

SOPRANO Ambient Middleware: Eine offene, flexible und markt-orientierte semantische Dienstplattform für Ambient Assisted Living

Andreas Schmidt¹, Peter Wolf¹, Michael Klein², Dirk Balfanz²

¹ FZI Forschungszentrum Informatik, Karlsruhe (andreas.schmidt | peter.wolf@fzi.de)

² CAS Software AG, Karlsruhe (michael.klein | dirk.balfanz@cas.de)

Kurzfassung

Das Europäische Integrierende Projekt SOPRANO zielt auf die Schaffung einer offenen Plattform für Ambient-Assisted-Living-Lösungen, die älteren Menschen länger ein unabhängiges selbstbestimmtes Leben im eigenen Zuhause ermöglichen sollen. Um solche Lösungen marktauglich zu machen, ist es von herausragender Bedeutung, sie durch einen flexiblen Aufbau anpassbar an die jeweiligen Bedürfnisse zu gestalten und eine funktionierende Marktökologie zu schaffen, bei der nicht ein einziger Anbieter alle Funktionen und Dienste entwickelt/betreibt, sondern unterschiedliche Akteure mit ihren jeweiligen Stärken zusammenwirken können. Dazu wird eine technische Plattform konzipiert, die es ihnen ermöglicht ihre Dienste, Produkte und Erfahrungen zu integrieren. Doch eine rein technisch verstandene Modularität reicht nicht aus, um in einer komplexen Domäne wie AAL ein reibungsarmes Zusammenspiel von unabhängig voneinander entwickelten Systemteilen zu gewährleisten. Hierzu werden Methoden zur Sicherung der semantischen Kohärenz benötigt, die in SOPRANO durch einen ontologiegestützten Architekturansatz und semantische Verträge zwischen den einzelnen Komponenten umgesetzt werden.

Abstract

The European Integrated Project SOPRANO aims at an open platform for ambient assisted living solutions, allowing older people a longer life on their own in their own environment. To be successful on the market, such solutions need to be flexible and adaptable to individual needs, supported by a living market ecology in which not a single vendor provides everything, but different actors can bring in their strengths. As an enabler, a technical platform is designed that allows for the integration of their services, products, and experiences. But a simple technical notion of modularity is not sufficient for complex domains like AAL to ensure a frictionless orchestration of independently developed system components. Here, methods for sustaining semantic coherence are needed. To that end, SOPRANO builds upon an ontology-based architecture and semantic contracts between individual components.

1 Einführung

Ambient-Assisted-Living-Lösungen für das eigene Zuhause, die ein längeres selbstbestimmtes Leben im gewohnten Umfeld ermöglichen, sind keine einheitlichen Produkte. Zu verschieden sind die Anforderungen, Wünsche, Voraussetzungen und finanziellen Möglichkeiten. So sollen sie einen sanften Einstieg bieten, indem sie Komfortfunktionen für nicht pflegebedürftige Menschen erbringen, aber auch Bestandteil einer umfassenden Pflegeversorgung (z.B. mit ambulanten Pflegediensten oder im Rahmen von betreutem Wohnen) sein. Damit spricht man unterschiedliche Zielgruppen, aber auch unterschiedliche Lebensphasen an, zwischen denen ein sanfter Übergang stattfinden soll. Eine solche Sichtweise fordert Flexibilität und Erweiterbarkeit der Lösungen [5].

Gleichzeitig müssen solche Lösungen auch kostengünstig herzustellen sein. Dies ist nur möglich, wenn man statt einer Vielzahl von speziellen Einzellösungen auf eine gemeinsame Plattform setzt, die sich leicht durch unterschiedliche kostengünstige Komponenten erweitern lässt. Wie viele Beispiele (z.B. in der PC-Welt oder auch im Bereich der Telekommunikation) zeigen, darf sich das Komponentenangebot dabei nicht nur auf einen einzelnen Her-

teller (den Plattformanbieter) beschränken, sondern muss für andere Marktteilnehmer offen stehen, um ein breites Angebot (auch an Nischenkomponenten) machen zu können und gleichzeitig die Skalenvorteile der Plattform zu nutzen.

Dieses Geschäftsmodell setzt voraus, dass sich eine funktionierende Marktökologie um die Plattform bildet, für die Plattform die folgenden notwendigen (aber nicht hinreichenden) Voraussetzungen erfüllen muss:

- *Klare, einfache und offene Schnittstellen*, um die Angebotshürden möglichst gering zu halten
- *Hinreichend feingranulare und mächtige Erweiterungsmöglichkeiten*, um kleine Nischenanwendungen zu ermöglichen
- *Robustes Zusammenspiel* von Komponenten unterschiedlicher Hersteller.

Die in Kapitel 2 beschriebene Analyse bisheriger Ansätze für AAL-Plattformen zeigt allerdings, dass sich umfassende Lösungen auf den Integrationsaspekt konzentrieren und weniger auf die Öffnung der Plattform. Das Europäische Integrierende Projekt SOPRANO (<http://www.soprano-ip.org>) zielt demgegenüber auf die Schaffung einer offenen Plattform für Ambient-Assisted-Living-Lösungen ab.

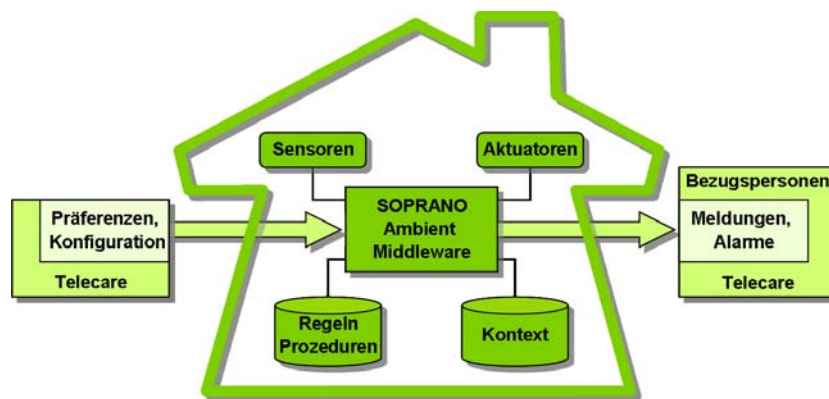


Bild 1: Übersicht SOPRANO-System

2 Existierende Systeme

Unter dem Etikett „Plattformen für AAL“ gibt es zahlreiche Ansätze, die in folgende Klassen eingeteilt werden können (vgl. [2]):

- **Hausautomationssysteme** haben das Ziel, den Wohnkomfort und die Sicherheit der Bewohner durch Automatisierung täglicher Aufgaben zu erhöhen. Aufgrund der technischen Herausforderungen bei der Integration sind die Erweiterungsmöglichkeiten auf einen bestimmten, sehr techniknahen Anwendungsfokus begrenzt. Beispiele für solche Systeme sind das IN-HOME-Projekt¹, welches auf die personalisierte Verwaltung von Audio- und Videomedien abzielt, sowie das Projekt EASY-LINE², das Intelligenz in Küchengeräte wie Kühlschränke und Waschmaschinen bringen will.
- **Agenten-basierte AAL-Systeme** zeichnen sich durch Geräte aus, die auf Basis von Agententechnologie autonom intelligent handeln. Durch Einbringen neuer Agenten kann ein solches System praktisch beliebig erweitert werden, allerdings müssen sich die neu eingebrachten Komponenten selbst um das Zusammenspiel mit anderen kümmern, so dass eine hohe Einstiegshürde zu überwinden ist und insbesondere die Robustheit nicht gewährleistet ist. Beispiele solcher Projekte sind das DynAMITE-Projekt³, das mittels einer SodaPop-Infrastruktur [21] die Grundlage für eine selbstorganisierende Anwendung schafft. Ähnlich geht das PERSONA-Projekt⁴ vor, das eine Bus-Technologie wählt, um die intelligenten Geräte verbinden zu können, sowie das AMIGO-Projekt⁵, in welchem ein dienstorientierter Ansatz und Techniken der Dienstkomposition zum Einsatz kommen. Jede

Komponente implementiert so eigene Integrationsstrategien.

- **Monolithische intelligente Systeme.** Systeme dieser Gruppe sind durch ein zentrales „Gehirn“ charakterisiert, welches in Form einer Blackbox arbeitet und die Intelligenz für alle angeschlossenen Geräte liefert. Solche Systeme basieren auf hoch spezialisierten Algorithmen meist unter Verwendung logischen Schließens. Erweiterungen am Gesamtsystem sind schwierig, da unabhängig austauschbare Komponenten fehlen. Forschungsprojekte dieser Kategorie sind das EMBASSI-Projekt⁶, welches sich auf Intelligenz für Wohnzimmer und Fahrzeuge spezialisiert hat sowie das MAP-Projekt [5], das vorwiegend im Büro-Umfeld agierte. Ein neuerer Ansatz ist das europäische Projekt EMERGE⁷, das Lösungen zur Reaktion auf Notfälle im Haus bereitstellen soll.

Bis heute hat keine der vorgestellten Lösungen eine hohe kommerzielle Aufmerksamkeit erreicht. Als Garant für einen Erfolg am Markt sehen wir eine flexible Architektur, die es ermöglicht Geschäftsinteressen verschiedener Gruppen unabhängig voneinander zu unterstützen.

2 Aufgaben einer AAL-Plattform und die Marktakteure

Legt man eine abstrakte Sichtweise wie in Abb. 1 auf ein AAL-System für das eigene Zuhause zugrunde, lassen sich folgende Marktakteure identifizieren.

- **Hardwarehersteller von Sensoren und Aktuatoren.** Diese entwickeln Geräte, die Aspekte der Umgebung direkt messen oder direkt auf sie einwirken. Hierzu gehören beispielsweise Sensoren zur Lokalisierung und zur Messung von Vitalwerten oder aber Aktuatoren zur Steuerung der Haustechnik.

¹ <http://www.ist-inhome.eu>

² <http://www.arenqueks.com/easynet>

³ <http://www.dynamite-project.org>

⁴ <http://www.aal-persona.org>

⁵ <http://www.amigo-project.org>

⁶ <http://www.embassi.de>

⁷ <http://www.emerge-project.eu>

- **Software-Entwickler von Mehrwertdiensten.** Da sich Sensorinformationen und die Möglichkeiten von Aktuatoren üblicherweise auf das direkt Messbare beschränken, sind in vielen Fällen weitergehende Dienste nützlich, die beispielsweise zusätzliches Wissen oder die Historie miteinbeziehen. Hierzu gehören Data Mining-Verfahren, die aus den Sensordatenströmen Muster ableiten (Aktivitätsmuster, „Gewohnheiten“), um bei Abweichungen Ereignisse auslösen zu können, aber auch einfache regelbasierte Verfahren, die Informationen unterschiedlicher Sensoren fusionieren und so neue Daten gewinnen.
- **Lösungsentwickler im Telecare-Bereich** spezifizieren typisches Systemverhalten durch generische Vorlagen für eine Systemkonfiguration. Dabei werden vorhandene Informationen der Sensoren und daraus abstrahierte Informationen kombiniert und in konkrete Systemreaktionen übersetzt. Ein Beispiel ist die Erinnerung an nicht abgeschaltete Geräte beim Verlassen der Wohnung. Hierzu werden beispielsweise Informationen, die von Stromverbrauchssensoren und RFID-Empfängern an der Wohnungstüre stammen, kombiniert, mit installations- und personenabhängigen Zusatzwissen, z. B. über Geräte die nicht angeschaltet bleiben dürfen. Als Systemreaktion wird die Person per Sprachausgabe an das Abschalten dieser Geräte erinnert.
- **Pflegedienstleister** haben durch ihren Kontakt mit den jeweiligen Kunden und ihr eigenes ergänzendes Dienstleistungsangebot das Wissen darüber, was für eine konkrete Installation benötigt wird. Dadurch können sie aus den durch die Lösungsentwickler bereitgestellten Verhaltensmustern auswählen und diese gezielt und ohne großes technisches Know-How anpassen. Dies gilt auch für versierte Endbenutzer oder deren Angehörige, sodass kein Pflegedienstleister vorausgesetzt werden muss.

3 Technische Implikationen: Die SOPRANO Ambient Middleware

Die wichtigste Implikation aus den vorangegangenen Betrachtungen ist die Anforderung der Erweiterbarkeit auf unterschiedlichen Abstraktionsebenen: auf der Ebene der Sensoren und Aktuatoren, auf der Ebene der Informationsverarbeitung, -anreicherung und -analyse sowie der Ebene des Systemverhaltens. Dabei ist es die Aufgabe der Middleware (und nur ihre Aufgabe), zwischen den unterschiedlichen Ebenen eine semantische Kohärenz herzustellen; für die jeweiligen Marktakteure muss eine Beschränkung auf ihre jeweilige Abstraktionsebene möglich sein.

Wie lässt sich eine solche Middleware konzipieren und realisieren? Die Grundprinzipien der SOPRANO Ambient Middleware sind (vgl. [3]):

- **Dienstorientierter Grundansatz** auf der Basis etablierter Technologien wie OSGi.
- Klare Trennung von unterschiedlichen Systemfunktionen entsprechend den **unterschiedlichen Abstraktionsebenen** der Realwelt.
- Verbindung der unterschiedlichen Ebenen und Funktionen über eine gemeinsame **Domänenontologie**
- Klare **semantische Verträge** zwischen den zentralen Middlewarekomponenten und ihren Erweiterungen

Diese Grundprinzipien sollen anhand der einzelnen Komponenten der SOPRANO-Middleware (Context Manager, Procedural Manager, Composer, Semantic Service Registry) näher erläutert werden.

3.1 Context Manager – Sammeln und abstrahieren von Kontextinformationen

Der *Context Manager* sammelt alle von den Sensoren gelieferten Informationen über die Umgebung ein und bietet seinen Dienstnehmern eine (synchron wie asynchron) anfragbare Zustandsdatenbasis, die dem in einer Kontextontologie repräsentierten gemeinsamen Verständnis entsprechen. Dieses gemeinsame Verständnis ist das Ergebnis eines iterativen und partizipativen Prozesses, der alle Interessensgruppen zusammenbringt (vgl. [4]).

Diese gesammelten Informationen sind in der Regel auf einem niedrigen Abstraktionsniveau (z.B. RFID-Scanner Y erkennt RFID-Tag X) und müssen z.B. durch Hintergrundwissen über die Konfiguration (RFID-Tag X identifiziert Person Z, RFID-Scanner ist an der Wohnungstür angebracht) in eine höhere Abstraktionsebene übersetzt werden (Person Z betritt oder verlässt das Haus).

Dabei ist dieses Übersetzen und Abstrahieren (man spricht auch von „semantic uplifting“) nicht immer trivial (vgl. [1]):

- Unterschiedliche Sensoren können aufgrund ihrer Unvollkommenheit für ein und dasselbe Kontextelement widersprüchliche Informationen liefern (z.B. Positionsbestimmung mittels RFID, Radar oder Stimmenerkennung in einem Raum). Hier muss der Kontextmanager Konflikte mit Hilfe von Heuristiken auflösen, um darauf aufsetzenden Komponenten ein konsistentes Bild zu liefern.
- Durch den unvollständigen Einblick, den Sensoren in die Realwelt ermöglichen, müssen gesammelte Informationen auch wieder kontrolliert vergessen werden, um statt falscher Informationen eher keine Informationen zu liefern.
- Manche Kontextinformationen erfordern die Kombination unterschiedlicher Sensoren oder die

Analyse eines längeren Zeitraumes (z.B. Erkennung von Abweichungen vom Normalverhalten).

Hier können Entwickler von Mehrwertdiensten hochwertige Speziallösungen bereit stellen, z.B. mittels Data-Mining-Techniken für die Erkennung von Gewohnheiten. Diese sogenannten „Augmentation Services“ können an den Kontextmanager über das Blackboard-Muster [8] angebunden werden, indem sie bei Änderungen benachrichtigt werden und ihre „Erkenntnisse“ (also Ergebnisse der Verarbeitung) in den Kontextmanager für die anderen Komponenten zurückschreiben.

3.2 Procedural Manager – Auswahl eines geeigneten Systemverhaltens

Der *Procedural Manager* lässt sich vom Context Manager über Änderungen benachrichtigen und wählt auf dieser Basis eine adäquate Systemreaktion auf. Die Systemreaktion und das auslösende Ereignis werden dabei in der Form: *[Ereignis, Kontextbedingungen]* => *Workflow* spezifiziert:

- Ein *Ereignis* ist dabei die Änderung eines bestimmten Kontextelementes (z.B. Person betritt einen anderen Raum, Temperator ändert sich).
- *Kontextbedingungen* sind zusätzliche Einschränkungen, die typischerweise vom Ereignis abhängen (z.B. betretener Raum ist die Küche). Zusammen mit einem Ereignis können somit auftretende relevante Situationen erfasst werden.
- Der *Workflow* ist ein Ablauf von Dienstaufrufen in der Geschäftsprozess-Sprache BPEL, die für die Orchestrierung von Web-Services konzipiert wurde, übliche Programmiersprachenprimitive wie Verzweigungen, Schleifen und anderen Berechnungsmöglichkeiten enthält und um einige spezielle Operatoren für den Umgang mit semantischen Informationen erweitert wurde.

Dabei wird ein SOPRANO-basiertes System nicht mit konkreten Prozeduren ausgeliefert (z.B. sind zu benachrichtigende Personen oder auch eine „angenehme“ Temperatur installationsspezifisch). Stattdessen existieren sogenannte *Procedure Templates*, die in generischer Form von Lösungsentwicklern im Telecare-Bereich auf der Basis ihres Domänenwissens aufgestellt wurden und gezielt Platzhalter enthalten, um dann von Pflegedienstleistern (oder versierten Angehörigen/den Personen selbst) fallspezifisch instanziiert zu werden.

Hierbei ist wichtig, dass aus Wartbarkeitsgründen das Systemverhalten möglichst abstrakt beschrieben werden sollte (also: „Warnen einer Person“ statt „Optische Signalleuchte anschalten“).

3.3 Composer – Ausführung des Systemverhaltens mit abstrakten Diensten

Der *Composer* empfängt den abstrakten Workflow und setzt diesen kontextabhängig und auf die Befindlichkeit der betreuten Person abgestimmt in den Aufruf real vorhandener Geräte (in Form von Diensten) mit den korrekten, detaillierten Parametern um. Hierzu gehört die Ermittlung der jeweils aktuell und aktiv im System zugänglichen Dienste (durch eine OSGi Service-Registry), die intelligente Auswahl der zum Workflow passenden Dienste, sowie die Zusammenstellung der konkreten Aufrufparameter. So wird z.B. die abstrakte Anweisung „Warnung der Person“ übersetzt in einen konkreten Text (Auswahl aus parametrisierten Standardsprachelementen), die Ermittlung des aktuell verfügbaren Ausgabegerätes als das eingeschaltete Fernsehgerät und der konkrete Aufruf des speziellen Services welcher den Warnungstext auf dem TV darstellt.

Dieses Herunterbrechen stellt die Gegenrichtung zur Funktionalität des Context Manager dar. Hierzu werden sog. *virtuellen Dienste* eingesetzt, die die Zustandsänderung „Person gewarnt“ übertragen in „Optische Signalleuchte X ist angeschaltet“, falls die betreffende Person unter Hörproblemen leidet.

Mehrwertdiensteanbieter können analog zur Kontextanreicherung auch hier durch besondere virtuelle Dienste ein komplexes Systemverhalten ermöglichen oder erleichtern.

3.4 Semantic Service Registry – kontextspezifische Auswahl von konkreten Diensten

Der Composer versucht, die abstrakt gewünschte Funktionalität durch konkret vorhandene Dienste (Aktuatoren) zu erbringen. Hierzu müssen die entsprechenden Dienste semantisch beschrieben werden. Hierzu kommt eine semantische Dienstbeschreibungssprache auf der Basis der DIANE Service Description (DSD, [7]) zum Einsatz, die in die Registrierungsinformationen gemäß OSGi integriert wurde.

Entwickler von Aktuatoren müssen demnach ihre Dienste gemäß den Vorgaben von OSGi entwickeln und durch eine semantische Dienstbeschreibung ergänzen. Dann kann der neue Dienst eingebunden werden.

4 Fazit

Die SOPRANO Ambient Middleware baut auf in anderen Kontexten erprobten semantischen Techniken zum Umgang mit Kontextinformationen [1] und zur Dienstbeschreibung [7] auf und kombiniert dies mit Industriestandards wie BPEL und OSGi. Wie in Abbildung 2 nochmals zusammengefasst ist eine wesentliche Eigenschaft der Architektur der Middleware, dass sie für die unterschiedlichen Akteure klare Erweiterungspunkte vorsieht, die nicht nur durch syntaktische Vereinbarungen (Einhaltung einer

bestimmten Methodensignatur oder eines bestimmten XML-Schemas), sondern auch durch ein gemeinsames Domänenverständnis (einer Ontologie) spezifiziert sind. Dabei wurde besonderer Wert auf die Klarheit und Einfachheit der Schnittstellen gelegt, um keine technischen Hürden aufzubauen.

- *Anbieter von Sensoren und Aktuatoren* müssen nur entsprechend der Kontextontologie beschreiben, was ihre Geräte liefern, und auf der Basis von OSGi ansprechbar gestalten. Sie können innovative Sensoren mit zusätzlichen Softwarekomponenten (wie z.B. Analysediensten) ausliefern, müssen dies jedoch nicht und können diese Aufgabe entweder den bereits vorhandenen Standardkomponenten oder anderen Spezialanbietern überlassen. Dadurch können auch einfache und damit kostengünstige Sensoren eingebunden werden – anstatt die Intelligenz eines Systems auf viele kleine Komponenten zu verteilen.
- *Mehrwertdiensteanbieter* zur Abstraktion von Sensordaten können sich über bestimmte Änderungen benachrichtigen lassen, ihre Berechnungen durchführen und ihre Änderungen zurückschreiben. Sie müssen sich hierbei nur an die Kontextontologie und ihre Semantik halten.
- *Lösungsentwickler* von Abläufen für ein bestimmtes Systemverhalten kennen durch die Kontextontologie die möglichen Informationen, auf die sie zurückgreifen bzw. Eigenschaften, auf die sie einwirken können. Sie müssen sich nicht darum kümmern, welcher konkrete Dienst verfügbar ist oder auf welchem Wege beispielsweise die Position einer Person ermittelt wird. Dies übernehmen Composer und Context Manager.
- *Pflegedienstleister, Angehörige oder die direkten Nutzer* selbst können vorgefertigtes Systemverhalten aktivieren/deaktivieren und durch ihre Präferenzen anpassen, ohne komplexe Programmierung vornehmen zu müssen.

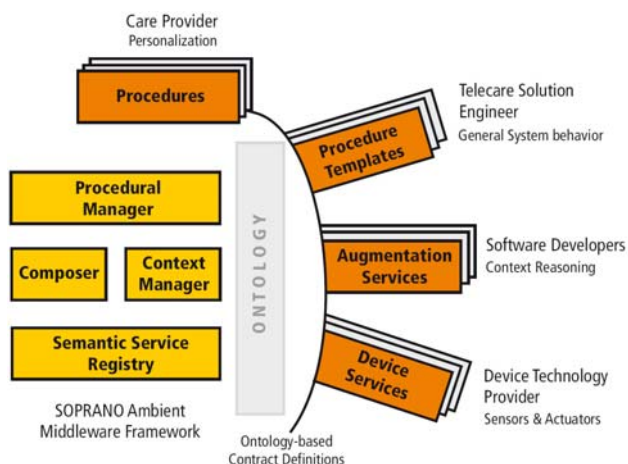


Bild 2 Marktakteure und ihre technischen Rollen

Die SOPRANO Ambient Middleware wird im Rahmen der Evaluation mit unterschiedlichem Dienstangebot in einer groß angelegten Feldstudie und im Rahmen spezieller Smart-Home-Umgebungen getestet. Parallel dazu sollen die semantischen Ansätze zur Dienst- und Kontextbeschreibung und –verwaltung in Standardisierungsbemühungen im AAL- und OSGi-Umfeld einfließen.

Das Integrierte Projekt SOPRANO wird von der Europäischen Kommission im 6. Rahmenprogramm im Bereich eInclusion kofinanziert. Wir danken den anderen Projektpartnern für die konstruktiven Diskussionen, die die Konzeption der SOPRANO Ambient Middleware begleitet haben.

5 Literatur

- [1] Andreas Schmidt: Ontology-Based User Context Management: The Challenges of Imperfection and Dynamics. In: International Conference on Ontologies, Databases and Applications of SEMantics (ODBASE 2006), On the Move Federated Conferences (OTM 2006), Montpellier, Lecture Notes in Computer Science vol. 4275, Springer, 2006, pp. 995-1011
- [2] D. Balfanz, M. Klein, A. Schmidt, M. Santi: Partizipative Entwicklung einer Middleware für AAL-Lösungen, GMS Medizinische Informatik, Biometrie und Epidemiologie, vol. 4, no. 3, 2008
- [3] P. Wolf, A. Schmidt, M. Klein: SOPRANO - An extensible, open AAL platform for elderly people based on semantical contracts. In: 3rd Workshop on Artificial Intelligence Techniques for Ambient Intelligence (AITAmI'08), 18th European Conference on Artificial Intelligence (ECAI 08), Patras, Greece, 2008
- [4] M. Santi, A. Schmidt, W. Beinhauer, M. Klein, J. Link: SOPRANO - Partizipative Entwicklung dienstorientierter Infrastrukturen für das Ambient Assisted Living. In: 1. Deutscher Kongress Ambient Assisted Living (AAL 2008), VDE Verlag, 2008
- [5] C. Kunze, A. Schmidt, C. Holtmann, W. Stork: Kontextsensitive Technologien und intelligente Sensorik für Ambient-Assisted-Living-Anwendungen. In: 1. Deutscher Kongress Ambient Assisted Living (AAL 2008), VDE Verlag, 2008
- [6] D. Balfanz D, J. Schirmer, M. Grimm M.R. Tazari: Mobile situation-awareness within the project MAP. Computers & Graphics 27(6), 2003: 893-898
- [7] U. Küster, B. König-Ries, M. Klein, M. Stern: DIANE - A Matchmaking-Centered Framework for Automated Service Discovery, Composition, Binding and Invocation, Proceedings of the 16th International World Wide Web Conference (WWW2007), Mai 2007
- [8] B. Hayes-Roth, 'A blackboard architecture for control', Artificial Intelligence, 26, 251-321, 1985.