschiedenen Touren führen, ist eine vollständige Berechnung in einer annehmbaren Zeit nicht zu schaffen. Auch eine vorherige Berechnung und Speicherung aller möglichen Touren ist aufgrund der Vielfältigkeit nicht möglich. Das Ergebnis muss innerhalb weniger Sekunden ermittelbar sein und möglichst nah an der optimalen Lösung liegen. Einem Touristen ist es nicht wichtig, die Tour mit den meisten Punkten zu haben. Er möchte lieber eine möglichst optimale Tour haben, die ihn interessiert und in akzeptabler Zeit (max. 5 Sekunden) berechnet ist.

Der angewandte Algorithmus verwendet eine Tiefensuche. Um eine Berechnung identischer Touren zu verhindern, wird die Kandidatenliste durch Entfernen der ausgewählten Sehenswürdigkeit angepasst. Bei steigender Anzahl von Sehenswürdigkeiten ist diese Vorgehensweise sehr effektiv.

**Abb. 4** zeigt vergleichsweise zwei Algorithmen für die Suche einer Tour. Die x-Achse stellt dabei das Produkt von verfügbaren Tourbausteinen (#availTBB) und der Anzahl der ausgewählten Tourbausteine bzw. der Tiefe der Rekursion (#TBBinTour) dar. Dem gegenüber ist in der Y-Achse der Punkteverlust im Verhältnis zur optimalen Punktezahl (IMPloss) dargestellt.



**Abb. 4:** Benchmarkergebnis Tourberechnung

Der „bucket“-Algorithmus teilt die Kandidatenliste in mehrere Teile, die jeweils separat verarbeitet werden. Durch das Weiterverarbeiten der Kandidaten eines weiteren „buckets“ wird das Ergebnis noch weiter verbessert. Wie aus **Abb. 4** zu entnehmen ist, beträgt der Verlust der Punkte in einer Destinationen bestehend aus mehr als 1000 TBBs mit jeweils 10 TBBs je Tour weniger als 7%.

## 7 Touradaption

Während der Tourist die Tour, geleitet von einer Navigationssoftware, abläuft, ermittelt und überwacht der DTG im Hintergrund die Position und Laufgeschwindigkeit des Touristen. Diese Informationen zusammen mit dem Interessensprofil machen den persönlichen Kontext aus. Teile dieser Daten werden ständig im Hintergrund aktualisiert. So kann z.B. eine gravierende Veränderung der Laufgeschwindigkeit eine Neuberechnung der Tour zur Folge haben (z.B. durch Entfernen eines TBB), um so zu garantieren, dass der Tourist in der vorgegebenen Zeit zurück ist.

Nähert sich der Tourist einer Stelle, an der die zu besuchende Sehenswürdigkeit sichtbar wird, macht der DTG den Touristen darauf mit einleitenden Informationen aufmerksam. Um dabei Verwirrungen zu vermeiden, muss die Richtung, aus der der Tourist kommt beachtet werden. Anschließend wird die Navigationssoftware vom DTG unterbrochen und tritt dadurch in den Hintergrund.

Solange sich der Tourist in der Nähe der Sehenswürdigkeit aufhält, bekommt er Informationen, z.B. zu dessen Geschichte, in Form von Audiosignalen über einen Kopfhörer. Auch hierbei werden die Informationen an seine aktuelle Position angepasst. Das heißt, wenn er sich z.B. an der Rückseite eines Gebäudes befindet oder in dessen Innenhof, erhält er spezielle Informationen darüber. Entfernt sich der Tourist von der entsprechenden Sehenswür-



digkeit, wird die Informationswiedergabe nach einer kurzen Verabschiedung gestoppt. Danach startet der DTG wieder die Navigation zur nächsten Station.

Trotz Navigationshinweisen könnten einige Touristen sich verlaufen oder gezielt den vorgeschlagenen Weg verlassen um eine andere Attraktion zu besuchen. Der DTG muss die Situation richtig beurteilen und entscheiden welcher Fall vorliegt. Die Navigationssoftware hingegen versucht ständig den Touristen auf den ursprünglichen Weg zurückzubringen. Beim Annähern an eine andere Sehenswürdigkeit, die auf dem neuen Weg liegt und den Interessen des Touristen entspricht, ermittelt der DTG kurzfristig vorhandene Informationen darüber und stellt diese bereit. Beim Besuch eines Kaufhauses oder Restaurants wartet der DTG bis der Tourist aus dem Gebäude kommt und die Tour fortsetzen möchte und berechnet diese für die Restzeit neu.

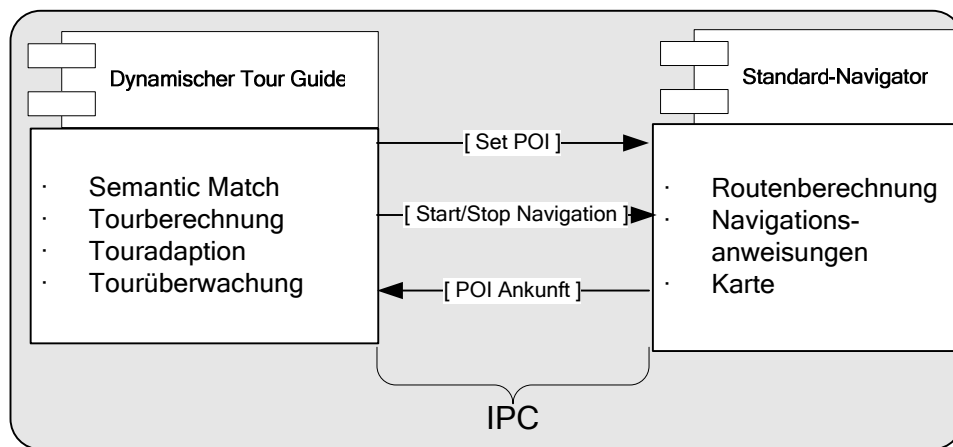


Abb. 5: Interaktion DTG mit Navigator

## 8 Schlussfolgerung

Unabhängig von der aktuellen Position und Zeit ermittelt der DTG alle notwendigen Informationen um eine individuelle, kontextbasierte Tour zu berechnen. Dazu entdeckt und befragt er die verfügbaren Webdienste. Der Fortschritt der Tour wird permanent überwacht, sodass auf zeitliche Verschiebungen mit einer Anpassung der Tour reagiert werden kann.

Für die richtige Auswahl der Sehenswürdigkeiten entsprechend der angegebenen Interessen sorgt ein Semantic Match Algorithmus. Aufgrund einer gemeinsamen Ontologie werden alle Sehenswürdigkeiten modelliert und anhand der persönlichen Interessen eines Touristen bewertet. Diese Bewertungen werden von einem Tiefensuch Algorithmus verwendet um eine Tour zu erstellen, die für die verfügbare Zeit optimal ist. Umfangreiche Benchmarks

haben ergeben, dass bereits in den ersten 5 Sekunden eine Tour gefunden werden kann, die maximal 5% von der optimalen Tour abweicht.

Durch diese neue Technologie wird es einem Touristen ermöglicht, ein Reiseziel, durch eine individuelle Tour mit aktiver Unterstützung durch den DTG, in Echtzeit zu erfahren. Außerdem verhilft der DTG zu einer breiteren Verteilung der Touristen über das gesamte Zielgebiet, abhängig von ihren Interessen.

## Danksagungen

Dieses Projekt ist Teil des Forschungsvorhabens VESUV (Verteilte Agenten für rechtsverbindliche Aufgabendelegation) und wird vom Bundesministerium für Wirtschaft und Arbeit (BMWA) gefördert. Wir danken den Anregungen der Projektteilnehmern von der Siemens AG, dem Microsoft European Innovation Center (EMIC), der Stadt Görlitz und dem Fraunhofer Institut (IGD).

## Literatur

- AgentCities (2004): <http://www.agentcities.org>
- Bussler, Christoph; Cardoso, Jorge; Fensel, Dieter; Sheth, Amit (2002): *Semantic web services and processes: Semantic composition and quality of service*, Irvine CA
- Crumpet (2004): <http://www.ist-crumpet.org/>
- Fujitsu (2004): <http://sf.us.agentcities.net/sf/>
- Godart, Jean-Marc (2003); *Beyond the Trip Planning Problem for Effective Computer-Assisted Customization*, Information and Communication Technologies in Tourism 2003, Andrew Frew et al. (eds.), Springer Computer Science
- Enarro (2004): <http://www.enarro.com>
- Goerlitz (2003): <http://www.goerlitz.de/de/tourimus/kunst/kultur-denkmale2003.pdf>
- Horrocks, Ian; Li, Lei (2003): *A software framework for matchmaking based on semantic web technology*
- Korkea-aho, Mari (2000): *Context aware applications survey*, Helsinki University of Technology. <http://www.hut.fi/~mkorkeaa/doc/context-aware.html#chap3.2>
- Lopez, Beatriz (2003): *Holiday Scheduling for City Visitors*, Information and Communication Technologies in Tourism 2003, Andrew Frew et al. (eds.), Springer Computer Science
- Maedche, Alexander; (2003), Staab, Steffen; *Services on the Move: Towards a P2P-Enabled Semantic Web Services*, Information and Communication Technologies in Tourism 2003, Andrew Frew et al. (eds.), Springer Computer Science
- Navigon (2004): <http://www.navigon.de>
- Schmidt-Belz, Barbara; Posland, Stefan (2003a): *User Validation of a mobile Tourism Service*; Workshop "HCI mobile Guides", Udine (Italy)
- Schmidt-Belz, Barbara; Laamanen, Heimo; Poslad, Stefan; Zipf, Alexander (2003b): *Location-based Mobile Tourist Services – First User*, Information and Communication Technologies in Tourism 2003, Andrew Frew et al. (eds.), Springer Computer Science