A. M. Heinecke, H. Paul (Hrsg.): Mensch & Computer 2006: Mensch und Computer im Struktur*Wandel*.
München, Oldenbourg Verlag, 2006, S. 213-222

Adaptive Nutzerschnittstelle für intelligente Wohnumgebungen

Christian Ressel, Jürgen Ziegler

Fraunhofer Institut Mikroelektronische Schaltungen und Systeme,
Universität Duisburg-Essen, Lehrstuhl Interaktive Systeme und Interaktionsdesign

Zusammenfassung

Moderne Geräte des täglichen Lebens verfügen oft über eine Vielzahl von Funktionen. Daher kommt einer effizienten Bedienungsführung solcher Geräte eine entscheidende Bedeutung zu. Die Menüführung einzelner Geräte ist heutzutage in jedem Gerät selbst fest verankert und wird von jedem Hersteller eigenständig festgelegt. Dies führt dazu, dass sich sogar vom Funktionsumfang gleichartige Geräte von der Bedienung her stark unterscheiden können und der Benutzer daher typischerweise mit inkonsistenten Bedienkonzepten konfrontiert ist. Er muss bei jedem neuen Gerät die Logik der jeweiligen Bedienung erst mühsam erlernen, ehe er das Gerät effizient bedienen kann. Noch problematischer wird diese Situation, wenn zukünftig eine sehr große Anzahl vernetzter Geräte und übergeordnete Softwaredienste so genannte integrierte und intelligente Umgebungen bilden. Hier müssen viele Geräte mit einer großen Bandbreite an Funktionen mit Hilfe einer sinnvollen Nutzerführung, zu einem beherrschbaren System verwoben werden. In dieser Arbeit stellen wir ein Konzept für die Erzeugung einer integrierten Bedienoberfläche für solche Umgebungen vor. Die Bedienlogik (Menüführung) und die Darstellung passen sich hierbei auf die Präferenzen des Benutzers an, so dass sich nicht der Mensch an die Umgebung, sondern die Umgebung an den Menschen adaptiert. Unbekannte Umgebungen, mit denen ein Benutzer – als mobiles Wesen – zukünftig häufig interagieren muss, werden hierdurch intuitiv bedienbar. Des Weiteren trennt der vorgestellte Ansatz die Bindung zwischen Funktionalität und zugehörigem Gerät auf, so dass vollkommen neue den Bedürfnissen des Nutzers angepasste Menüführungen entstehen können. Es entsteht dadurch eine bisher nicht erreichte Qualität in der Benutzerführung.

1 Einleitung

Die Frage wie wir in der Zukunft wohnen werden wird seit längerem von unterschiedlichen Seiten, wie den Medien und wissenschaftlichen Organisationen, mit Hilfe von vielfältigen Szenarien beleuchtet. Im Kern der meisten dieser Szenarien steht die Vorstellung einer intelligenten Umgebung, die den Bewohnern zeitraubende oder lästige Arbeiten abnimmt und neue Dienste in den für die Nutzer wesentlichen Bereichen wie Sicherheit, Energieeffizienz oder Gesundheit bietet (Tränkler 2001). Bei einer derartigen intelligenten Umgebung (smart

environment) werden nicht mehr die isolierten Funktionen der beteiligten Einzelgeräte betrachtet, sondern „vielmehr das abgeglichene und sinnvolle Zusammenspiel der Teile in einem gewerkeübergreifenden Gesamtsystem“ (Scherer & Grinewitschus 2002, 153). So soll auch dem Benutzer vermittelt werden, dass er es mit einem Gesamtsystem und nicht mit einer Ansammlung von Einzelgeräten zu tun hat. Dies kann durch die Einführung eines einheitlichen Zugangs zur Bedienung erreicht werden (Ressel 2004), bei dem sich die Funktionen der Einzelgeräte sinnvoll in ein Gesamtbedienungskonzept eingliedern.

Da der Mensch im Zuge wachsender Mobilität zwischen unterschiedlichen Umgebungen wechselt, wird er in Zukunft mit unterschiedlichen vernetzten Umgebungen interagieren müssen. Dabei wird jede Umgebung ihre eigene Auswahl an Geräten, Diensten und eigenen Funktionen besitzen, welche durch unterschiedliche Hersteller und Provider realisiert werden können.

Die „Adaption“ des Menschen an die jeweiligen Bedienschnittstellen (der Mensch passt sich mit dem Durchlaufen einer Lernphase an, ehe er in der Lage ist eine effiziente Bedienung durchzuführen), wie sie noch heutzutage bei modernen Haus- und Entertainmentsystemen üblich ist, ist in einer Welt, die aus mehreren unterschiedlichen vernetzten Umgebungen bestehen soll, nicht vorstellbar. Schnell würde sich der Mensch überfordert fühlen und die neuen Systeme nicht akzeptieren und daher meiden.

Um den Lernaufwand eines Benutzers bei einem Umgebungswechsel möglichst gering zu halten, muss daher gefordert werden, dass sich unterschiedliche Umgebungen mit ihren unterschiedlichsten Geräten möglichst ähnlich präsentieren, so dass der Benutzer mit bekannten (trainierten) Strukturen interagieren kann.

Auch sollte beachtet werden, dass sich Menschen in ihrem Denken und ihrer Vorgehensweise grundlegend unterscheiden können (Norcio & Stanley 1989). Eine Personalisierung der Bedienstrukturen sollte daher möglich sein, so dass z.B. auf dem jeweiligen Nutzer angepasste Menüstrukturen aufgebaut werden können. So könnte eine grafische Benutzerschnittstelle für den einen Benutzer die aufrufbaren Operationen nach den Standorten der zugehörigen Geräte sortiert haben, während ein zweiter Benutzer die Operationen lieber nach von ihm festgelegten Themen sortiert haben möchte. Unbekannte Bedienoperationen für zuvor nicht bekannte Geräte in unterschiedlichen Umgebungen sollten sich automatisch und sinnvoll in die jeweilige Struktur eingliedern.

In dieser Arbeit wird ein Konzept vorgestellt, mit dem eine abstrakte umgebungs- und geräteunabhängige Beschreibung einer personalisierten Bedienstruktur definiert werden kann, die den Benutzer in unterschiedliche Umgebungen begleitet. Den Umgebungen und ihren einzelnen Komponenten wird somit die Möglichkeit gegeben, ihre Funktionen auf ein dem Benutzer bekanntes und bevorzugtes Bedienkonzept zu adaptieren und eine in der Struktur und in der Präsentation personalisierte Benutzerschnittstelle aus den Beschreibungen der verfügbaren Dienste zu generieren. Dem Benutzer bietet dieser Ansatz eine intuitivere Bedienung wechselnder Geräte in unterschiedlichen Umgebungen, da diese in einer ihm vertrauten und von ihm anpassbaren Struktur erfolgt.

Das hier vorgestellte Konzept wird im Rahmen des EU Projekts „Amigo“ (Amigo) entwickelt, welches sich mit der Schaffung einer Middleware für „Ambient Intelligence Environments“ beschäftigt.

2 Verwandte Arbeiten und Grundlage

In der Literatur finden sich viele Arbeiten im Bereich der Benutzerschnittstellenadaptivität. Meist werden unterschiedliche Faktoren betrachtet, auf die sich die Adaptivität bezieht. So existieren Ansätze, die sich auf die Fähigkeiten des Gerätes beziehen, welches für die Interaktion benutzt wird. Dabei wird z.B. die Informationspräsentation den Fähigkeiten (meist der Bildschirmgröße) des Gerätes anpassen (Menkenhaus & Pree 2002). Auch existieren Ansätze, die eine Anpassung einer Applikation anhand des Wissenstands des jeweiligen Benutzers versuchen. Hierzu gehören z.B. wissensadaptive Lernsysteme, wie sie von Granic und Glavinic (Granic & Glavinic 2000) vorgestellt wurden. Seit den 90ern können auch immer mehr kontext-adaptive Systeme beobachtet werden, die nicht nur die Identität (und damit den Wissensstand, Verhaltensweisen, Interessen, Behinderungen) und den aktuellen Standort des Benutzers einbeziehen, sondern auch sämtlich andere Informationen, die den Kontext beeinflussen können in ihre Entscheidung einfließen lassen (Kaltz & Ziegler 2006). Neben den oben genannten Größen können dies z.B. auch der aktuelle Zeitpunkt, Umgebungsinformationen, aktuelle Benutzertätigkeiten (Was macht der Benutzer?), das soziale Umfeld oder die technische Infrastruktur sein (Oppermann 2005).

In dieser Arbeit bezieht sich die Anpassung auf die Erzeugung einer Benutzerschnittstelle, die den Funktionsumfang intelligenter Umgebung auf eine personalisierte Menüstruktur abbildet. Die Generierung benötigt neben einem Modell einer abstrakten Menüführung auch Modelle der Umgebung und der damit verbundenen bedienbaren Dienste. Die modellbasierte Erzeugung von Nutzerschnittstellen wurde in der Forschung bereits intensiv untersucht (s. z.B. Janesen et al. 1993, Shirgane & Fukazawa 1998). Hierbei werden aus Daten- und Dialogmodellen meist prototypische Systeme erzeugt, die vom Designer nachbearbeitet werden müssen. Die Verbindung der modellgetriebenen Generierung mit adaptierbaren oder adaptiven Nutzerschnittstellen ist eine noch weitgehend ungelöste Frage.

Einige Ansätze zielen darauf ab, Bedienoberflächen während der Laufzeit zu erzeugen, die nicht durch einen Designer oder Programmierer nachgebessert werden. Hierzu zählen Systeme wie der von Nichols et al. (Nichols 2004) vorgestellte Ansatz des „Personal Universal Controllers“ (PUC). Auch das vorgestellte Konzept des „Universal Interactors“ (Hodes et al. 1997), welches der Idee eines PUC entspricht, fällt in diese Gruppe.

Das hier vorgestellte Konzept lässt sich ebenfalls in die Gruppe der Systeme zur Erzeugung von UI zur Laufzeit einsortieren. Es unterscheidet sich aber von der Philosophie der oben genannten Systeme. So beziehen sich die bisherigen Ansätze immer auf eine einzige Anwendung und entwerfen eine Bedienoberfläche für diese eine Anwendung. Zwar kann der Benutzer die Anwendung wechseln, er erhält aber bestenfalls nur eine bidirektionale Fernbedienung, die getrennte Oberflächen einzelner Geräte zur Verfügung stellt, ähnlich einer Universalfernbedienung. Eine integrierende Bedienung von Umgebungen, die auch die Bindung von Bedienfunktionalitäten zu Geräten trennen kann, um unterschiedliche Funktionalitäten von unterschiedlichen Geräten in einer Menüebene zu bündeln, wird damit nicht erreicht und somit kann z.B. eine Bedienseite, die alle Funktionen zum Energiesparen über alle Geräte hinweg anbietet, bisher nicht automatisch erzeugt werden. Eine Personalisierung der Menüführung, welche gerade für intelligente Umgebungen sinnvoll erscheint, lassen die oben dargestellten Systeme ebenfalls vermissen.

In den letzten Jahren haben wir im Rahmen des inHaus-Projekts (inHaus) ein System entwickelt, welches unterschiedliche Hausbusse (wie EIB, EHS, IP/Ethernet) und deren Geräte abstrahiert und diese zusammenführt. So werden Geräte von unterschiedlichen Subsystemen abstrahiert durch Softwareproxies dargestellt. Diese Abstraktion entspricht nicht nur dem bisherigen inHaus-Ansatz, sondern wird nun auch im Amigo-Projekt eingesetzt. Sie ermöglicht übergeordneten Softwarediensten einen einfachen Zugriff auf die Geräte unabhängig von der tatsächlichen physikalischen Anbindung oder den benutzten Protokollen. Ein Dienst, wie z.B. ein GUI-Dienst, der für die Generierung von graphischen Bedienschnittstellen zuständig ist, kann nun auf die Softwareproxies der einzelnen Geräte zugreifen, und braucht dabei nicht die darunter liegenden (verborgenen) Schichten zu kennen. Der jeweilige Proxy ist für den Abgleich mit dem eigentlichen physikalischen Gerät verantwortlich.

Basierend auf einem solchen System konnten nun weitere Dienste aufgesetzt werden, die einen Mehrwert bieten, also z.B. Informationen aus Einzelgeräten, wie z.B. Sensoren zusammenführen und auch selbstständig Einstellungen an Geräten vornehmen können. Für die Erforschung von intelligenten Wohnumgebungen wurde in Duisburg ein spezielles Wohnhaus gebaut, in welchem wir unser System einschließlich der höherwertigen Funktionen seit Jahren entwickeln, testen und bewerten können. Der geschilderte Systemaufbau bildet die Grundlage und das zukünftige Testsystem für das im Folgenden beschriebene Konzept.

3 Erzeugung von Interaktionsstrukturen

In einer intelligenten Umgebung bieten vernetzte Geräte und übergeordnete Applikationen ihre Dienste sowohl anderen Diensten als auch dem Benutzer an. Für die Darstellung der durch den Benutzer durchführbaren Operationen (z.B. die Lautstärkeeinstellung) ist bei einer entsprechenden Funktionsvielfalt eine übersichtliche Präsentation unabdingbar. Deswegen werden Operationen gruppiert und die einzelnen Gruppen hierarchisch z.B. in einer Menüstruktur angeordnet. Das „Durchlaufen“ dieser Gruppierungen bis zur eigentlichen Bedienoperationswahl wird als Navigation bezeichnet. Die Art und Weise der Gruppierungen (die Strategie nach der gruppiert wird) wird im Folgenden Navigationsstrategie genannt.

Vorstellbare personalisierte Navigationsstrategien reichen dabei von ortsbezogenen Konzepten (der Benutzer wählt erst das Stockwerk, dann den Raum, dann das Gerät und dann die entsprechende Operation aus) über lebensbereichsorientierte Navigation (hier werden Geräte oder Bedienoperationen Bereichen des Lebens, wie Entertainment, Heath-Care, Vorratshaltung, etc. zugeordnet) bis hin zu Konzepten, bei denen eine herstellerabhängige Sortierung vorliegt. Neben grundsätzlichen Strukturierungsunterschieden können aber auch nur kleinere Variationen der Navigationsstrategien vorliegen. So könnte ein Benutzer, der prinzipiell eine ortsbezogene Idee bevorzugt, auf jeder Navigationsebene direkte Primärfunktionalitäten einsortiert haben wollen, die die Atmosphäre des Ortes beeinflussen (z.B. die Raumtemperatur) oder Energiesparfunktionen. Diese Funktionalitäten sind dann nicht mehr einem Gerät zugeordnet, sondern werden praktisch zu Funktionen des Raumes. Für die Erzeugung der Bedienstrukturen einer Umgebung wird neben einer Beschreibung der abstrakten Navigationsstrategie (siehe unten) auch ein Umgebungsmodell benötigt, welches die Beziehungen innerhalb der Umgebung beschreibt. Ein vereinfachter Auszug aus dem verwendeten Kon-

zept ist in Abbildung 1 dargestellt. Geräte und Softwaredienste bringen eine entsprechende Beschreibung mit, welche vom jeweiligen Hersteller des Dienstes bzw. des Gerätes geliefert werden muss. Die Daten der Umgebungskonfiguration (z.B. räumlicher Aufbau, Aufstellungsort) können teilweise durch andere Dienste ermittelt werden oder müssen durch den Benutzer bzw. einen Systemintegrator während der Inbetriebnahme eingegeben werden. Abbildung 2 zeigt eine vereinfachte graphische Darstellung eines solchen Beziehungsmodells einer beispielhaften Umgebung.

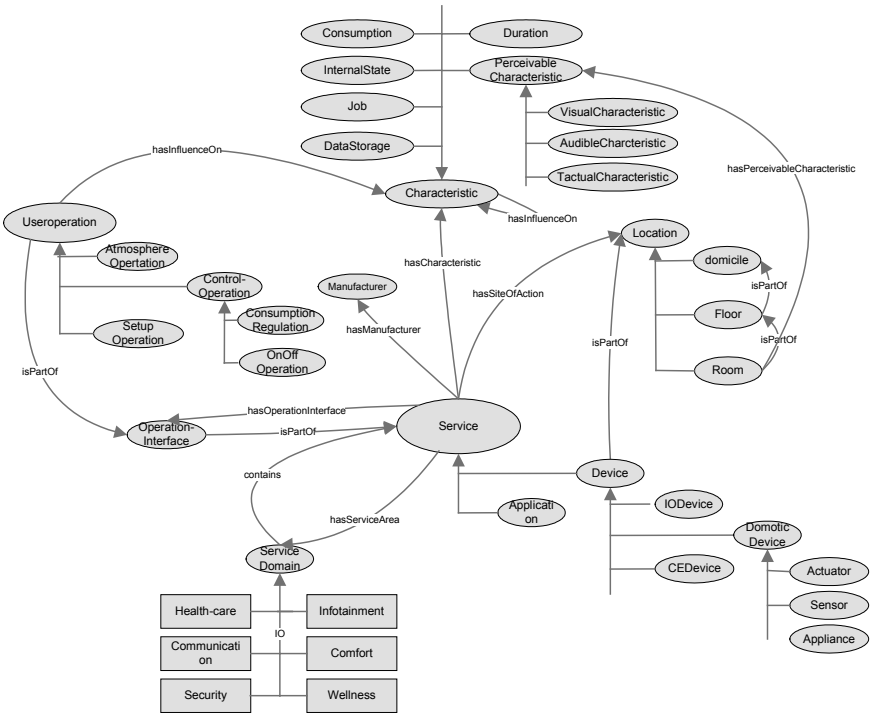

Abbildung 1: Auszug aus dem Ontologiekonzept, welches für die Modellierung der Umgebung benutzt wird

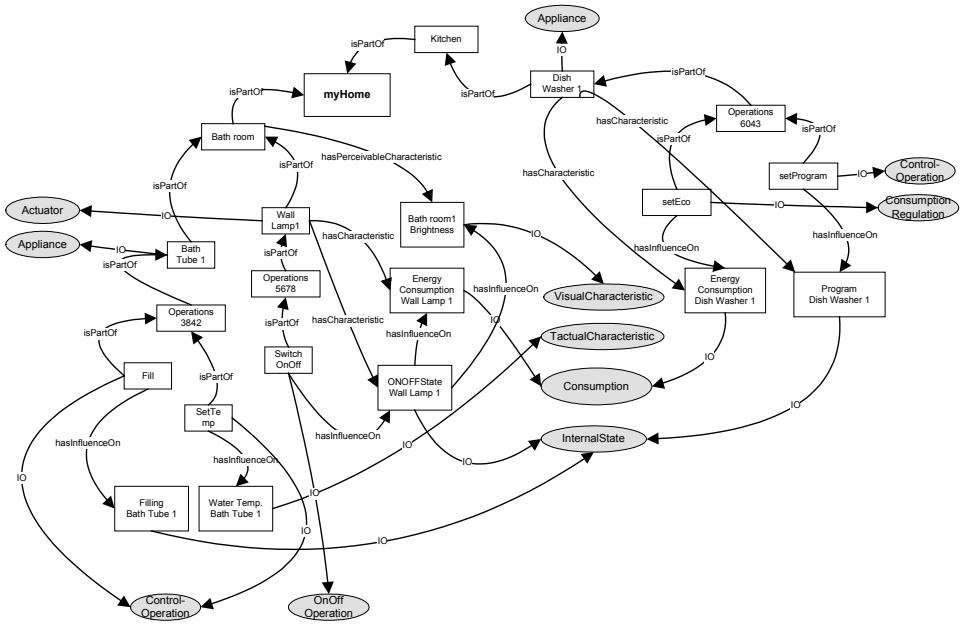

Abbildung 2: Darstellung eines beispielhaften Beziehungsmodells einer Umgebung unter Verwendung des Konzeptes aus Abbildung 1. Zur Steigerung der Übersichtlichkeit werden Beziehungen zu Konzepten nur teilweise dargestellt

Innerhalb dieses Graphen können nun Teilgraphen gesucht werden, deren Knoten in einer bestimmten Beziehung zueinander stehen. Diese Suche wird bei der Erzeugung der persönlichen Navigationsstrukturen eingesetzt. Das Modell der persönlichen Navigationsstrategie beschreibt deshalb, welche Beziehung Instanzen innerhalb einer gewissen Menüebene besitzen müssen, um in dem entsprechenden Navigationsschritt dargestellt zu werden. Hierbei werden feststehende Beziehungen innerhalb des Umgebungsmodells betrachtet, als auch Benutzereingaben aus den vorhergegangenen Navigationsschritten. Die durch den Benutzer getroffenen Entscheidungen in den höheren Ebenen bilden somit eine Verfeinerung der Beziehungsbedingungen in den unteren Ebenen.

Für die Beschreibung der Beziehungen der Menüeinträge innerhalb der Umgebung und der Beziehungen zu zuvor getätigten Benutzereingaben wurde von uns eine XML-Struktur entworfen. Wird diese Beschreibung rekursiv durchlaufen, entsteht die tatsächliche Navigationsstruktur zur Bedienung der aktuellen Umgebung.

Beispielhaft sei eine einfache „Raum – Gerät – Funktion-Navigationsstruktur“ in Worten dargestellt: Suche erst alle Instanzen des Konzeptes Room. Stelle dann alle Instanzen des Konzeptes Device dar, welche mit der vorherigen Benutzerwahl in einer isPartOf-Beziehung stehen. Dann zeige sämtliche Operationen des vorher gewählten Gerätes an (isPartOf-Beziehung). Natürlich können auch komplexere Suchen auf den Graphen angewandt werden. Die Suche nach Operationen, die z.B. direkten oder indirekten Einfluss (z.B. transitive Be-

ziehungen) auf eine Verbrauchseigenschaft (Consumption) haben, kann benutzt werden, um eine Bedienseite mit sämtlichen Operationen der Umgebung zur Energieeinsparung zu generieren. Für die Abbildung 2 würde dies auf die Operationen SwitchOnOff und SetEco zutreffen. Der beschriebene Ansatz erlaubt auch das multiple Auftauchen von Menüeinträgen in unterschiedlichen Menüebenen oder das mehrdimensionale Navigieren des Benutzers, z.B. nach Lebensbereichen und Räumen (eine mehrdimensionale Navigation ermöglicht dem Benutzer in unterschiedliche Richtungen zu navigieren, dabei schränkt die Spezialisierung einer Dimension (z.B. Lebensbereich) die Auswahlmöglichkeit in der anderen Dimension ein (z.B. Räume, die den gewählten Lebensbereich betreffen) und umgekehrt).

4 Filterung und Optimierung

Operationen, die zum betrachteten Zeitpunkt nicht durchführbar sind, sollten dem Benutzer nicht angeboten werden. Sie müssen also aus der Interaktionsstruktur herausgefiltert werden. Des Weiteren müssen die Operationen auf benötigte, vom Benutzer einzugebenden Informationen gemappt werden, damit dem Benutzer ein entsprechendes Interaktionselement zur Eingabe angeboten werden kann. Für diese Aufgaben wird ein weiteres Umgebungsmodell, das Abhängigkeitsmodell, herangezogen. Dienste wie auch Geräte werden innerhalb des Modells in einzeln ansprechbare Datenelemente und Funktionsmodule zerlegt. Datenelemente, welche den „Characteristics“ aus dem Beziehungsmodell entsprechen, soweit sie durch den Benutzer eingesehen und/oder manipuliert werden können, bilden die unterste Ebene des Modells. Zwischen ihnen existieren die folgenden Abhängigkeiten:

- Eine Änderung des Wertes ändert den aktuellen Wert eines anderen Datenelements sofort (zeitlich unmittelbar; z.B. Bandrestanzeige bei der Videorekorderprogrammierung).
- Eine Änderung des Wertes ändert die Menge möglicher Werte eines anderen Datenelements (z.B. ändert eine Bouquetwahl die auswählbaren Programmsender beim digitalen Satellitenreceiver).
- Ein bestimmter Wert des betrachteten Datenelements macht ein anderes Datenelement nur noch lesbar und nicht mehr schreibbar (z.B. die DHCP Aktivierung sperrt die Eingabe der IP-Adresse, welche aber noch darstellbar ist).
- Ein bestimmter Wert des betrachteten Datenelements inaktiviert ein bestimmtes Datenelement. Dieses ist dadurch weder les- noch beschreibbar (z.B. das Ausschalten des Fernsehers verhindert die Änderung der aktuellen Lautstärke).
- Ein bestimmtes Datenelement besitzt eine Affinität / Informationszusammenhang zu einem anderen. Dies bedeutet, dass der Benutzer zur Bearbeitung des einen Datenelements den aktuellen Wert der anderen gebrauchen kann (z.B. bei der Einstellung der Raumtemperatur).

Jedes Datenelement besitzt eine Selbstbeschreibung, welche für die Darstellung benötigt wird (siehe unten). Des Weiteren besitzt es ein Funktionsmodul, welches die Anbindung an die tatsächliche Methode des Softwareservice zum Updaten eines Wertes beschreibt. Nicht jedes Funktionsmodul besitzt ein zugehöriges Datenelement, da nicht alle Funktionalitäten

nur durch Datenelemente darstellbar sind. So existieren Funktionsmodule, welche nicht explizit auf ein bestimmtes Datenelement zugreifen und Funktionsmodule, welche zwar ein Datenelement verändern, die Art der Änderung aber nicht vom Benutzer bestimmt wird (z.B. das automatische Setzen der Uhrzeit).

Die einzelnen Funktionsmodule können nun zu Bedienoperationen, die dem Benutzer angeboten werden, zusammengefasst werden. Bedienoperationen beschreiben die Änderung eines oder mehrer Datenelemente als durchführbare Aufgabe, ohne über die Art der Änderung eine Aussage zu treffen (z.B. Lampeschalten). Jede Bedienoperation setzt sich somit aus einem oder mehreren Funktionsmodulen zusammen. Über die Funktionsmodule sind die Bedienoperationen an die Beschreibungen der Datenelemente (welche Eingaben kann der Benutzer machen) und an die tatsächlich aufzurufenden Methoden gebunden. Über die Bindungen an die Datenelemente wird auch die Verfügbarkeit von Bedienoperationen geregelt. Besitzt eine Bedienoperation ein Datenelement, das zum betrachteten Zeitpunkt nicht verfügbar ist, so steht die gesamte Operation dem Benutzer zurzeit nicht zur Verfügung und wird dem Benutzer daher nicht angeboten.

5 Aufbereitung für die Präsentation

Ein Layout-Modul fügt Informationen für die Darstellung zur Navigationsstruktur hinzu und passt diese Darstellung an die Möglichkeiten des darstellenden Gerätes an. Die Eingabe von Werten für Datenelemente erfolgt bei der Bedienung normalerweise nach der Wahl der entsprechenden Operation. Für Operationen, die nur ein einziges Datenelement betreffen, ist es sinnvoll, das Eingabeelement ohne eine vorherige explizite Wahl der Operation anzubieten. Dies muss natürlich durch das darstellende Gerät unterstützt werden. Der Benutzer braucht nicht erst die gewünschte Operation auswählen und muss dann den Wert setzen (erst „Lampe dimmen“ wählen und dann den Dimmwert setzen), sondern setzt den Wert direkt und drückt somit aus, dass er die entsprechende Operation mit dem eingestellten Wert durchführen möchte.

Bekanntlich besitzt jedes Datenelement eine Beschreibung, welche unter anderem die Beschreibung der semantischen Bedeutung des enthaltenen Wertes enthält (z.B. Helligkeit). Diese semantische Beschreibung wird für die Darstellung benötigt, um ein passendes Widget für die Repräsentation (Ein- und Ausgabe) aus einem dem Benutzer persönlich zugeordneten Widget-Repository herauszusuchen. Bei der Suche werden ebenfalls die Möglichkeiten des darstellenden Gerätes beachtet, so dass im persönlichen Widget-Repository eine Darstellungsform für eine bestimmte semantische Beschreibung für bestimmte Eigenschaften des darstellenden Gerätes gesucht wird. Sollte keine passende Darstellungsform gefunden werden, so wird eine semantisch ähnliche (meist generellere) Darstellungsform gesucht. So kann ein „Ein-und Ausschalten für Helligkeit“ durch ein generelles „Ein-Aus-Schalten“ ersetzt werden. Damit diese semantische Suche funktioniert, müssen sämtlich mögliche semantischen Beschreibungen in einer (erweiterbaren) Taxonomie erfasst werden.

Die Anordnung der Widgets für eine Bedienfunktionalität auf dem darstellenden Gerät erfolgt mit Hilfe der Abhängigkeitsbeschreibung. Widgets, die interne Informationsträger repräsentieren, werden in einer Gruppe zusammengefasst, wenn Abhängigkeiten zwischen

ihnen existieren. Innerhalb einer Gruppe werden die Widgets nach ihrem Einfluss, die sie auf andere Widgets haben, geordnet. In den obersten Ebenen befinden sich die Repräsentanten von Informationsträgern, die von keinem anderen Informationsträger abhängen und nur Einfluss auf andere Informationsträger haben. Die Sortierung spiegelt die Anordnung der Widgets auf der Oberfläche wieder, bei der im oberen Bereich die Eingaben getätigt werden, die Einfluss auf die weiteren Eingabeelemente besitzen. Für die Darstellung können zum einen frei zugängliche Bedienterminals (z.B. Touchscreens), bei denen sich der Benutzer z.B. über eine RFID-Armbanduhr identifiziert oder Personen zugeordnete PDAs als persönliches Bediengerät benutzt werden.

6 Zusammenfassung

Das von uns vorgestellte Konzept zur Erstellung von personalisierten Bedienstrukturen für intelligente Umgebungen ermöglicht es dem Benutzer unbekannte Umgebungen weitgehend intuitiv zu bedienen. Dazu werden Funktionalitäten der Umgebung auf bekannte, personalisierte Navigationsstrukturen, die sich als Menüführung darstellen lassen, abgebildet.

Personalisierte Navigationsstrategien können dabei, wie bisher üblich, geräte-orientiert sein, können aber auch die Bindung zwischen Gerät und Funktionalität auflösen und statt dessen Bedienseiten anbieten, die Geräte übergreifende Funktionen enthalten. Anders als bei bisherigen Ansätzen kann die Personalisierung von Menüstrukturen auf unbekannte Dienste und Funktionen angewandt werden, da nicht die Sortierung von bestimmten Instanzen in einer festen Menüstruktur festgelegt wird. Es wird vielmehr die Semantik, die in einer gewünschten Menüstruktur steckt, beschrieben, so dass auch neue Dienste sinnvoll integriert werden können. Durch die Einführung eines persönlichen Widget Repositorys wird auch bei der Darstellung ein möglichst hohes Angleichen an einer dem Benutzer bekannte Form erreicht. Methoden zur personalisierten Suche innerhalb einer Umgebungsbeschreibung und die Erzeugung von persönlichen Interaktionsstruktur als Finite-State-Machine wurden von uns auf Serverseite als Service realisiert und getestet. Clientoberflächen können diesen Service nutzen, um eine personalisierte Bedienschnittstelle darzustellen. Zur Zeit erfolgt eine Implementierung einer beispielhaften Clientoberfläche, welche die dargestellten Mechanismen der Serverseite nutzt. Die weitere Umsetzung des Konzepts erfolgt im Rahmen des Amigo-Projektes und wird auch in das inHaus integriert. Erste Nutzertests sind für Juni 2006 geplant.

Literaturverzeichnis

Amigo: Ambiente Intelligence for the networked home. http://www.amigo-project.org; Integriertes Projekt des 6. EU-Rahmenprogramms; Vertragsnummer: IST 004182.

Granic, A.; Glavinic, V. (2000): Functionality specification for adaptive user interfaces, in: 10th Mediterranean Electrotechnical Conference, 2000. MELECON 2000., Vol. 1, S. 123-126.

Hodes,T. D.; Katz, R. H.; Servan-Schreiber, E.; Rowe, L. (1997): Composable ad-hoc mobile services for universal interaction. In Proccedings of the 3rd annual ACM/IEEE international Conference on Mobile Computing and Networking, Seite 1-12.

inHaus: inHaus-Innovationszentrum Intelligente Raum- und Gebäudesysteme. http://www.inhaus-duisburg.de

Janssen, C.; Weisbecker, A.; Ziegler, J. (1993): Generating user interfaces from data models and dialogue net specifications. In: Proceedings of INTERCHI'93, ACM Press, Seite 418-423.

Kaltz, J. W.; Ziegler, J. (2006): Supporting Systematic Usage of Context in Web Applications. 19th International FLAIRS Conference, Special Track on Modeling and Applying Contexts in the Real World (MAC-06).

Menkhaus, G.; Pree, W. (2002): A hybrid approach to adaptive user interface generation, in: Proceedings of the 24th International Conference on Information Technology Interfaces, ITI 2002., Seite 185-190.

Nichols, J.; Myers, B. A.; Litwack, K.; Higgins, M.; Hughes, J.; Harris, T. K. (2004): Describing Appliance User Interfaces Abstractly with XML. In: Workshop on Developing User Interfaces with XML: Advances on User Interface Description Languages, Satellite Workshop at advanced visual interfaces, Gallipoli, Italy, Seite 9-16.

Norcio, A. F.; Stanley, J. (1989): Adaptive human-computer interfaces: a literature survey and perspective, in: IEEE Transactions on Systems, Man and Cybernetics, 2 / 1989, S. 399-408.

Oppermann, R. (2005): From User-adaptive to Context-adaptive Information Systems. i-com – Zeitschrift für interaktive und kooperative Medien, 4. Jahrgang (2005), Nr. 3, Seite 4-14.

Ressel, C. (2004): Integrierte Bedienung in intelligenten Umgebungen. In: Kongressdokumentation ehome – Home Solutions 2004, Messe-Berlin, Seite 70-72(engl.), Seite 73-75(deut.).

Scherer, K.; Grinewitschus, V. (2002): Das intelligente Haus: Neue Nutzeffekte durch integrierende Vernetzung im Bereich Wohnen und Arbeiten. In: VDE (Hrsg.): NetWorlds – Leben und Arbeiten in vernetzten Welten – VDE-Kongress 2002, Berlin, VDE-Verlag, S153-159.

Shirgane, J.; Fukazawa, Y. (1998): Method of User-Customizable GUI Generation and Its Evaluation. In: Proceedings on the 5th Asia-Pacific Software Engineering Conference, IEEE Computer Society, Digital Object Identifier: 101109/APSEC.1998.733742, Pages 377-384.

Tränkler, H. R (2001): Zukunftsmarkt intelligentes Haus. In: Tränkler,H.R.; Schneider, F. (Hrsg.): Das intelligente Haus, Seite 17- 34, München: Pflaum-Verlag GmbH & Co.KG.