

Architektur für die Verarbeitung von Kontextinformationen – Architekturkonzept –

Interner Forschungsbericht
vorgelegt

von
Dipl.-Ing. Agnieszka Lewandowska

28. September 2007



Inhaltsverzeichnis

1	Einleitung und Aufbau	3
2	Überblick über die Architektur von SFINKS	4
3	Erfassen und Sammeln der Kontextinformationen	6
3.1	Direkte Erfassung der Kontextinformationen	6
3.2	Indirekte Erfassung der Kontextinformationen	7
3.3	Erfassung der Kontextinformationen mit Hilfe von Sensoren	7
3.4	Optimale Ausgangsposition zur Kontexterfassung	8
3.5	Eigenschaften von Kontextinformationen	9
4	Übertragung der Kontextinformationen	11
5	Auswerten der Kontextinformationen	13
5.1	Datenbankstruktur zur Speicherung von Kontextinformationen	13
5.2	Ordnungsprozess	15
5.3	Profil des Benutzers	15
6	Verarbeiten und Anwenden der Kontextinformationen	19
6.1	Beobachtungsprozess	19
6.2	Suchprozess	20
6.3	Adaptierende Prozesse	21
6.4	Priorisierungsprozess	22
6.5	Vorgehen beim Verarbeiten und Anwenden der Kontextinformationen	23
7	Zusammenfassung der Architekturdarstellung	25
	Literatur	27

1. Einleitung und Aufbau

In modernen Kommunikationsnetzen wächst mit jedem Tag die Bedeutung kontextsensitiver Dienste. Daneben ist auch die Attraktivität der Mobilfunknetze ungebrochen. So möchte der Benutzer heutzutage Informationen und Dienste auch über sein mobiles Endgerät abrufen können. Diese Möglichkeit des Dienstzugriffs kann verbessert werden, indem man Informationen und Dienste kontextsensitiv adaptiert und damit dem Benutzer einen optimal angepassten Dienst bietet. Der Kontext drückt sich dabei zum einen in den Fähigkeiten und Interessen des Benutzers und zum anderen in den Möglichkeiten, die das mobile Endgerät und der aktuelle Netzzugang bieten, aus.

Um die kontextsensitiven Dienste erbringen zu können, ist eine entsprechende Architektur notwendig. Hier steht man vor der Aufgabe ein System zu entwickeln, was sämtliche für den Verwendungsfall gestellten Anforderungen erfüllt, gleichzeitig aber so modular und flexibel ist, um auch andere Einsatzvoraussetzungen zu bedienen. Die Kontextinformationen müssen in geeigneter Form ausgewertet werden. Dann müssen auf den Kontext angepasste Anwendungen bzw. Dienste gesucht und aufgerufen werden. Ein Vorschlag einer solchen Architektur wird in diesem Bericht dargestellt.

Am Anfang wird ein allgemeiner Überblick über die gesamte Architektur gegeben, sodass in den folgenden Abschnitten bestimmte Teile des Systems beschrieben werden können. Es wird zuerst erläutert, wie die Kontextinformationen des Benutzers erfasst und gesammelt werden können. Das Erfassungsverfahren kann auf unterschiedliche Weise erfolgen, was später detailliert ausgeführt wird. Außerdem wird eine Methode vorgeschlagen, die die Kontexterfassung optimiert.

Dann erfolgt eine Beschreibung, wie die Informationen zwischen einem Benutzer und dem System übertragen werden können und welche Einschränkungen hierbei berücksichtigt werden sollten. Dabei wird die Konzeption für eine Kommunikationsinfrastruktur vorgestellt.

Der nächste Abschnitt befasst sich mit dem Auswerten der Kontextinformationen. Hier wird die Datenbankstruktur zur Speicherung des Benutzerkontextes und anderer benötigter Informationen beschrieben. Dazu wird gezeigt, wie sie mit Hilfe des Ordnungsprozesses in der Datenbank gruppiert werden. Aufgrund der erfassten Informationen wird ein Benutzerprofil erstellt.

Dann werden die Verarbeitung und die Anwendung der Kontextinformationen behandelt. Es wird dargestellt, wie die vom Benutzer oder System angeforderten Informationen und Dienste entsprechend dem Kontext abgerufen und angepasst werden können und was dabei zu berücksichtigen ist. An dieser Stelle wird die ganze Struktur der Informationsverarbeitung dargestellt.

2. Überblick über die Architektur von SFINKS

Um die kontextsensitiven Telekommunikationsdienste zu liefern, wurde ein kontextsensitives System **SFINKS** (*flexibles System für Informationen in kontextsensitiven Telekommunikationsdiensten*) entworfen. Das Hauptziel des Systems ist, einem mobilen Benutzer Informationen und Dienste, die an genau ihn und seinen Kontext angepasst worden sind, zu erbringen [LDS05].

Das System basiert auf einem Client-Server-Modell. Das bedeutet, dass prinzipiell alle erfassten Informationen an einen Server übermittelt und erst dort bearbeitet werden. Das System wird die Kontextinformationen eingangs erfassen und sammeln, dann an den Server übertragen, dort werden sie ausgewertet, schließlich verarbeitet und angewandt. Die Architektur von SFINKS ist in Abbildung 1 zu sehen.

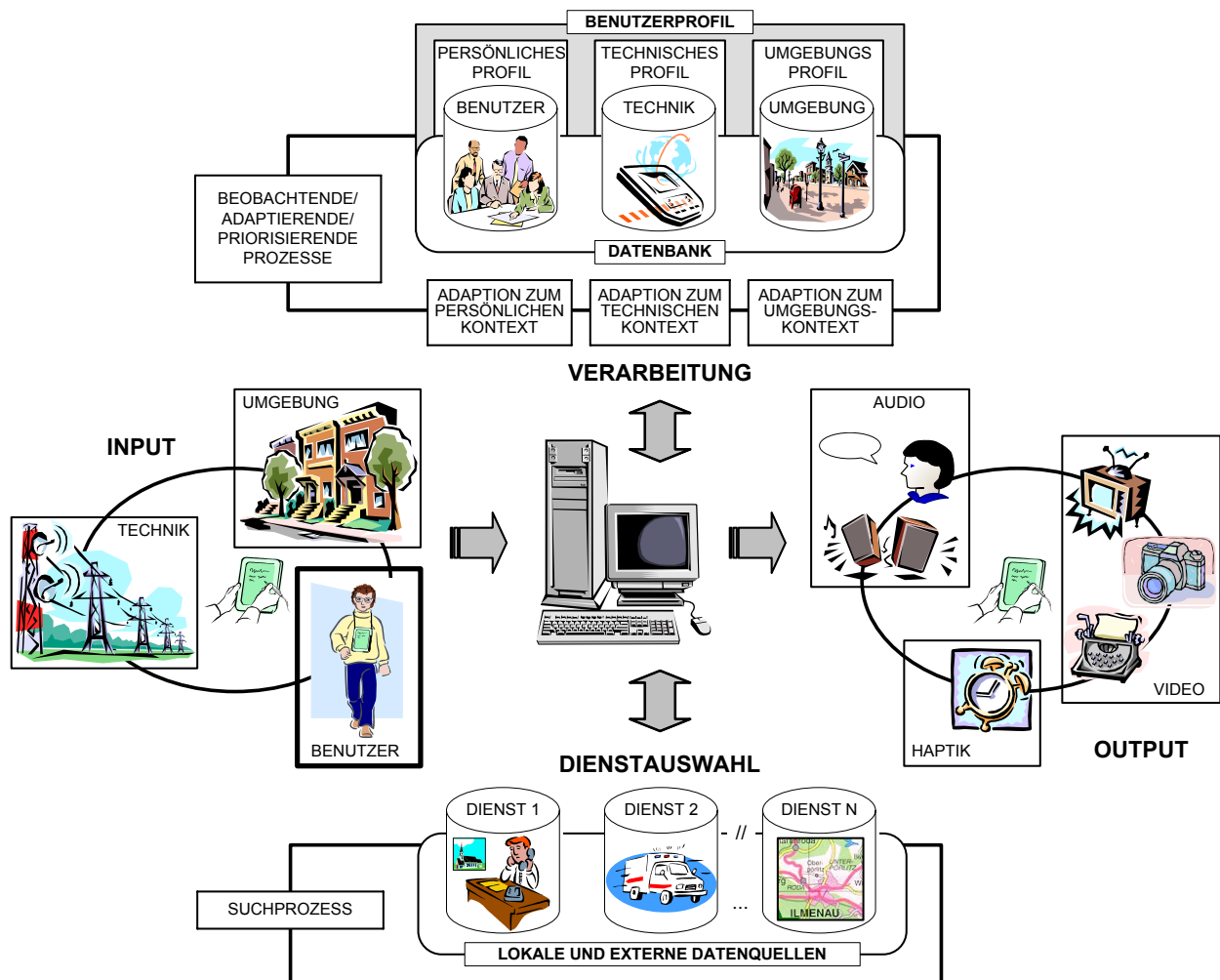


Abbildung 1: Architektur von SFINKS zur Erbringung kontextsensitiver Dienste

Der **Input des Systems** liefert die Kontextinformationen. Es geht um die Informationen über den Benutzer selbst, das Endgerät, vorhandene Kommunikationsnetze sowie die aktuelle Umgebung des Benutzers. Die Kontexterfassung kann beispielsweise direkt, indirekt oder mit

Hilfe von Sensoren erfolgen. Noch eine Möglichkeit wäre, alle diese Methoden gleichzeitig zu verwenden. Die erfassten Kontextinformationen können aber verschiedenartig sein, deshalb benötigt man Eigenschaften, mit denen sie beschrieben werden können. Mehr darüber wird im Abschnitt „Erfassen und Sammeln der Kontextinformationen“ beschrieben.

Nachdem die Kontextinformationen erfasst worden sind, werden sie an den Server übermittelt. Zur **Kommunikation** mit dem Server sind unterschiedliche Netztechniken wie WLAN, GSM, UMTS, GPRS, analoges Telefon, ISDN usw. vorgesehen. Jedoch spielen an dieser Stelle die Funktechniken eine herausragende Rolle. Der Abschnitt „Übertragung der Kontextinformationen“ konzentriert sich auf die Kommunikationsinfrastruktur für SFINKS.

Die an den Server gesendeten Kontextinformationen werden entsprechend ausgewertet, verarbeitet und angewendet. Sie laufen zuerst durch einen Ordnungsprozess, der sie in der **Datenbank** speichert. Die Konzeption einer Datenbankstruktur zur Speicherung des Benutzerkontextes wird im Abschnitt „Auswerten der Kontextinformationen“ dargestellt. Aufgrund der erfassten Informationen wird ein **Benutzerprofil**, das auf drei Arten von Kontextinformationen, nämlich auf dem persönlichen Kontext, dem technischen Kontext sowie dem Umgebungskontext, basiert, erzeugt. Das Benutzerprofil wird für die Ausführung kontextsensitiver Dienste benötigt.

Die in der Datenbank verarbeiteten Kontextinformationen werden später von verschiedenen Prozessen zur Anpassung von Diensten und Informationen verwendet (**Verarbeitung**). Die kontextsensitiven Dienste können vom Benutzer durch eine direkte Anfrage oder vom System als eine Reaktion des Beobachtungsprozesses abgerufen werden. In jedem Fall werden zuerst Dienste und Informationen in den lokalen und externen Datenquellen gesucht (**Dienstauswahl**). Der Suchprozess erfolgt sehr allgemein und liefert eine Anzahl von Ergebnissen. Erst die adaptierenden Prozesse ermitteln mit Hilfe des Benutzerprofils die für den aktuellen Kontext passenden Dienste. Das Gesamtbild der Verarbeitungsprozesse wird im Abschnitt „Verarbeiten und Anwenden der Kontextinformationen“ vorgestellt.

Der **Output des Systems** liefern kontextsensitive Informationen und Dienste. Aufgrund der Sicherheitsanforderung, dass der Benutzer auch bei einem Systemausfall mit den wichtigsten Daten versorgt wird, werden sie vom Priorisierungsprozess nach Wichtig- und Dringlichkeiten gewichtet und dann an den Benutzer gesendet. Sie werden dem Benutzer je nach dessen Endgerät über unterschiedliche Ausgabemethoden wie z. B. Video, akustisch oder haptisch präsentiert.

3. Erfassen und Sammeln der Kontextinformationen

Der erste Schritt, um einem Benutzer kontextsensitive Informationen und Dienste liefern zu können, ist die Erfassung der Informationen über den Kontext des Benutzers. Danach kann der Benutzerkontext ausgewertet und weiter verarbeitet werden. Das erste Problem besteht darin, wie die zur Beschreibung notwendigen Informationen gesammelt werden können [LDS05].

Der Vorgang der Kontexterfassung kann von vielen verschiedenen Quellen erfolgen. Zum Beispiel erfordert die Anwesenheit des Benutzers in einem Büro eine Informationssammlung von mehreren unterschiedlichen Sensoren. Informationen, die auf sensorischem Weg gewonnen werden, können mit Messfehlern behaftet sein. Deshalb verknüpft man die Erfassungsergebnisse von mehreren Sensoren, um ein möglichst fehlerfreies Gesamtbild einer Situation zu bekommen.

Die Kontexterfassung ist bei der Nutzung mobiler Endgeräte noch schwieriger. Sie ermöglichen dem Benutzer maximale Bewegungsfreiheit. In der Konsequenz können sich die Kontextinformationen wie z. B. der Aufenthaltsort des Benutzers oder die Menschen und Objekte in seiner Umgebung usw. schnell ändern. Damit sich das System diesen Änderungen anpassen kann, sollten sie in Echtzeit registriert werden. In den meisten Fällen besitzt der Benutzer sehr begrenzte Möglichkeiten die Informationen mit seinem mobilen Endgerät aufzunehmen. Deswegen kann der Benutzerkontext vom Endgerät nicht vollständig erkannt werden. Also erfolgt die Kontexterfassung von verschiedenen Quellen aus und die Informationen werden auf einem Server gesammelt. Somit basiert das System auf einer Client-Server Architektur.

3.1 Direkte Erfassung der Kontextinformationen

Die Kontextinformationen können auf unterschiedliche Weise erfasst werden. Eine Möglichkeit besteht darin, dass die Kontextinformationen direkt durch den Benutzer erfasst und dem System übergeben werden. Diese Methode wird nach Poslad [PLM01] auch *direkte Erfassung* (*explicit feedback*) genannt (Abb. 2). Die Informationen werden dann beispielsweise durch direktes Fragestellen oder Rundfragen an den Benutzer bzw. durch Ausfüllen von Fragebögen durch den Benutzer gewonnen und dem System über ein entsprechendes Benutzerinterface übergeben. Diese Methode ist zwar sehr genau und liefert glaubwürdige Ergebnisse, der Benutzer wird jedoch bei seiner aktuellen Tätigkeit gestört. Er verliert dadurch Zeit und damit stellt sich eventuell eine Unzufriedenheit gegenüber dem System ein. Des Weiteren zeigt die Erfahrung, dass nur ungern Formulare ausgefüllt oder langwierige Eingaben getätigt werden. Hinzu kommt, dass sich hierbei auch die kleine Displaygröße von Handhelds nachteilig auswirkt. Die Fragestellungen und Eingaben müssen so eindeutig und einfach gestaltet sein, dass der Benutzer mit der Beant-

wortung nicht überfordert wird. Ihm muss zu jedem Zeitpunkt klar sein, welche Informationen für die Kontextbeschreibung wesentlich und angemessen sind.

3.2 Indirekte Erfassung der Kontextinformationen

Das Gegenteil zur Methode der direkten Erfassung wird in [PLM01] als *indirekte Erfassung* (*implicit feedback*) definiert. Hier werden die zur Beschreibung des Kontextes notwendigen Informationen durch das Beobachten bzw. Verfolgen der Interaktionen zwischen dem Benutzer und dem System gewonnen (Abb. 2). Das System registriert sozusagen die Verhaltensweise des Benutzers. Wenn dieser z. B. immer wieder bestimmte Informationen vom System abrufen kann, kann dadurch festgestellt werden, dass sich der Benutzer für diese ganz bestimmte Art von Information interessiert. Diese Informationen können dann entsprechend für ihn aufbereitet und angeboten werden. Andererseits könnte er auch auf neue Informationen dieser Art aufmerksam gemacht werden. Der Vorgang, der die Erkennung wiederholter Anfragen eines Benutzers ermöglicht, ist auch als History-Funktionalität bekannt. Ein großer Vorteil dieser Methode besteht darin, dass der Benutzer bei seiner Tätigkeit weder unterbrochen noch anderweitig gestört wird. Außerdem trägt der Benutzer auch keine Verantwortung bezüglich der Richtigkeit der aufgenommenen Informationen. Man kann davon ausgehen, dass sich die durch das System gewonnenen Informationen mit den realen Interessen und Präferenzen des Benutzers decken. Nachteilig ist jedoch, dass die gesicherte Feststellung der Interessen und Präferenzen des Benutzers eine entsprechend hohe Anzahl von detektierten Ereignissen erfordert, was wiederum eine gewisse Zeitdauer in Anspruch nimmt. Das System muss also erst die Vorlieben des Benutzers lernen.

3.3 Erfassung der Kontextinformationen mit Hilfe von Sensoren

Die dritte Möglichkeit Kontextinformationen zu erfassen, besteht darin, Sensoren zu verwenden (Abb. 2). Mit Hilfe von Sensoren ist es möglich, Informationen über den Benutzer, das Endgerät oder die Umgebung zu sammeln. Die Baugröße der Sensoren ist auf ein Minimum gesunken, deshalb sind aktuelle Sensoren für beliebige Bereiche nutzbar. Sie können sich dazu auf dem Körper des Benutzers, in bzw. an den mobilen Endgeräten oder in der unmittelbaren Umgebung des Benutzers befinden. Unsichtbar und vom Benutzer nicht wahrnehmbar passen sie sich in jede Umgebung ein und liefern die gewünschten Kontextinformationen. Solche Informationen können Temperatur (Körper, Umgebung), Helligkeit, Luftfeuchtigkeit, Luftverschmutzung, Bewegungen o. ä. sein. Der Benutzer merkt von dieser Informationserfassung gar nichts. So kann mit Hilfe von Sensoren im Fußboden eines Raumes detektiert werden, ob sich hier gerade eine Person befindet oder nicht. Soll auch erkannt werden, um welche Person es sich dabei handelt, müsste diese zusätzlich mit einem Sender o. ä. ausgestattet sein, der beispielsweise eine eindeu-

tige Identifikationsnummer an den Sensor übermittelt. Ein weiteres Anwendungsgebiet von Sensoren ist z. B. das Bestimmen der Geschwindigkeit des Benutzers bzw. der Feststellung seiner Blick- oder Bewegungsrichtung. Nach Auswertung dieser Daten können ihm dann z. B. bei höheren Bewegungsgeschwindigkeiten nur die wichtigsten ortsabhängigen Informationen übermittelt werden, während ihm bei kleineren Geschwindigkeiten stets detaillierte Informationen zur Verfügung gestellt würden, da nun mehr Zeit verbleibt, um diese auch lesen bzw. nutzen zu können. Diese Informationen können auch, wie z. B. in [WSK03] dargelegt, als Audio oder Video ausgegeben werden. Vorteilhaft ist die Unabhängigkeit zum System selbst. Das heißt, das System fragt nur zu den Zeitpunkten den zugehörigen Sensor ab, wenn die entsprechende Information auch benötigt wird. Als nachteilig können der erhöhte Hardwareaufwand und die damit verbundenen Management-, Wartungs- und Administrationsaufgaben gesehen werden.

3.4 Optimale Ausgangsposition zur Kontexterfassung

Die Anwendung dieser drei oben dargestellten Methoden ist auch im Einzelfall von der Art der Kontextinformation abhängig. Es gibt Informationen wie z. B. Name oder Alter des Benutzers, die nur direkt vom Benutzer gewonnen werden können. Wenn es aber um die Positionsbestimmung des Benutzers geht, stehen mehrere Möglichkeiten zur Auswahl, obwohl es in diesem Fall sinnvoll ist, entsprechende Sensoren zu verwenden.

Diese drei Methoden unterscheiden sich auch in ihrer möglichen Fehlerhaftigkeit. Informationen, die durch Beobachtungen oder Sensoren gewonnen werden, können mit Messfehlern behaftet sein. Bei einer direkten Anfrage ist die Wahrscheinlichkeit der Fehler geringer. Sensoren und die indirekte Methode ermöglichen auch eine Datenerfassung ohne direkte Einflussnahme des Benutzers. Er weiß nicht, welche personenbezogenen Daten über ihn erhoben werden, was zum Problem des Datenschutzes führen kann.

Die drei Methoden der Kontexterfassung werden in Abbildung 2 dargestellt.

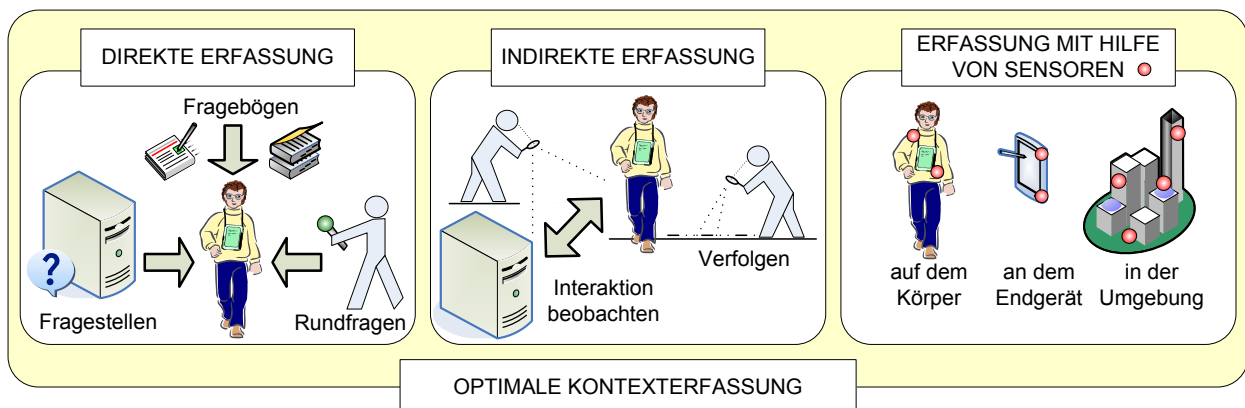


Abbildung 2: Erfassung des Kontextes

Anhand der Vor- und Nachteile kann man feststellen, dass keine dieser Methoden allein die Anforderungen eines umfassenden kontextsensitiven Systems erfüllen könnte. Deshalb muss je nach Verwendungszweck zwischen den drei Methoden gewählt oder kombiniert werden, um so eine optimale Ausgangsposition zur Erfassung des Kontextes zu schaffen.

3.5 Eigenschaften von Kontextinformationen

Am Anfang dieses Abschnittes wurde auf das Problem der Kontexterfassung hingewiesen. Das zweite Problem sind die Kontextinformationen selbst. Sie können, wie das schon erwähnt wurde, aus verschiedenen Quellen abgeleitet werden. Dadurch sind sie auch nach Art und Qualität sehr unterschiedlich. Einige davon ändern sich so schnell, dass sie schon nach kurzer Zeit veraltet sind. Dazu können sie fehlerbehaftet oder doppeldeutig sein. Das alles führt dazu, dass die Kontextinformationen in einem System sehr schwer zu verwenden sind. Dennoch lassen sich einige Kontextinformationen allgemein nach [HIR02] durch *Eigenschaften* wie zeitliche Veränderungen des Kontextes, Fehlerhaftigkeit, alternative Repräsentationsformen sowie Abhängigkeiten und Wechselbeziehungen beschreiben.

Aus Sicht der *zeitlichen Veränderungen* können Kontextinformationen statisch oder dynamischen sein. Informationen sind statisch, wenn sie unveränderliche Eigenschaften beschreiben wie z. B. ein Geburtsdatum. Die Mehrheit der Kontextinformationen ist aber dynamisch, ändern sich also über die Zeit. Die zeitlichen Eigenschaften der Kontextinformationen sind auch bei der Auswahl der Erfassungs- und Übermittlungsmethoden behilflich. Abbildung 3 fasst nach [DLS05] die Übermittlungsarten der Kontextinformationen zusammen. Sie zeigt Beispiele für Informationen zu den Methoden einmalige Übermittlung, Übermittlung bei Statuswechsel und periodische Übermittlung.

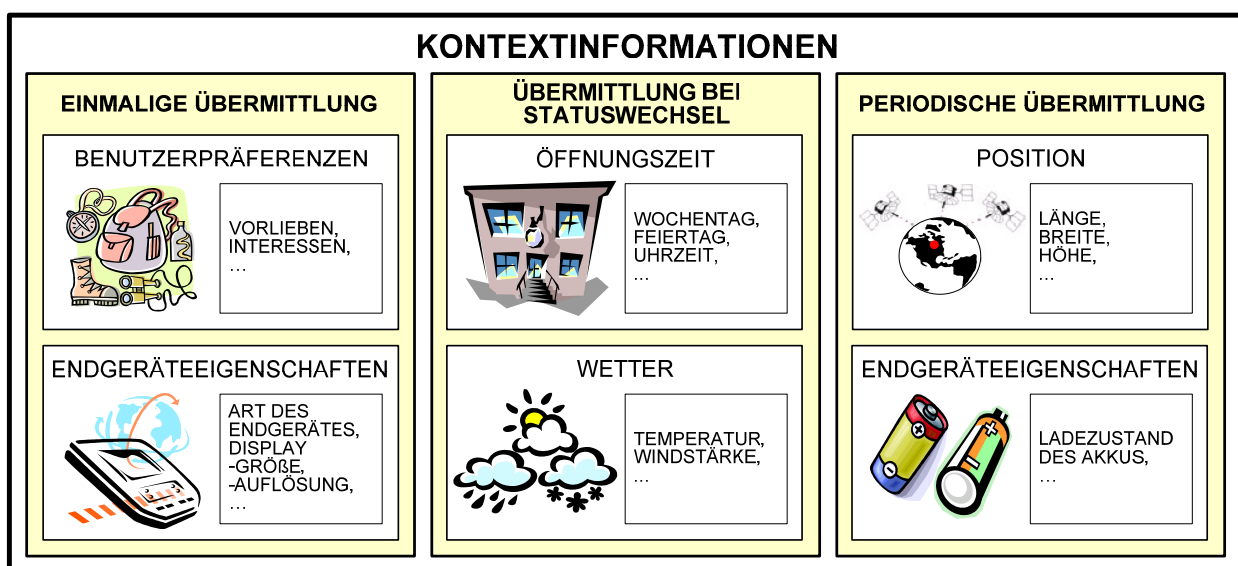


Abbildung 3: Übermittlungsarten der Kontextinformationen

Um statische Informationen zu bekommen, kann der Benutzer direkt gefragt werden. Solche Informationen werden üblicherweise nur einmalig an den Server übertragen. Sie sollten aber auch beim Statuswechsel gesendet werden, wenn sich ein Zustand ändert oder wenn ein festgelegter Grenzwert über- oder unterschritten wird. Zur Ermittlung sich häufig ändernder Informationen, die regelmäßig übertragen werden sollen, ist es günstiger, Sensoren zu verwenden.

Über die *Fehlerhaftigkeit* der Kontextinformationen wurde schon im vorigen Abschnitt diskutiert. Die erfassten Informationen können falsch, ungenau oder unvollständig sein. Das kann von den Erfassungsmethoden oder aber auch von den Kontextquellen, die zum gleichen Sachverhalt widersprüchliche Informationen liefern, abhängig sein. Dazu kommt eine Verzögerungszeit der Kontextinformationen, die zwischen Erfassung und Anwendung auftritt. Es geht um Informationen, die sich besonders häufig ändern. Obwohl sie korrekt gemessen wurden, sind sie schon nach kurzer Zeit wegen der Alterung falsch. Um das soweit wie möglich zu vermeiden, sollten dynamische Informationen mit einem Zeitstempel, der den Zeitpunkt ihrer Erfassung dokumentiert, ausgestattet werden.

Kontextinformationen, die von Sensoren bereitgestellt wurden, können oft nicht direkt von einer Applikation verwendet werden. Hier geht es um Low- und High-Level- Kontextinformationen. Das liegt an unterschiedlichen Formaten oder aber auch am Abstraktionslevel dieser Informationen. Deshalb müssen sie zuerst verarbeitet werden. Mit Hilfe von geeigneten Programmen lassen sich *alternative Repräsentationsformen* für die Informationen erzeugen. Derentwegen wurde in dieser Arbeit angenommen, dass alle benötigten Informationen durch solche Programme unabhängig vom System entsprechend vorbereitet werden. Dadurch werden die Beschreibungen solcher Verfahren an diesen Stellen vermieden.

Aus den zeitlichen Verläufen der Kontextinformationen lassen sich auch weitere Informationen ableiten. Beispielsweise kann aus dem Verlauf von Positionsbestimmungen die Bewegungsgeschwindigkeit eines Benutzers berechnet werden. Ähnlich können auch aus der Verknüpfung unterschiedlicher Low-Level-Kontextinformationen weitere Informationen abgeleitet werden. Das ergibt sich aus den *Abhängigkeiten und Wechselbeziehungen* unterschiedlicher Informationen. Beispielsweise kann anhand der Benutzerposition und den Geräuschinformationen, die von akustischen Sensoren an demselben Ort registriert wurden, deduziert werden, dass der Benutzer in einer Besprechung teilnimmt. Einerseits können solche Ableitungen zur Bestätigung existierender Informationen verwendet werden. Andererseits können sie zu falschen Informationen führen und dadurch Doppeldeutigkeiten oder undefinierte Zustände entstehen.

4. Übertragung der Kontextinformationen

Wurden die Kontextinformationen erst einmal erfasst, müssen sie zur weiteren Verarbeitung an eine zentrale Instanz übertragen werden. Dazu steht eine Vielzahl an Techniken zur Verfügung. Einen allgemeinen Überblick über die gesamte Kommunikationsinfrastruktur von SFINKS zeigt Abbildung 4.

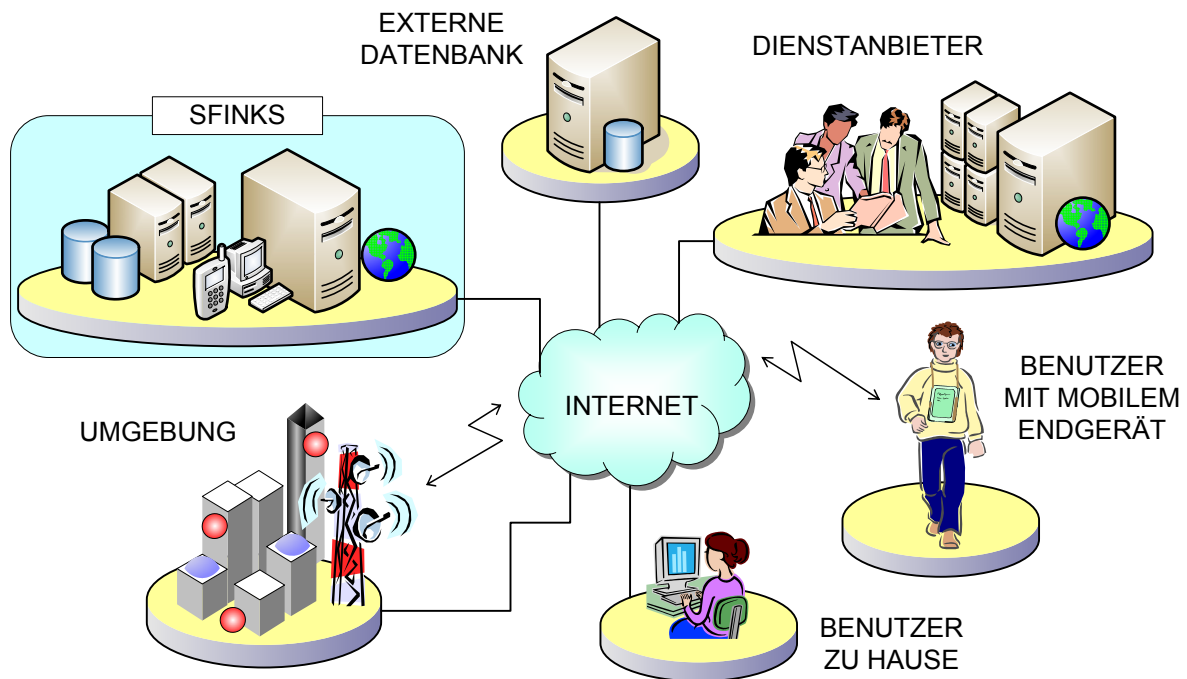


Abbildung 4: Kommunikationsinfrastruktur

Die Verbindung mit dem SFINKS ist durch das Internet möglich. So werden Informationen in externen Datenbanken gesucht und Dienste von Diensteanbieter in Anspruch genommen. Der Benutzer kann einen Zugriff zum SFINKS unabhängig davon haben, ob er unterwegs oder zu Hause ist.

Die Kontextinformationen werden direkt oder vom Endgerät des betreffenden Benutzers an den Server, der diese Informationen auswertet, gesendet. Bei der direkten Variante handelt es sich um Sensoren, die nicht mit dem Benutzer selbst interagieren. In der Regel werden sie stationär verwendet, um Kontextinformationen aus einer bestimmten Umgebung zu sammeln (z. B. Wetterdaten). Diese Variante der Kommunikation ist vor allem dann interessant, wenn viele Benutzer einmalig oder selten auf die Sensoren angewiesen sind. Zur Übertragung der Sensordaten muss im einfachsten Fall das durch die Sensoren vorgegebene Protokoll verwendet werden.

Bei Nutzung der herkömmlichen Netztechniken ist es ähnlich. Entweder ist das Sensorsystem mit Sendern und Empfängern ausgestattet, die eine standardisierte Verbindung zum Server

unterstützen oder die Sensoren müssen solche Standards selber unterstützen, damit eine direkte Kommunikation möglich wird. Die Netztechniken zur Kommunikation mit dem Server entsprechen dann denen, die für die Kommunikation zwischen Endgerät und Server vorgesehen sind (beispielsweise WLAN). Andererseits haben, begründet durch die feste Position der Sensoren, Vorgänge wie Roaming oder Handover keinen Einfluss auf die Kommunikation.

Werden dagegen die erfassten Kontextdaten mit Hilfe des Endgeräts an den Server übertragen, sind auch solche Vorgänge wie vertikaler und horizontaler Handover sowie Roaming zu berücksichtigen. Da die einzelnen Netze nicht überall verfügbar sind, sollte zur Aufrechterhaltung der Kommunikation bzw. Datenübertragung ein Wechsel in andere Netze möglich sein. Dazu sind entsprechende Regeln, z. B. wie und wann diese Netzwechsel zu erfolgen haben, aufzustellen. Die Auswahl könnte hierbei z. B. in Abhängigkeit der zu erwartenden Übertragungskosten erfolgen. Unabhängig davon, welche Kriterien zur Auswahl des Netzes festgelegt werden, sollte der Benutzer von einem Netzwechsel nichts bemerken, d.h. er sollte während der Nutzung des Systems nicht beeinträchtigt werden.

Als Übertragungstechniken lassen sich prinzipiell alle Varianten wie WLAN, GSM, UMTS, GPRS sowie auch das analoge Telefonnetz und ISDN nutzen. Die Bitraten, die von den einzelnen Netztechniken unterstützt werden, unterscheiden sich sehr stark und reichen von 9,6 kbit/s (GSM) bis 600 Mbit/s (IEEE 802.11n). Demnach muss je nach Anwendungsfall entschieden werden, welche Techniken tatsächlich geeignet sind, da sich die benötigten Bitraten je nach Art und Menge der Kontextinformationen stark unterscheiden können. Neben der Leistung spielen bei der Wahl der Technik in der Praxis häufig auch die Kosten eine wichtige Rolle.

Ein kontextsensitives System wird als Backbone in der Regel ein Festnetz nutzen, sodass dort dementsprechend eine Vielzahl standardisierter Protokolle (z. B. IP, TCP ...) genutzt werden kann und sollte, um von Anfang an Interoperabilitätsproblemen aus dem Wege zu gehen. Die Kontextinformationen selber sollten dann mit Hilfe einer allgemeingültigen Beschreibungssprache übertragen werden. Nach [\[LDS05\]](#) würde sich dazu beispielsweise XML sehr gut anbieten.

Auf die Übertragung der angeforderten Dienste wird an dieser Stelle nicht explizit eingegangen, da man hier eigentlich die gleichen Probleme, wie weiter oben beschrieben, zu lösen hat. Die Übertragungsrichtung ist zwar umgekehrt und anstatt der Kontextinformationen werden jetzt Anwendungsdaten übertragen. Das hat jedoch keine Auswirkungen auf die verwendeten Protokolle. Zu beachten wäre aber die verfügbare Bitrate, da für die Nutzung bestimmter Dienste (z. B. Streamingdienste) Mindestbitraten, konstante Verzögerungszeiten usw. benötigt werden. Unter diesen Umständen müssen diese Kriterien auch unbedingt bei einem Netzwechsel (Handover, Roaming) berücksichtigt werden.

5. Auswerten der Kontextinformationen

Dieser Abschnitt befasst sich mit der Frage, wie die übertragenen Kontextinformationen ausgewertet werden können. Innerhalb der vorgestellten Architektur (siehe Abbildung 1) werden die Informationen erfasst, an den Server gesendet und dort mit einer Datenbank abgeglichen.

5.1 Datenbankstruktur zur Speicherung von Kontextinformationen

Für einen effektiven Abgleich muss die Datenbank entsprechend strukturiert sein. Hierfür kann der Kontext beispielsweise in persönlichen und technischen Kontext sowie Umgebungskontext gegliedert werden. Zur Beschreibung eines bestimmten fiktiven Kontextes dienen die in der folgenden Aufzählung enthaltenen Kontextinformationen, wobei deren Umfang je nach Anwendungsfall beliebig erweitert werden kann [DLS05].

Nach [CDM00] gehören dann in die Gruppe des persönlichen Kontextes alle Informationen, die unmittelbar die Person des Benutzers betreffen. Dazu gehören beispielsweise:

- die **Benutzerdaten**: Angaben zur Identität (Name, Vorname, Geburtsdatum und Heimatort, Adresse,...), Interessen (Hobbys, aktuelle Interessen, ...), kognitive Informationen (auf Erfahrung basierende Informationen), Präferenzen (Vorlieben des Benutzers),
- die **Gruppenmitgliedschaft**: Einzelperson oder Zugehörigkeit zu einer Gruppe (Touristengruppe),
- der **Status**: Körperfunktionen (Herzfrequenz, Blutdruck, Körpertemperatur, Zuckerspiegel, ...), emotionaler und psychischer Status, Verhalten bzw. Tätigkeit, Bewegungsgeschwindigkeit (langsam, normal, schnell),
- und **Behinderungen/Einschränkungen**: Art und Grad der Behinderung bzw. Einschränkung des Benutzers.

Zur Gruppe des technischen Kontextes zählen Informationen über das Endgerät sowie über die technischen Möglichkeiten der Kommunikation. Folgende Informationen sollte die Datenbank enthalten:

- das **Gerät**: Typ (PDA, Handy, Laptop,...), Mobilität (stationär, mobil), Display (Größe, Auflösung, ...), Hardwaretasten (Anzahl), Audioausgang (ja/nein), Sprachsynthese (ja/nein), Batteriestatus (Ladezustand, Restlaufdauer),
- und die **Technische Umgebung**: Netz (ISDN, GSM, WLAN, ...).

Die letzte Kontextgruppe umfasst Informationen über die Umgebung des Benutzers. Man spricht hier auch vom Umgebungskontext. Nach [Zip01], [SPZ02] und [CDM00] wird diese Kontextgruppe durch folgende Informationen beschrieben:

- die **Zeit**: Art (Öffnungszeit, verfügbare Zeit, ...), Typ (Datum, Uhrzeit, Tageszeit, Jahreszeit, ...),
- die **Position**: Ort (außer-/innerhalb von Gebäuden), Koordinaten und -art (GPS-, RFID-Daten, Feldstärke),
- die **Orientierung**: Orientierungsart (Richtung, Koordinaten, ...),
- die **Umgebung**: Wetterbedingungen (Temperatur, Bewölkung, Niederschlag, ...), und Sonstiges (zusätzliche wichtige Informationen über die Umgebung).

Ein Beispiel für eine Datenbankstruktur, die die oben aufgezählten Kontextinformationen einschließt, ist in Tabelle 1 grafisch dargestellt.

PERSONLICHER KONTEXT	BENUTZERDATEN	PERSONALANGABEN		ID	0176
				Name	Anna
		INTERESSE		Hobbys	Musik
				aktuelle Interessen	Einkaufen
	GRUPPEMITGLIEDSCHAFT	ZUGEHÖRIGKEIT		Einzelperson	x
			Gruppe	---	
	STATUS	KÖRPERFUNKTIONEN		Herzfrequenz (Schläge/Minute)	70
				Blutdruck (mmHg)	120/80
				Körpertemperatur (°C)	36,2
				Zuckerspiegel (mg/dl)	---
BEWEGUNGS- GESCHWINDIGKEIT		Langsam	---		
		Normal	x		
		Schnell	---		
BEHINDERUNGEN/ EINSCHRÄNKUNGEN			Art	Sehbehinderung	
			Grad (%)	50	
TECHNISCHER KONTEXT	GERÄT	TYP		Pocket PC	
		MOBILITÄT		Stationär	---
				Mobil	x
		DISPLAY		Auflösung (dpi)	240x320
				Farbe	Monochrom
		HARDWARETASTEN		Anzahl	4
				Notruf	x
		AUDIOAUSGANG		x	
	SPRACHSYNTHESE		x		
	BATTERIESTATUS (%)		Ladezustand (%)	30	
Restlaufdauer (Minuten)			35		
TECHNISCHE UMWELT	NETZ		ISDN	---	
			GSM/ GPRS	x	
			WLAN	---	
UMGEBUNGSKONTEXT	ZEIT	ART	verfügbare Zeit (Stunden)	2,5	
		TYP	Datum	15.12.2004	
			Uhrzeit	08:30:46	
	POSITION	ORT		außerhalb von Gebäuden	x
				innerhalb von Gebäuden	---
		ART		GPS	10°56'22''öL 50°40'57''nB
				GSM	---
				RFID-Daten	---
	ORIENTIERUNG	ORIENTIERUNGSART		Richtung	Nord
				Koordinaten	---
UMGEBUNG	WETTERBEDINGUNGEN		Temperatur (°C)	10	
			Niederschlag	Regen	

Tabelle 1: Beispiel für einen Datenbankeintrag (x: ja/vorhanden, -: nein/nicht vorhanden)

5.2 Ordnungsprozess

Die oben beschriebenen Kontextinformationen werden an den Server gesendet. Bevor sie in der Datenbank gespeichert werden, laufen sie durch einen so genannten **Ordnungsprozess**. In diesem Prozess werden die Daten in den entsprechenden Datenbankfeldern (Benutzer, Technik, Umgebung) gruppiert, sodass ein Zugriff auf den Kontext des Benutzers möglich wird [LDS05]. Abbildung 5 zeigt, wie die Kontextinformationen vom Ordnungsprozess verarbeitet werden.

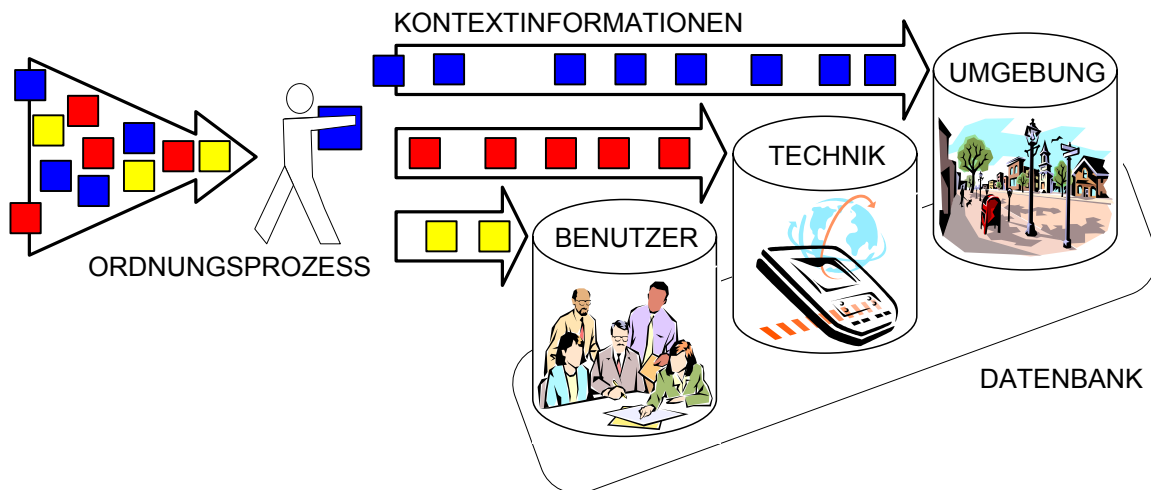


Abbildung 5: Ordnungsprozess

Problem dabei ist, dass für die Datenbank eines kontextsensitiven Systems keine feste Struktur definiert werden kann. Das liegt an den Kontextinformationen (siehe Abschnitt „Eigenschaften von Kontextinformationen“). Es kann nicht festgelegt werden, welche Informationen gebraucht werden. Das kann nur geschätzt werden, weil das mehr von den Anwendungen abhängt. Deshalb wird am Anfang eine beispielhafte Datenbankstruktur angenommen, die mit der Zeit weiter ausgebaut wird. Diese Datenbankstruktur spielt hier eine wichtige Rolle, weil sie als Grundlage für ein Benutzerprofil, das zur Anpassung der Anwendung und der Informationsverarbeitung verwendet wird, dient.

5.3 Profil des Benutzers

Aufgrund der erfassten Informationen kann nun ein Benutzerprofil erstellt werden. Dieses dient zur Schaffung eines Gesamtbildes des Benutzerkontextes. Um das besser zu verstehen, wird ein Beispiel aus dem Leben genommen. Was muss gemacht werden, wenn eine Person gefunden werden soll? Zuerst werden alle möglichen Informationen, die diese Person direkt sowie indirekt betreffen, erfasst. Dann werden sie an einer Stelle gesammelt und gespeichert. Anhand dieser Informationen wird für das Gesamtbild ein allgemeines Profil dieser Person erstellt. Dann werden weitere Informationen gewonnen und analysiert, um das Profil immer genauer zu entwi-

ckeln und schließlich das Personenprofil zu erreichen. Aufgrund des Personenprofils kann mit einer gewissen Wahrscheinlichkeit vorausgesehen werden, was diese Person brauchen, machen oder denken wird, sowie wo sie sich befinden kann usw. Folglich kann diese Person mit großer Wahrscheinlichkeit gefunden werden. So eine Konzeption wird auch in dieser Arbeit verwendet. Nun wird der Benutzer nicht gesucht, sondern aufgrund seines Profils werden zuerst Informationen und Dienste für ihn gesucht und dann angepasst. Der weitere Teil des Abschnittes wird sich mit dieser Aufgabe beschäftigen.

Um ein Benutzerprofil zu bekommen, müssen zuerst Informationen über einen Benutzer zusammengefasst werden. Dazu ist eine Benutzermodellierung, wie in [WOH03] beschrieben, notwendig. Abbildung 6 zeigt einen Überblick, wie das Benutzerprofil erstellt wird.

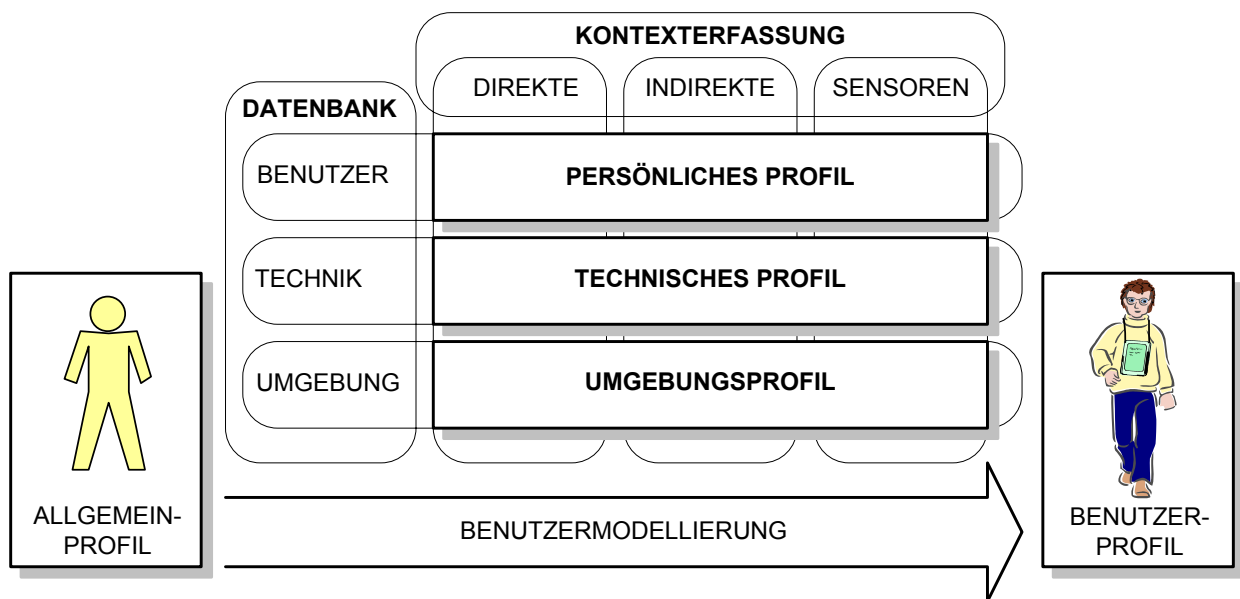


Abbildung 6: Erstellung des Benutzerprofils

Die **Benutzermodellierung** bezeichnet die Erfassung der Informationen über den Benutzer und seinen Kontext sowie die Bestimmung, was sich im Zentrum seines Interesses befindet [PLM01]. Informationen über den Benutzerkontext befinden sich in der Datenbank. Um festzustellen, was ihn interessieren kann, können neben den direkt vom Benutzer erhaltenen Aussagen bezüglich seiner Interessen und Präferenzen auch eine History-Funktionalität und seine Positionsbestimmung genutzt werden.

Bei der **History-Funktionalität** geht es um die Analyse der in der Datenbank schon seit einiger Zeit gespeicherten Informationen. Das kann eine Erkennung wiederholter Anfragen oder das Verhalten eines Benutzers sein. Zum Beispiel, wenn der Benutzer oft nach demselben fragt oder regelmäßig um eine bestimmte Zeit dasselbe macht. Daraus kann sich mit einer gewissen Wahrscheinlichkeit ergeben, wofür er sich interessiert, oder was er beabsichtigt. Dadurch kann

das System kontextabhängig entscheiden, welche Informationen und Dienste für den Benutzer interessant sein könnten und sie ihm zur Verfügung stellen.

Die aktuelle **Positionsbestimmung des Benutzers** kann zum Beobachten seines Verhaltens, zur Voraussicht seiner Bedürfnisse, zum Spezifizieren seiner Wünsche oder als Hinweis zu Sachen, die für ihn interessant sein können, verwendet werden.

Für eine optimale Ausgangsposition zur Bestimmung, was sich im Zentrum des Benutzerinteresses befindet, kann je nach Möglichkeit zwischen den Methoden gewählt oder können diese verknüpft werden. Ein Beispiel aus [PLM01] verdeutlicht das. Man kann beispielsweise davon ausgehen, dass ein Benutzer, der sich innerhalb einer Region für alte Kirchen interessiert, auch Interesse an anderen sich in der Nähe befindlichen Altbauten oder historischen Gebäuden hat.

Aufgrund dieser Informationen kann nun ein **Benutzermodell** erstellt werden. Es hilft dabei, Informationen zusammenzuführen und sie als ein allgemeines Profil des Benutzers nutzen zu können. Das **allgemeine Profil** umfasst allgemeine Informationen über einen Benutzer, die bei kontextsensitiven Anwendungen meistens verwendet werden. Das Profil dient als Grundlage, um daraus ein Benutzerprofil für einen bestimmten Benutzer anfertigen zu können. Deshalb wird es ständig durch die Benutzermodellierung um neue Informationen ergänzt, sodass schließlich das Benutzerprofil entsteht.

Das **Benutzerprofil** beschreibt die Charakteristik des Benutzers. Es hilft nicht nur ein Gesamtbild des Benutzers und seines Kontextes zu schaffen, sondern auch detaillierte nur diesen Benutzer betreffende Informationen zu bekommen. Das Profil enthält Informationen, die es ermöglichen einen bestimmten Benutzer zu beschreiben (Interessen, Präferenzen sowie Möglichkeiten usw.). Das sind Informationen über seinen Kontext wie die Umgebung, in der er sich gerade befindet, sowie das Endgerät, das er gerade verwendet usw. Deswegen ist auch das Benutzerprofil in ein **persönliches**, ein **technisches** und ein **Umgebungsprofil** eingeteilt. Mit jedem Teil des Profils kann ein bestimmter Teil des Benutzerkontextes beschrieben werden.

Wie sich das Benutzerprofil in der Datenbank widerspiegelt, wird in Abbildung 7 gezeigt. Dort sieht man, dass das Benutzerprofil eigentlich keine richtigen Informationen enthält. Es enthält eine Struktur von Informationsarten, die für das Profil relevant sind. Als Grundlage für diese Struktur wird die Datenbankstruktur genommen. Sie wird nur durch die Benutzermodellierung von einem allgemeinen Profil in ein Benutzerprofil umgewandelt. Dieser Vorgang wird hintereinander für das persönliche, technische und Umgebungsprofil durchgeführt. Die Informationsarten zeigen auf die nutzerbezogenen Daten, die in der Datenbank gespeichert wurden. Bei Bedarf

werden sie daraus abgelesen. Dadurch bleiben sie immer aktuell. Es wird damit auch Informationsredundanz vermieden.

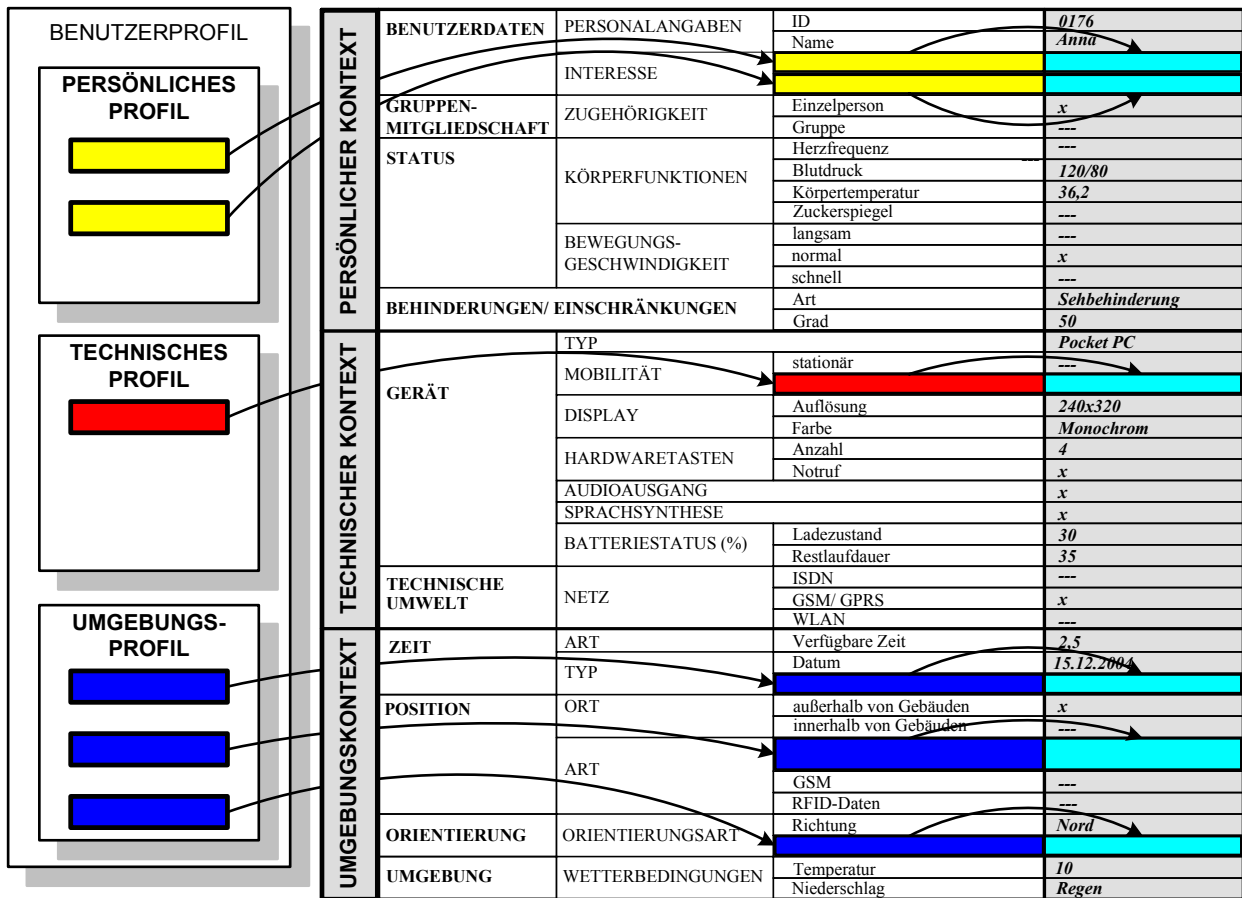


Abbildung 7: Profil des Benutzers

Mit einem Beispiel kann der Vorgang besser dargestellt werden. Um ein technisches Profil für das Endgerät des Benutzers zu erhalten, werden zuerst entsprechende Teile der Datenbankstruktur zusammengeführt. So entsteht ein allgemeines Profil des Endgerätes. Mit der Benutzermodellierung werden die Daten entsprechend ergänzt. Felder, für welche es keine Informationen in der Datenbank gibt, wie beispielsweise stationär, Notruf und Audioausgang (Abb. 7), werden hierbei nicht beachtet. Da ein Benutzerprofil für die adaptierenden Prozesse verwendet wird, werden auch nicht alle Informationsarten betrachtet. Dazu gehört beispielsweise der Typ des Endgerätes. Diese Information kann aber trotzdem verwendet werden, um mehr Informationen über das Endgerät an die Prozesse zu liefern. Für die Anpassung an den Benutzerkontext kann wiederum der Typ selbst nicht verwendet werden. Er kann nicht an den Benutzerkontext angepasst werden. Hinzu kommen aber Informationen, die von der History-Funktionalität erkannt wurden. Daraus folgt, dass der Benutzer in dem hier beschriebenen Beispiel oft Informationen in einer grafischen Form wie Bilder sehen wollte. Dieser zusätzliche Hinweis wird in der Daten-

bank unter den Präferenzen gespeichert und das Feld „Präferenzen“ wird in das Profil aufgenommen. Daraus ergibt sich das technische Profil für das Endgerät des Benutzers

Letztlich wird das Benutzerprofil bei Filterprozessen zur Suche von Informationen und Diensten verwendet. Neue Kontextinformationen führen zu einem ständigen Adaptionsprozess in Bezug auf den vom System erkannten Benutzerkontext.

6. Verarbeiten und Anwenden der Kontextinformationen

Nachdem die Kontextinformationen erfasst, übertragen und in einer Datenbank verarbeitet worden sind, ist es möglich, sie zur Anpassung von Diensten und Informationen zu nutzen. Ist dem System also der Kontext des Benutzers bekannt, sind verschiedene Ausgangsszenarien für den Abruf kontextsensitiver Dienste denkbar. So kann der Benutzer eine direkte Anfrage stellen und danach wird sein aktuell bekannter Kontext bei der Suche eines geeigneten Dienstes genutzt. Andererseits können dem Benutzer auch automatisch bestimmte Dienste oder Informationen anhand der aktuellen Kontextinformationen bereitgestellt werden.

6.1 Beobachtungsprozess

Um dem Benutzer automatisch entsprechende Dienste anzubieten, werden dazu die Interaktionen zwischen Benutzer und System beobachtet sowie entsprechende Ereignisse innerhalb der Datenbank detektiert. An dieser Stelle spricht man auch vom *Beobachtungsprozess*. Dieser Prozess arbeitet unabhängig vom Benutzer. Er unterteilt sich in drei Unterprozesse Kontrolle, Analyse und Reaktion. Die Abbildung 8 stellt den Beobachtungsprozess, in dem die drei Unterprozesse enthalten sind, dar.

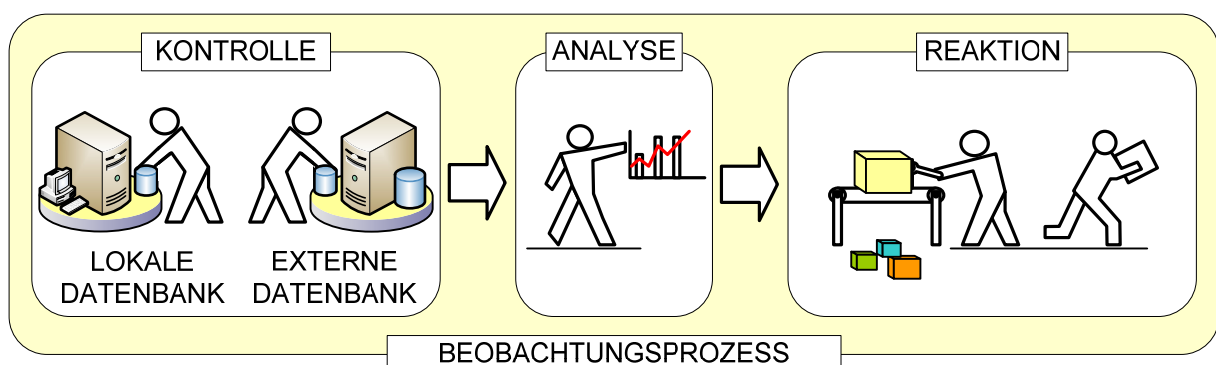


Abbildung 8: Beobachtungsprozess

Der *Kontrollprozess* beobachtet die Datenbank. Sofern dort die Änderung eines Wertes festgestellt wird, startet der *Analyseprozess*. Dieser Prozess wiederum überprüft, ob der neue Wert den (festgelegten) zulässigen Wertebereich überschreitet. Nur wenn dieser Bereich überschritten wird, startet der *Reaktionsprozess*. Das heißt, das System reagiert auf die Änderung(en)

des Benutzerkontextes und startet ein Suchen nach entsprechenden Informationen und aktiviert dadurch einen anderen Prozess oder ruft die dafür vorgesehenen Dienste auf.

Diese Vorgehensweise soll mit den folgenden Beispielen kurz verdeutlicht werden. So könnte der Kontrollprozess melden, dass sich der Zuckerspiegel des Benutzers verändert hat. Der Analyseprozess prüft den Neuwert und stellt eine beträchtliche Verschlechterung des Wertes fest. Daraufhin setzt der Reaktionsprozess beispielsweise einen Notruf ab. Ebenso wäre es mit der Änderung der Benutzerposition. Der Kontrollprozess registriert den neuen Wert. Der Analyseprozess überprüft, ob diese Änderung bedeutend ist, und abhängig davon wird der Reaktionsprozess in Gang gebracht. Das System kann anhand der neuen Benutzerposition und seiner Interessen prüfen, ob dort etwas für ihn interessant sein könnte. Das führt wiederum dazu, dass das Suchen nach entsprechenden Informationen gestartet wird.

Der Beobachtungsprozess ist nicht nur auf die Datenbank allein beschränkt. Auch Daten von entfernten Quellen können mit einbezogen und auf Änderungen überprüft werden. Beispielsweise können die Öffnungs- und Schließzeiten eines Museums beobachtet werden. Möchte der Benutzer nun dieses Museum besuchen, es aber aus „höheren“ Gründen kurzfristig geschlossen wurde, so kann er sofort darüber informiert werden. Daraus ergibt sich, dass es nicht ausreicht, die Öffnungs- und Schließzeiten des Museums in der Datenbank zu speichern. Gerade bei sich ständig ändernden Informationen (z. B. Wetter = zeit- und ortsabhängig) macht es wenig Sinn, diese lokal abzuspeichern. Solche Informationen sollten vielmehr immer wieder aktualisiert werden. Nur so können dem Benutzer aktuelle Informationen und Dienste geliefert werden. Dabei ist auch wichtig, dass sie ihm diskret gegeben werden. Dem Benutzer sollte nur signalisiert werden, dass ihm mehr Informationen zur Verfügung stehen. Im anderen Fall würde er bei seinen Tätigkeiten ständig gestört werden. Damit könnte sich eventuell eine Unzufriedenheit des Benutzers gegenüber dem System einstellen.

6.2 Suchprozess

Für das Suchen von Diensten und Informationen in den lokalen Ressourcen sowie im Netz ist dann der so genannte **Suchprozess** verantwortlich. Informationen und Dienste werden auf Grund einer Anfrage gesucht. Die Suchanfrage kann aus unterschiedlichen Informationen bestehen. Am Anfang kann das nur ein Stichwort sein. Wenn das Suchen nicht erfolgreich ist, dann kann ein Synonym des Stichwortes verwendet werden. Um die geeignete Information zu finden, wird das Stichwort oft mit anderen Informationen wie z. B. mit der Lokalisierung des Benutzers, der Zeit usw. verbunden. Die Hilfsinformationen werden je nach Bedarf unterschiedlich gewählt.

Der Suchprozess erfolgt normalerweise sehr allgemein und liefert eine Anzahl von Ergebnissen. Problem dabei ist, dass die Hälfte oder mehr davon unpassend ist. Die Ergebnisse stimmen überhaupt nicht mit dem überein, was gerade gefunden werden sollte. So funktionieren beispielsweise Suchmaschinen im Internet. Man gibt ein Stichwort und als Ergebnis bekommt man normalerweise Tausende oder noch mehr Informationen. Häufig passen die Suchergebnisse überhaupt nicht zu dem Stichwort, weil das Wort auf unterschiedliche Art und Weise interpretiert wurde. Wenn aber letztlich passende Ergebnisse gefunden werden, dann sind sie manchmal nicht mehr aktuell oder können vom Benutzer nicht dargestellt werden. Ein Grund dafür kann ein mobiles Endgerät des Benutzers sein. Das Gerät kann für die gefundenen Informationen z. B. zu alt sein oder ein zu kleines Display haben oder noch über ungenügende Leistung verfügen.

6.3 Adaptierende Prozesse

In jeder Situation, die oben dargestellt wurde, reicht es nicht dem Benutzer die gefundenen Informationen zu geben. Sie müssen zuerst entsprechend verarbeitet werden. Deswegen werden an dieser Stelle die so genannten *adaptierenden Prozesse* gebraucht. Erst sie ermitteln die für den aktuellen Benutzerkontext passenden Dienste. Abbildung 9 zeigt, wie die Informationen und Dienste von den adaptierenden Prozessen verarbeitet werden.

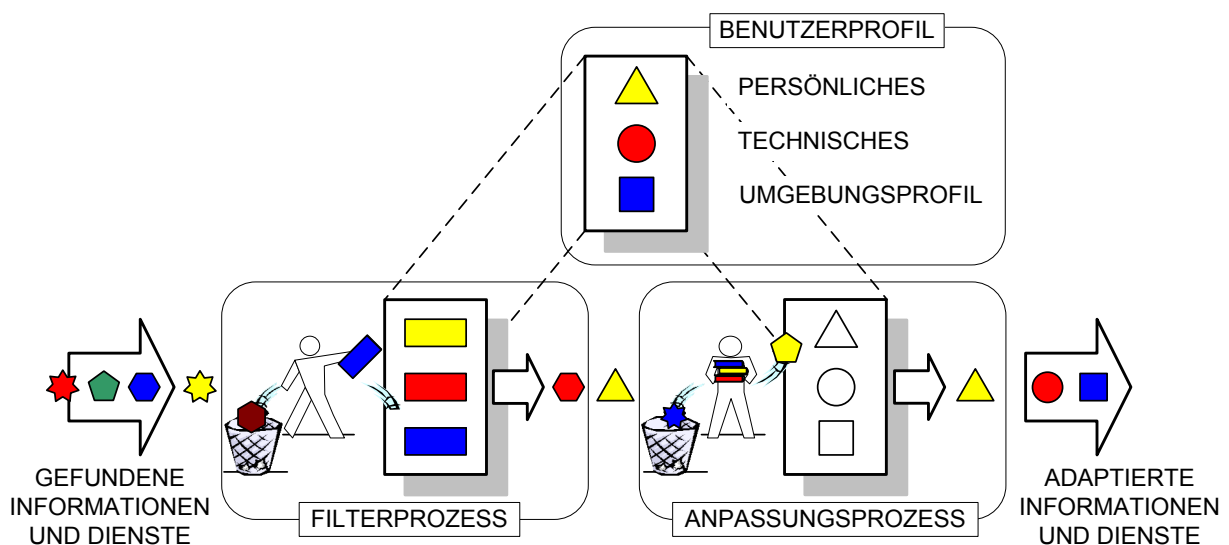


Abbildung 9: Adaptierende Prozesse

Wie auf dem Bild zu sehen ist, spielt ein Benutzerprofil an dieser Stelle eine sehr wichtige Rolle. Das Benutzerprofil besteht aus dem persönlichen, technischen und aus dem Umgebungsprofil (siehe Abschnitt „Datenbankstruktur zur Speicherung von Kontextinformationen“). Damit kann der Benutzerkontext, der am den persönlichen, technischen und auf den Umgebungs-kontext aufgeteilt wird, beschrieben werden.

Die durch den Suchprozess gefundenen Informationen und Dienste werden zuerst gefiltert, um die überflüssigen Informationen zu entfernen. Man spricht an dieser Stelle vom **Filterprozess**. Das Benutzerprofil liefert nun die Basis für diesen Prozess. Das heißt, der Filterprozess ist dafür verantwortlich, dass nur der Dienst favorisiert wird, der auf den aktuellen Kontext am besten zugeschnitten ist. Dieser Abgleich erfolgt über das Benutzerprofil.

Des Weiteren besteht die Möglichkeit, gefundene Dienste zu personalisieren, sofern diese das unterstützen. Die **Personalisierung** der Dienste bezeichnet die individuelle Anpassung an die Interessen und Möglichkeiten des Benutzers sowie an seinen Kontext.

Die Personalisierung erfolgt über den **Anpassungsprozess**. Hierbei wird der entsprechende Dienst mit Hilfe des Benutzerprofils möglichst gut auf den Kontext des Benutzers adaptiert. Es wird also versucht, den Dienst dem persönlichen, dem technischen sowie dem Umgebungsprofil und damit dem Benutzerkontext anzupassen. Die Anpassung, die während der Laufzeit der Anwendung stattfindet, kann beispielsweise an den Benutzer durch die Berücksichtigung seiner Interessen und Präferenzen, an sein Endgerät durch die Berücksichtigung der Gerätparameter wie der Art des Endgerätes und der Größe des Displays, sowie an die Umgebung durch die Beachtung seiner Position, Bewegungsgeschwindigkeit und Wetterbedingungen erfolgen. Dadurch wird der Anpassungsprozess mit Hilfe der Änderung von Darstellung, Inhalten, Datenfluss, Datenformat usw. realisiert. Damit kann z. B. die Präsentation von Informationen abhängig von der Bewegungsgeschwindigkeit des Benutzers erfolgen. Bei Eintritt der Dunkelheit kann der Kontrast für die Darstellung automatisch erhöht werden, damit das Lesen im Dunkeln leichter fällt. Oder dem Benutzer werden abhängig von seinen Präferenzen keine Texte sondern nur Bilder präsentiert.

6.4 Priorisierungsprozess

Neben den oben vorgestellten Möglichkeiten der kontextsensitiven Diensterbringung soll auch noch kurz diskutiert werden, wie sichergestellt werden kann, dass auch bei einem Systemausfall der Benutzer zumindest bis dahin mit den wichtigsten Daten versorgt wird. Dafür ist der **Priorisierungsprozess** verantwortlich. Hierbei wird festgelegt, welche Daten und Informationen in welcher Reihenfolge an den Benutzer gesandt werden. Wichtige oder grundlegende Informationen sollten zuerst übertragen werden, danach können Details oder spezifische und ergänzende Daten gesandt werden.

6.5 Vorgehen beim Verarbeiten und Anwenden der Kontextinformationen

Da alle benötigten Prozesse schon genauer betrachtet wurden, kann jetzt die Vorgehensweise beim Verarbeiten und Anwenden der Kontextinformationen innerhalb vom SFINKS dargestellt werden. In Abbildung 10 wird sie mit Hilfe eines Ablaufdiagramms gezeigt.

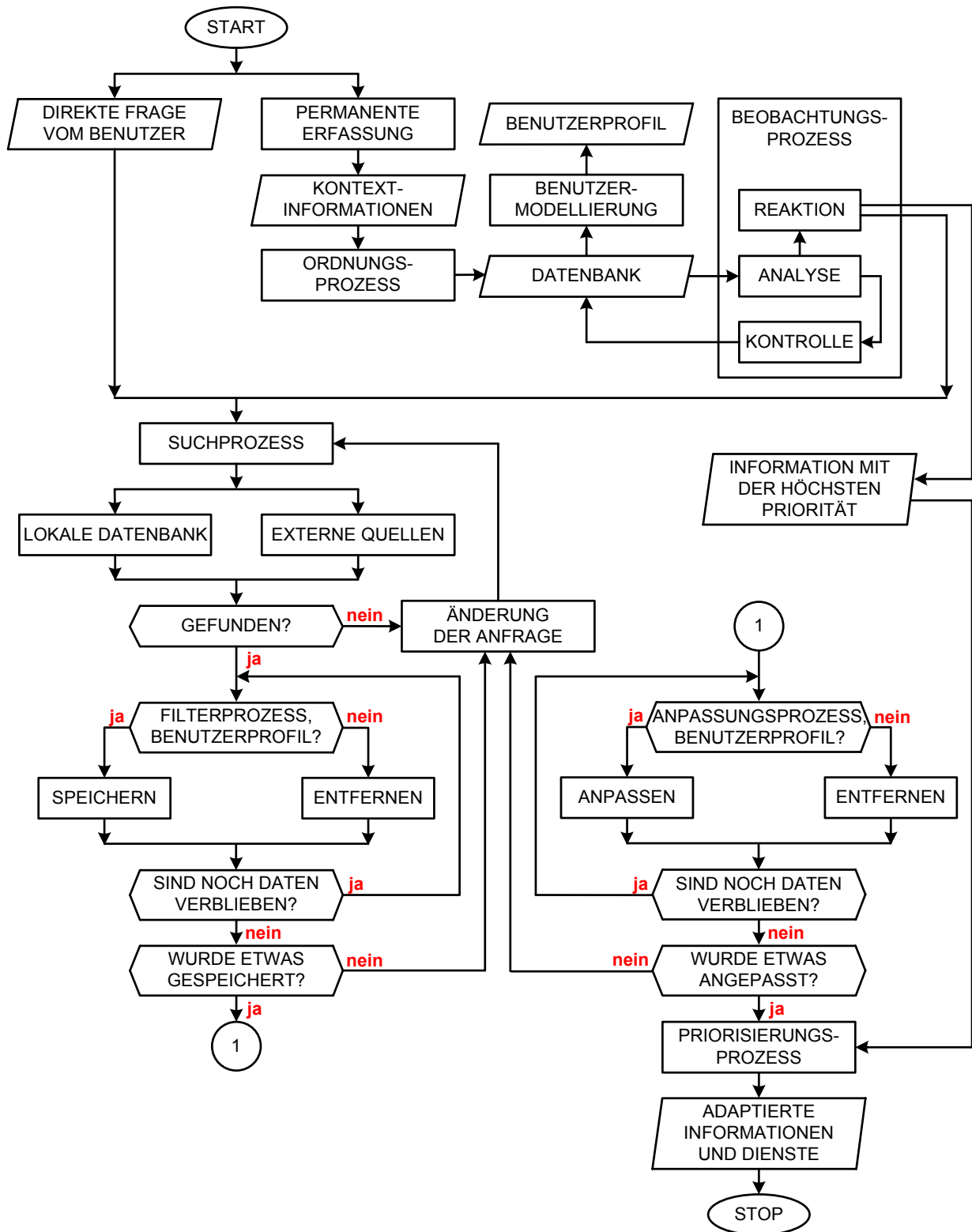


Abbildung 10: Vorgehensweise bei Verarbeiten und Anwenden der Kontextinformationen

Am Eingang des Systems befinden sich die Kontextinformationen, die permanent erfasst und an den Server gesendet werden. Sie werden zuerst vom Ordnungsprozess entsprechend verarbeitet und dann in der Datenbank gespeichert. Aufgrund dieser Informationen wird ein Benutzerprofil von der Benutzermodellierung erstellt. Das Benutzerprofil wird später der Datenbank angeglichen. Die Informationen innerhalb der Datenbank werden ständig vom Beobachtungsprozess überwacht. Wenn sich ein Wert ändert, dann wird er überprüft. Sofern der neue Wert den zugelassenen Wertebereich überschreitet, werden (im kritischen Fall) zuerst Informationen und Dienste mit der höchsten Priorität aufgerufen. Anderenfalls wird mit dem Suchen nach sachgemäßen Informationen begonnen.

Der Suchprozess kann ebenso auf Wunsch des Benutzers aktiviert werden. Es werden Informationen und Dienste in den lokalen sowie in den externen Ressourcen gesucht. Falls keine erforderlichen Daten gefunden werden, wird die Suchanfrage geändert und das Suchen beginnt von vorn. Die gefundenen Daten werden mit dem Benutzerprofil, das in dem Fall als ein Filter verwendet wird, gefiltert. Informationen und Dienste, die auf den aktuellen Kontext zugeschnitten sind, werden vom Filterprozess gespeichert. Alle anderen werden entfernt. Nachdem alle gefundenen Daten geordnet wurden, wird geprüft, ob etwas gespeichert werden konnte. Es kann geschehen, dass alle gefundenen Daten gelöscht wurden, weil sie nicht auf das Benutzerprofil zugeschnitten waren. In einer solchen Situation muss die Suchanfrage geändert werden, sodass die passenden Informationen und Dienste besser getroffen werden können. Wenn aber etwas geblieben ist, dann wird das dem Benutzerkontext basierend auf das Benutzerprofil angepasst. Es kann aber sein, dass nicht alle Informationen und Dienste auf den Kontext adaptiert werden können. Beispielsweise soll einem Benutzer eine Audiodatei geliefert werden. Das Benutzergerät ist entsprechend ausgestattet. Aber wegen einer veralteten Software können nur bestimmte Datenformate gelesen werden. Wenn die gefundene Audiodatei nicht entsprechend umgewandelt werden kann, braucht sie dem Benutzer nicht übermittelt werden. Falls am Ende keine angepassten Daten bleiben, muss die Suchanfrage umgestaltet und der Suchprozess nochmals gestartet werden. Daraus ergibt sich, dass Daten, die bezüglich des Benutzerprofils vom Anpassungsprozess adaptiert werden können, gespeichert und alle anderen entfernt werden. Die angepassten Daten werden danach vom Priorisierungsprozess in einer Reihenfolge angeordnet, sodass die wichtigen oder grundlegenden Informationen als erste übertragen werden. Am Ausgang des Systems werden solche kontextsensitiven Informationen und Dienste angeboten, die auf den Benutzer und sein Kontext angepasst sind.

7. Zusammenfassung der Architekturdarstellung

In diesem Bericht wurde die Architektur vom SFINKS für die Verarbeitung von Kontextinformationen beschrieben. Ziel dieser Architektur ist, Kontextinformationen eines Benutzers so zu verarbeiten, dass dafür geeignete Informationen und Dienste optimal auf die Bedürfnisse des mobilen Benutzers angepasst werden können.

Die Architektur wurde zuerst in einer kurzen Einleitung vorgestellt. Die darauf folgenden Abschnitte gingen auf die grundlegenden Aufgaben, die von einem kontextsensitiven System erfüllt werden sollten, ein.

Es wurde im Abschnitt „**Erfassen und Sammeln der Kontextinformationen**“ diskutiert, wie die zur Beschreibung nötigen Kontextinformationen erfasst und gesammelt werden können. Es wurde gezeigt, dass es vorteilhaft ist, wenn die Kontexterfassung in Echtzeit durch mehreren Informationsquellen erfolgt. Dann wurde auf die unterschiedlichen Arten der Kontexterfassung wie direkt, indirekt und mit Hilfe von Sensoren hingewiesen. Aufgrund der Vor- und Nachteile konnte festgestellt werden, dass eine optimale Kontexterfassung erfolgt, wenn je nach Verwendungszweck zwischen den drei Methoden gewählt wird oder diese kombiniert werden. Weil Kontextinformationen nach Art und Qualität sehr unterschiedlich sind, wurden sie allgemein durch die zeitlichen Veränderungen des Kontextes, alternative Repräsentationsformen sowie Abhängigkeiten und Wechselbeziehungen beschrieben.

Weiterhin wurde im Abschnitt „**Übertragung der Kontextinformationen**“ eine allgemeine Kommunikationsinfrastruktur für SFINKS dargestellt. Es wurde auch dargelegt, welche Möglichkeiten zur Übertragung der Kontextinformationen bestehen und was hierbei zu berücksichtigen ist.

Im Abschnitt „**Auswerten der Kontextinformationen**“ wurde besprochen, wie die übertragenen Kontextinformationen ausgewertet werden können. Dazu wurde eine Datenbankstruktur, die den persönlichen, den technischen und den Umgebungskontext enthält, beschrieben. Die an den Server gesendeten Kontextinformationen wurden zuerst vom Ordnungsprozess verarbeitet und danach in der Datenbank gespeichert. Die Datenbankstruktur und dort gespeicherte Informationen wurden als Grundlage für ein allgemeines Profil benötigt. Das Profil wurde bezüglich eines Benutzers von der Benutzermodellierung verarbeitet, um ein Benutzerprofil zu generieren. Das Benutzerprofil, das in ein persönliches, ein technisches, und ein Umgebungsprofil eingeteilt wurde, gibt ein zeitgemäßes Gesamtbild über den Benutzerkontext.

Im letzten Abschnitt „**Verarbeiten und Anwenden der Kontextinformationen**“ wurde dann darüber diskutiert, wie die Anwendungen mit Hilfe des Kontextes auf die Bedürfnisse des

Benutzers abgestimmt werden können. Als Ausgangsszenarien für den Abruf kontextsensitiver Dienste wurden direkte Anfragen vom Benutzer, Interaktionen zwischen Benutzer und System sowie Reaktionen auf Änderungen des Benutzerkontextes dargestellt. Die Änderungen des Kontextes wurden vom Beobachtungsprozess in den lokalen und externen Datenbanken verfolgt. Als Nächstes wurde der Suchprozess, der für das Suchen von Informationen und Diensten im Internet verantwortlich war, betrachtet. Danach wurden adaptierende Prozesse dargestellt. Sie wurden in Filter- und Anpassungsprozess geteilt. Damit wurden die Daten mit Hilfe des Benutzerprofils gefiltert und angepasst. Daraus ergaben sich Informationen und Dienste, die optimal auf den Benutzerkontext adaptiert wurden. Sie wurden in Abstimmung mit dem Priorisierungsprozess an den Benutzer übermittelt. Am Ende des Abschnittes wurde noch einmal grafisch die ganze Vorgehensweise beim Verarbeiten und Anwenden der Kontextinformationen erläutert.

Literaturverzeichnis

- [CDM00] K. Cheverst, N. Davies, K. Mitchell, A. Friday, C. Efstratiou: „*Developing a context-aware electronic tourist guide: some issues and experiences*”, Proceedings of the 2000 Conference on Human Factors in Computing Systems (CHI-00), N. Y., Thea Turner, Gerd Szwillus, Mary Czerwinski and Paternó Fabio (eds.), ACM Press, pp. 17-24, April, 2000.
<http://www.comp.lancs.ac.uk/~efstrati/research/papers/CHI.pdf>
- [DLS05] M. Debes, A. Lewandowska, J. Seitz: „*Definition and Implementation of Context Information*”, Joint 2nd Workshop on Positioning, Navigation and Communication 2005 (WPNC '05) & 1st Ultra-Wideband Expert Talk 2005 (UET '05), University of Hannover, Hannover, Germany, March, 17th 2005, Hannoversche Beiträge zur Nachrichtentechnik, Shaker Verlag, Band 0.2, S. 63 - 68, ISBN 3-8322-3746-1, Aachen, Germany 2005.
- [HIR02] K. Henricksen, J. Indulska, A. Rakotonirainy: „*Modelling Context Information in Pervasive Computing Systems*”, In Proceedings of the First International Conference on Pervasive Computing, Zurich, Switzerland, August 26-28 2002, Volume 2414 of Lecture Notes in Computer Science., Springer (2002), pp. 167-180, 2002.
<http://citeseer.ist.psu.edu/henricksen02modeling.html>
- [LDS05] A. Lewandowska, M. Debes, J. Seitz: „*An Architecture for Context-Sensitive Telecommunication*”, WEBIST 2005: „1st International Conference on Web Information Systems and Technologies”, Miami, U.S.A, May 26 -28, 2005, INSTICC - Institute for Systems and Technologies of Information, Control and Communication, INSTICC PRESS, pp. 40 - 47, ISBN 972-8865-20-1, Miami, USA 2005.
- [PLM01] S. Poslad, H. Laamanen, R. Malaka, A. Nick, P. Buckle, A. Zipf: „*CRUMPET: creation of user-friendly mobile services personalised for tourism*”, In: Proceedings of: 3G 2001 - Second Int. Conf. on 3G Mobile Communication Technologies. 26-29.03.2001. London. UK, 2001.
<http://www2.geoinform.fh-mainz.de/~zipf/3g-crumpet2001.pdf>
- [SPZ02] B. Schmidt-Belz, S. Poslad, A. Zipf: „*Creation of User-friendly Mobile Tourism Services*”, Proceedings of the workshop of “HCI in Mobile Tourism Services”, 2002.
http://fit-bscw.gmd.de/pub/bscw.cgi/d33335856-2/*/*mTourismHCI-crumpet.pdf
- [WOH03] R. Wasinger, D. Oliver, D. Heckmann, B. Braun, B. Brandherm, C. Stahl: „*Adapting Spoken and Visual Output for a Pedestrian Navigation System, based on given Situational Statements*”, ABIS Workshop on adaptively and user modelling in interactive software systems, pp. 343-346, 2003.
<http://w5.cs.uni-sb.de/~stahl/m3i/ABIS2003.pdf>
- [WSK03] R. Wasinger, C. Stahl, A. Krüger: „*M3I in a Pedestrian Navigation & Exploration System*”, Proc. of the Fourth International Symposium on Human Computer Interaction with Mobile Devices, 2003, pp. 481-485, 2003.
<http://www.dfki.de/~krueger/PDF/mobileHCI03.pdf>
- [Zip01] A. Zipf: „*Interoperable GIS-Infrastruktur für Location-Based Services (LBS) - M-Commerce und GIS im Spannungsfeld zwischen Standardisierung und Forschung*”, In: GIS Geo-Information-Systeme. Zeitschrift für raumbezogene Information und Entscheidungen. 09/2001. 37-43, 2001.
<http://www2.geoinform.fh-mainz.de/~zipf/GIS-LBS2001.zipf-final.pdf>