

– DYNAMITE –

Dynamisch Adaptive Multimodale IT Ensembles

Thomas Kirste
Fraunhofer-Institut für Graphische Datenverarbeitung
Thomas.Kirste@igd.fraunhofer.de

Kurzfassung

Multimodale, intelligente Interaktion mit verteilten, dynamisch veränderlichen und spontan vernetzten Geräteinfrastrukturen – wie etwa dem intelligenten Haus – erfordert Softwarekonzepte, die ein kohärentes Agieren der Einzelkomponenten ermöglichen. Die Entwicklung eines solchen Softwareframeworks ist Thema des Verbundprojektes „DYNAMITE“ (Dynamisch Adaptive Multimodale IT-Ensembles). Das DYNAMITE-Framework soll es ermöglichen, dass Geräte und Softwarekomponenten unterschiedlichster Hersteller spontan kooperieren können, um gemeinsam die Interaktion des Nutzers zu analysieren, seine Intentionen zu bestimmen und Strategien für die Realisierung des Nutzerwunsches durch den Geräteverbund umzusetzen. Diese *semantische* Kohärenz ist oberhalb der üblichen Middlewarefunktionen für Spontanvernetzung und Plug-and-Play angesiedelt.

1 Einführung

1.1 Allgegenwärtige intelligente Geräte

Schlagworte wie „Ambient Intelligence“, „Pervasive Computing“ und „Information Appliances“ versprechen für die Zukunft eine allgegenwärtige Verfügbarkeit von Informationstechnik und Multimedia-Informationen. Unsere alltägliche Umwelt wird in die Lage versetzt, unsere Ziele und Bedürfnisse zu erkennen und kann uns intelligent und situationsgerecht unterstützen. Projekte wie „Oxygen“ am MIT, „Ambience“ in der EU oder die BMB+F-Leitprojekte EM-BASSI und SmartKom entwickeln Beispiele für solche Umgebungen.

In Zukunft werden nahezu alle Gegenstände des täglichen Lebens mit IT-Funktionalität angereichert sein. Die persönliche Umgebung wird intelligent, multimodal, proaktiv und global drahtlos vernetzt. Die allgegenwärtige Informationstechnologie unterstützt den Anwender jederzeit und an jedem Ort, ohne ihn mit komplizierten Konfigurations- und Bedienvorgängen abzulenken. Mein PDA erzählt meinem Wecker, wann er zu klingeln hat, beim Projektmeeting zeigt mein Notebook automatisch die aktuell relevanten Dokumente, der Projektor schaltet automatisch auf meinen Notebook um, sobald ich meine Präsentation beginne, mein Fernseher stört mich nicht, wenn er mich bei entspannender Lektüre erwischt, sondern zeichnet die Nachrichten für mich auf, mein Telefon weiss, wann und mit wem ich über welches Projekt gesprochen habe und erzählt meinem PDA, dass ich versprochen habe, zum nächsten Meeting ein überarbeitetes Proposal mitzubringen. So zumindest einige der Visionen im Bereich „Intelligent Environments“, „Multimedia Appliances“ und „Pervasive Computing“ (siehe z. B. auch einschlägige Projekte wie [5, 8, 10, 13]).

Die Realisierung intelligenter Umgebungen im praktischen Einsatz außerhalb der Labore erfordert jedoch die Interoperabilität von Diensten und Geräten, die Fähigkeit zur dynamischen



Abbildung 1: Umgebungen, die intelligent sein sollten: Konferenzräume

Vernetzung und zur Selbstorganisation. Technologien wie UPnP, Rendezvous und JINI bilden hierfür wichtige Grundlagen. Darüber hinaus ist aber die Fähigkeit zur semantischen Selbstorganisation notwendig, zur spontanen und automatischen Kooperation multimedialer Dienste und Geräte zur Erkennung und Erfüllung der Nutzerbedürfnisse. Die Entwicklung von hierfür geeigneten Software-Technologien ist Ziel des Projektes DYNAMITE [2].

Partner des Projektes sind das European Media Lab, das Fraunhofer-Institut für Graphische Datenverarbeitung (Konsortialführer) und die Loewe Opta GmbH.

1.2 Kohärente Ensembles

Eine Umgebung, in der verschiedenste Geräte selbständig und ad-hoc miteinander kooperieren um die Ziele des Nutzers zu identifizieren und diese gemeinsam zu erreichen, wirft vielfältige Fragen auf (siehe hier z. B. [1]). Auf der Softwareseite ist sicherlich eine der größten Fragen, wie die *intelligente ad-hoc Interoperabilität* zwischen verschiedensten Geräten realisiert werden kann: wie kann eine Menge lose gekoppelter Einzelgeräte, die in wechselnden Verbänden agieren, in die Lage versetzt werden, als *kohärentes Ensemble* zu agieren?

Für statische Infrastrukturen, in denen die Geräte, Nutzer, Dienste und ihre Beziehungen untereinander fest stehen, ist es im Prinzip immer möglich, kooperierende Geräte-Ensembles hart zu verdrahten und die Rolle der einzelnen Komponenten in der vom Ensemble realisierten Applikation durch den Systemdesigner festzulegen. In einer Welt der ad-hoc Ensembles verliert der Begriff „Applikation“ jedoch seine übliche Bedeutung. Anstelle von umfangreichen und komplexen Bündeln von Funktionen, welche den klassischen Applikationsbegriff charakterisieren, werden in einer intelligenten Umgebung durch die Mitglieder des Ensembles eine Vielzahl von Einzelfunktionen bereitgestellt, die sich „spontan“ zusammenfinden, um ein spezifisches Nutzerbedürfnis zu erfüllen. „Applikationen“ – genauer gesagt: flüchtige Funktionsensembles – werden je nach Bedarf und Verfügbarkeit gebildet und lösen sich nach der Umsetzung dieses Ziels wieder auf. Situationsbild, Intentionserkennung, effektorisches Potential, Strategieentwicklung und -ausführung sind die Bereiche, die durch ein derartiges funktionales Ensemble abgedeckt werden müssen.

Die Herausforderung besteht nun in der Entwicklung von Softwaremechanismen, die eine

solche dynamische und selbsttätige Erzeugung von kohärenten Geräteverbänden ermöglichen – denn ein Systemdesigner ist zur Laufzeit nicht verfügbar.

Damit die vielfältigen Visionen des Pervasive Computing und der Intelligenten Umgebungen auch in der Breite Wirklichkeit werden, müssen diese Softwaremechanismen dabei in Form einer *offenen Plattform* geschaffen werden, die den Geräten und Software-Komponenten unterschiedlichster Hersteller die intelligente ad-hoc Kooperation ermöglicht.

Ziel von DYNAMITE ist die Realisierung einer solchen Plattform – bestehend aus Spezifikation, Referenzimplementierung sowie verschiedenen exemplarischen Applikationen auf dieser Basis. Hierfür ist unter anderem die Ausarbeitung, Weiterentwicklung und Vertiefung von Ideen zu einem Software-Framework für den Aufbau selbstorganisierender, verteilter multimodaler Systeme vorgesehen, die von den Partnern in ersten Ansätzen im Rahmen des BMB+F-Leitprojektes EMBASSI [4] entwickelt wurden.

Eine besonderer Schwerpunkt liegt dabei auf der intelligente Nutzung heterogener multimodaler Ausgabegeräte. In einem vernetzten Haushalt kann es so z. B. möglich sein, dass ein Benutzer in der Küche über einen Lautsprecher per Sprachausgabe die Termine des Tages vorgelesen bekommt. Während er ins Wohnzimmer geht, wird die Ausgabe automatisch auf einen Lautsprecher im Wohnzimmer umgeleitet. Bisherige multimodale Dialogsysteme treffen stets Annahmen über die Ausgabegeräte, die durch das System verwendet werden können. Dadurch kann aber nur eine geringe, feste Menge an Geräte unterstützt werden. In DYNAMITE entwickeln wir ein flexibles Modell für eine Dienstbeschreibung von multimodalen Ausgabeagenten, das die ad-hoc Integration von Ausgabegeräten in die multimodale Präsentationsstrategie des Ensembles erlaubt. Insbesondere erlaubt dies Entwicklern eigene Ausgabekomponenten zu realisieren, die automatisch in einen multimodalen Dialog eingebaut werden können, ohne dazu existierende Komponenten (z. B. einen Präsentationsplaner) modifizieren zu müssen.

1.3 Existierende Ansätze

Existierende Agenteninfrastrukturen für verteilte, multimodale Systeme sind zum Beispiel die Galaxy Communicator Architektur (Mitre Corp., [9]), die Open Agent Architecture des SRI (Stanford, [11]), und das Metagluе-System (MIT Oxygen, [8]). Ziel dieser Systeme ist die Bereitstellung der Basisfunktionen für den Aufbau kooperierender dynamischer Agentensembles: Das Vermitteln von Botschaften zwischen Agenten (*Routing*) und das Identifizieren von Agenten mit bestimmten Fähigkeiten (*Yellow Pages*).

Der Galaxy Communicator setzt eine zentrale „Nabe“ (*Hub*) für das Vermitteln von Botschaften zwischen Agenten ein. Der sequentielle Fluß einer Botschaft durch ein Agenten-Ensemble wird mit Hilfe von *Routing-Regeln* beschrieben, die der Hub befolgt. Dadurch können die globalen Kommunikationsstrukturen einer Multi-Agenten-Anwendung definiert werden, ohne dass diese den Agenten selbst bekannt sein müssen.

Die Open Agent Architecture (OAA) verwendet ein Kommunikationsmodell, bei dem komplexe Anfragen eines Agenten selbständig in Teilanfragen für die verfügbaren Auftragnehmer zerlegt werden können (auf Basis eines Prolog-ähnlichen Pattern-Matching-Verfahrens). Diese dynamische Auftragsdekomposition entlastet den Auftraggeber von der komplexen Aufgabe, *selbst* ein Ensemble zusammenstellen und koordinieren zu müssen, das seinen Auftrag kooperativ lösen kann.

Das Metagluе-System bietet vor allem die Möglichkeit, dass Agenten Ressourcen reservieren können. Es übernimmt die Aufgabe, aus dem Pool der verfügbaren Ressourcen gemäß der aktuellen Anforderungen der Agenten geeignete Ressourcen auszuwählen und den Agenten für

die Nutzung zuzuweisen. Die Programmierung eines Agenten kann dadurch von den konkreten Ressourcen vollkommen abstrahieren. Lediglich die spezifischen Anforderungen sind zu definieren („ein Display mit einer Auflösung von mindestens 1024×768 “). Der Agent ist damit unabhängig von den aktuell im Ensemble verfügbaren Ressourcen.

Der Vollständigkeit halber ist festzuhalten, dass existierende Middleware für die Spontanvernetzung – wie etwa HAVi [6], Jini [12] und Universal Plug-and-Play [14] – im Hinblick auf die umgesetzten *Konzepte* ein Subset der Funktionen der o. g. Agentensysteme bereitstellen¹. Sie können im wesentlichen durch OAA ohne Auftragszerlegung subsummiert werden. Daher verzichten wir an dieser Stelle auf eine tiefergehende Darstellung.

Wie sehr eignen sich diese Ansätze für die Koordination und Kommunikation von ad-hoc Ensembles?

Zunächst kann festgestellt werden dass obige Ansätze *zentralistische* Kommunikationsmodelle verfolgen. Eine zentrale Systemkomponente (Router, Hub, Resource-Manager, . . .) ist für die Kommunikations- und Koordinationsaufgaben verantwortlich. Diese Komponente bietet typischerweise relativ komplexe Funktionen an und ist damit „schwergewichtig“. Dies erschwert die Realisierung in einem physisch verteilten und dynamisch veränderlichen Ensemble – insbesondere, wenn Geräte mit sehr begrenzten Ressourcen partizipieren sollen.

Eine größere Herausforderung ist jedoch die Tatsache, dass es sich bei den o. g. Konzepten um *universelle* Kommunikationsinfrastrukturen handelt.

Die Einzelkomponenten eines Ensembles decken jeweils eine oder mehrere funktionale Rollen ab (z. B. Eingabegerät, Strategieentwicklung, Präsentationsplanung, Ausführungssteuerung), zwischen denen spezifische Kommunikationsbeziehungen bestehen – die also die „Anwendungstopologie“ bzw. die *Architektur* des Ensembles definieren. Universelle Kommunikationsmechanismen legen *keinen* Rahmen von Schnittstellen fest, durch den die spezifischen Kommunikationsbeziehungen – die Anwendungstopologie – in einer Anwendungsdomäne vorgegeben werden (ein solcher Rahmen würde auch im Widerspruch zur Universalität stehen). Welche Anwendungstopologie bedient werden muss und welche Rolle eine Komponente in dieser Topologie spielt, ist bei den o. g. Ansätzen Teil der Komponentenlogik. Dadurch können sie nur unter großem Aufwand in *unterschiedlichen* Ensembles eingesetzt werden, in denen sich verschiedene Strategien zur Komponentenkoordination entwickelt haben².

So arbeiten auch bisherige multimodale Dialogsysteme (z. B. [16]) mit einer festen Menge von Ausgabeagenten, die nicht erweitert werden kann. Insbesondere müssen bei der Entwicklung neuer Ausgabeagenten erst Systemkomponenten (z. B. Präsentationsplaner) angepasst werden, bevor es möglich ist, die neuen Ausgabeagenten zu unterstützen.

2 Der DYNAMITE-Ansatz

2.1 Grundlagen

In Bezug auf die Selbstorganisation von Geräte-Ensembles lassen sich zwei unterschiedliche Aspekte unterscheiden:

Architektonische Selbstorganisation bezeichnet die sinnvolle Integration eines Gerätes in die Anwendungstopologie, die Kommunikationsstruktur des Ensembles.

¹Wobei diese Umsetzung jedoch in den Details z. T. sehr viel tiefer gehen – vgl. etwa die detaillierte Auflistung der Funktionen von A/V-Geräten in HAVi.

²Galaxy versucht dieses Problem über Routing-Regeln im Hub zu umgehen, jedoch auf Kosten einer schwergewichtigen Zentralkomponente.

Operationale Selbstorganisation macht neue Funktionalität, die durch die Integration eines Gerätes in das Ensemble entsteht, für den Nutzer zugänglich. Wenn z. B. ein DVD-Spieler in ein Ensemble integriert wird, das bereits einen DVD-Recorder enthält, entsteht die neue Funktion „DVD kopieren“.

Eine vollständige Behandlung von „Selbstorganisation“ erfordert die Berücksichtigung beider Aspekte – in DYNAMITE konzentrieren wir uns jedoch zunächst auf die *architektonische* Selbstorganisation³.

Grundlage des DYNAMITE-Ansatzes ist das Konzept des „Kanals“ – ein Kommunikationsbus, über den Agenten Botschaften austauschen können. Im Prinzip erfolgt die Kommunikation nach dem Publish/Subscribe-Modell: Auf einem gegebenen Kanal können Agenten Botschaften in der spezifischen Sprache des Kanals publizieren; diese Botschaft wird an alle Agenten weitergeleitet, die den Kanal subskribiert haben und deren Subskriptionsfilter die Botschaft akzeptiert. Kanäle repräsentieren damit $n : m$ Kommunikationsbeziehungen. Zusätzlich können sie Konflikte zwischen Subskribenten auflösen (falls mehrere potentielle Empfänger eine Botschaft exklusiv für sich beanspruchen) und komplexe Anfrage über mehrere Agenten zur kooperativen Bearbeitung verteilen. Kanäle dürfen auch individuelle Kommunikationsstrategien verwenden, die jeweils auf die spezifischen Eigenschaften der Kommunikationsinhalte des Kanals, der Subskriptionsmuster, und der Konfliktlösungsstrategien zugeschnitten sind.

DYNAMITE-Kanäle sind damit explizit *nicht* als universelle Kommunikationsmechanismen vorgesehen. Statt dessen bieten sie die Möglichkeit, für spezifische Kommunikationsinhalte maßgeschneiderte Kommunikations- und Kooperationsstrategien als *Infrastruktur* anzubieten. Dadurch kann DYNAMITE die Probleme universeller Kommunikationsinfrastrukturen vermeiden (vgl. Abschnitt 1.3).

Der Verzicht auf Universalität zieht allerdings die Verpflichtung nach sich, einen Satz von Standard-Kanälen zu definieren, mit dessen Hilfe sich die Kommunikationsbedürfnisse in Geräte-Ensembles abdecken lassen. Es ist also notwendig, eine „Standard-Topologie“ für Geräte-Ensembles zu identifizieren.

2.2 Ensemble-Topologien

Ausgangspunkt der Definition der Standard-Topologie ist die Beobachtung, dass es zwei grundlegende Kommunikationsinhalte gibt, über die sich Geräte in einem Ensemble austauschen müssen: Nutzerinteraktionen können über das Ensemble verteilt auftreten und komplexe Aktionen des Gesamtensembles müssen über die Mitglieder verteilt ausgeführt werden. Gleichzeitig haben Geräte in Bezug auf Nutzerinteraktionen und Geräteaktionen einen sehr homogenen Aufbau: Interaktionskomponenten des Gerätes (Schalter, Regler, GUI, ...) liefern Interaktionsereignisse an eine Steuerungsapplikation, die daraus die erforderlichen (vom Nutzer intendierten) Aktionen des Gerätes bestimmt und entsprechend Aktoren ansteuert (vgl. Abb. 2, Basistopologie links).

Somit erscheint es nahe liegend, diese implizit in Geräten vorliegende Datenfluss-Struktur („Interaktions-Pipeline“) explizit zu machen und die dort kommunizierten Inhalte über das Ensemble zu verteilen (Abb. 2, Basistopologie rechts). Damit liefert jedes Gerät seine individuelle

³Ein möglicher Ansatz für die Realisierung operationaler Selbstorganisation ist die explizite Modellierung der Semantik von Gerätefunktionen in Bezug auf ein geeignetes Umgebungsmodell, etwa als „Vorbedingung / Effekt“-Regeln. Auf dieser Basis lassen sich Planungsverfahren einsetzen, die dann die Fähigkeiten aller verfügbaren Geräte berücksichtigen, um ein bestimmtes Nutzerziel zu erreichen. Siehe z. B. [7] für einen solchen Ansatz.

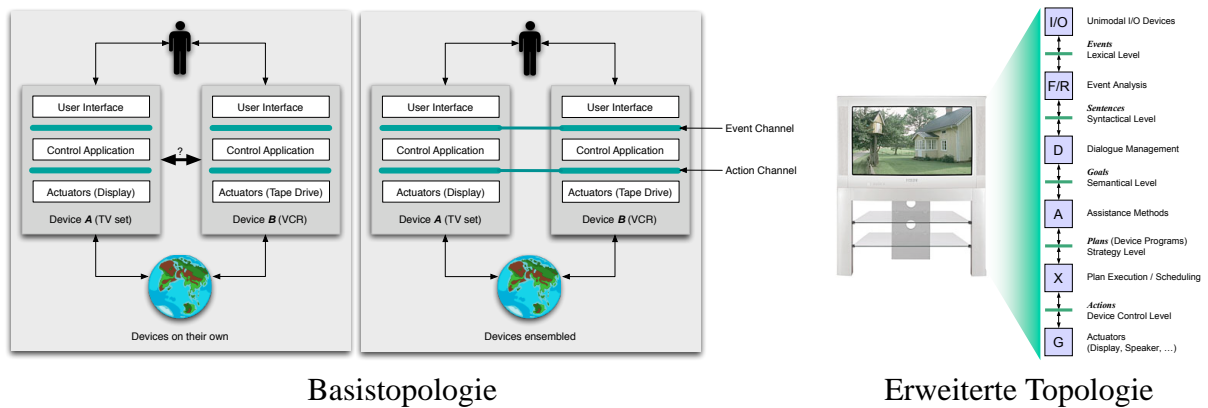


Abbildung 2: Kommunikationsabläufe

Beiträge zu der nun globalen, über das Ensemble verteilten Interaktions-Pipeline. Die Stufen dieser globalen Pipeline werden durch *Kanäle* voneinander getrennt. Diese Kanäle übernehmen auch die Koordination der funktionalen Beiträge der verschiedenen Geräte zu der vom jeweiligen Kanal belieferten Pipeline-Stufe.

Weiterhin verlangt man, dass Kanäle das dynamische An- und Abmelden von Lesern und Schreibern erlauben müssen. Dann ist es möglich, Geräte ad-hoc in ein Ensemble zu integrieren bzw. aus diesem zu entfernen. Die funktionalen Beiträge der Geräte zum Gesamtensemble werden von den kanalspezifischen Konfliktlösungsstrategien berücksichtigt.

Ausgehend von der sehr grobkörnigen Basistopologie in Abb. 2 können weitere Kanäle identifiziert werden – z. B. ein „Goal“-Kanal, auf dem explizite Repräsentationen der Nutzerintention übertragen werden. Der „Goal“-Kanal zerlegt die Steuerungsapplikation in eine Dialogmanagementkomponente, die Interaktionsereignisse auf Nutzerintentionen abbildet und eine Strategiekomponente, die Intentionen in (geräteübergreifende) Aktionssequenzen übersetzt (siehe [7] für eine detailliertere Darstellung des Konzeptes der *goal-based interaction*). Eine abermals verfeinerte potentielle Pipeline-Struktur zeigt Abb. 2 ganz rechts, für eine detailliertere Diskussion dieser Topologie siehe ebenfalls [7].

Neben der Identifikation der Kanäle und kanalspezifischen Kommunikationsinhalte, die die Topologie eines Ensembles definieren, ist es auch erforderlich, die Koordinations- und Konfliktlösungsstrategien zu bestimmen, mit denen Kanäle die exklusive oder kooperative Bearbeitung von Inhalten durch Subskribenten regeln. Eine wesentliche Herausforderung ist dabei, Algorithmen zu identifizieren, die eine effiziente verteilte Implementierung zulassen um so sinnvoll eine dezentrale Realisierung zu ermöglichen.

2.3 Projektziele

Ziel von DYNAMITE ist die Entwicklung von Softwaremechanismen, die auf Basis der oben vorgestellten Konzepte von „Kanälen“ und „Ensemble-Topologien“ eine dynamische und selbsttätige Erzeugung von kohärenten Geräteverbänden ermöglichen. Dies beinhaltet die folgenden Teilziele:

- Realisierung einer unterlagerten Kommunikationsinfrastruktur, die eine dynamische, verteilte Erzeugung und Nutzung von Kanälen mit spezifischen Konfliktlösungs- und Kommunikationsstrategien ermöglicht.

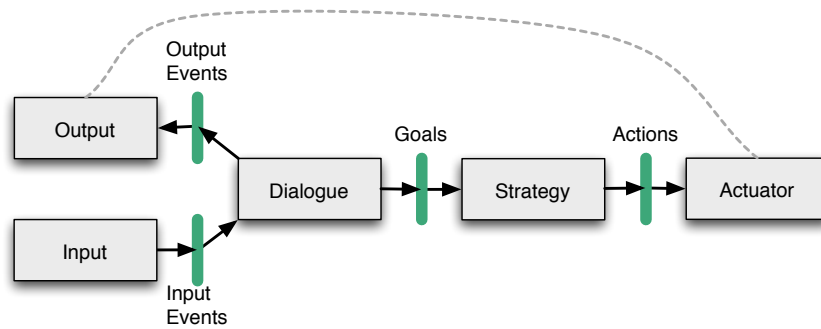


Abbildung 3: Vereinfachte Ensemble-Topologie

- Identifikation einer Standard-Topologie für Geräte-Ensembles (d.h., Bestimmung der relevanten Kanäle und ihrer Kommunikationsinhalte)
- Entwicklung von geeigneten Konfliktlösungsstrategien für die Kanäle der Standard-Topologie
- Verifikation der Nutzbarkeit der Infrastruktur durch den Einsatz in spezifischen Experimentalszenarien („Demonstratoren“)

Die entwickelte Software-Infrastruktur wird in Form einer Referenzimplementierung zur Verfügung gestellt, auf deren Basis Dritte die von DYNAMITE entwickelte Technologie nutzen können. Gleichzeitig wird im Rahmen des Projektes ein Forum etabliert, in dessen Rahmen die Anforderungen und Lösungsalternativen für selbstorganisierende Geräte-Ensembles diskutiert werden können.

3 Experimente und Technologien

3.1 Ensemble-Topologie

Basistopologie von Ensembles und kanalspezifische Konfliktlösungsstrategien sind Inhalt laufender Arbeiten im Projekt. Gegenstand der Untersuchungen ist die Definition der Kanäle der in Abb. 3 dargestellten Topologie, die den aktuellen Diskussionsstand der „Standard-Topologie“ repräsentiert und die Entwicklung potentieller Konfliktlösungsstrategien:

- **Input Events.** *Concurrent Parsing* mit „*longest parse wins*“-Strategie.
- **Goals, Actions.** „*best offer wins*“-Strategie auf der Basis eines dynamischen Kriterienkatalogs.
- **Output Events.** Capability-basierte multimodale Ausgabeplanungsstrategie [3].

Details zu diesen Strategien können [2] entnommen werden.

3.2 Die DYNAMITE Demonstratoren

Zur Untersuchung des DYNAMITE Framework ausserhalb von Simulationen auf Basis des Open Source Framework und zum Nachweis der Realisierbarkeit auf realen Hardware-Komponenten wird an der Umsetzung von zwei Demonstratorumgebungen gearbeitet. Die Umgebung



Abbildung 4: Wohnzimmer-Szenario

„Konferenzraum“ (Abb. 1) wird Funktionen wie das automatische Um/Einschalten der Präsentation des aktuellen Vortragenden unterstützen. Die Umgebung „Wohnzimmer“ (Abb. 4) bietet Dienste wie das automatische Aufzeichnen einer Sendung bei Detektion einer Unterbrechung des Nutzers (z. B. durch Telefon) und das „Follow Me“ von Applikationen / Präsentationen bei Ortswechsel des Nutzers.

Es ist hier zu beachten, dass diese Szenarien durchaus sinnvoll mit konventionellen Mitteln realisiert werden können (d.h., durch Entwicklung einer dedizierten Steuerapplikation), vgl. Abschnitt 1.2. Erst wenn man berücksichtigt, dass *dasselbe* Gerät selbständig *unterschiedliche* Rollen in verschiedenen Szenarien (Ensembles) übernehmen können muss, ergibt sich die Notwendigkeit einer neuen Form von Middleware, wie sie DYNAMITE anbietet.

3.3 Das DYNAMITE Open Source Framework

Ein wichtiges Ziel von DYNAMITE ist es, frühzeitig interessierte Dritte aus Forschung und Industrie mit den Ideen selbstorganisierender Systeme vertraut zu machen, deren Anforderungen an zukünftige Geräteumgebungen kennen zu lernen und auch eine Sensibilisierung für dieses wichtige Forschungsgebiet zu schaffen. Seit April 2004 ist daher eine erste Version des DYNAMITE-Frameworks als Open Source auf der Projekt-Webseite verfügbar, das die Kernkonzepte des Frameworks bereitstellt und die experimentelle Evaluierung des Framework erlaubt.

Die aktuelle Version des Framework bietet unter anderem:

- eine Java API zur Realisierung eigener Komponententopologien, d.h. Implementierung eigener Komponenten und Kanälen
- vier Kanalstrategien zur Konfliktlösung im Falle konkurrierender Komponenten (darunter eine Strategie für verteilte Problemlösung/Koordination von multimodalen Ausgaben)
- verschiedene Demonstratoren, die als Beispielimplementationen dienen.

Zu den Demonstratoren gehört auch ein umfangreicheres „Wohnzimmer“-Szenario, das exemplarisch die Realisierung der Selbstorganisation von komplexen, realistischen Ensembles auf der Basis des DYNAMITE Framework zeigt. Das Szenario enthält vielfältige Komponenten, wie Spracherkennung, eine graphische Bedienoberfläche, ein virtueller Charakter, eine synthetische Sprachausgabe, Steuerungsapplikationen und mehrere Aktoren wie MP3-Player und

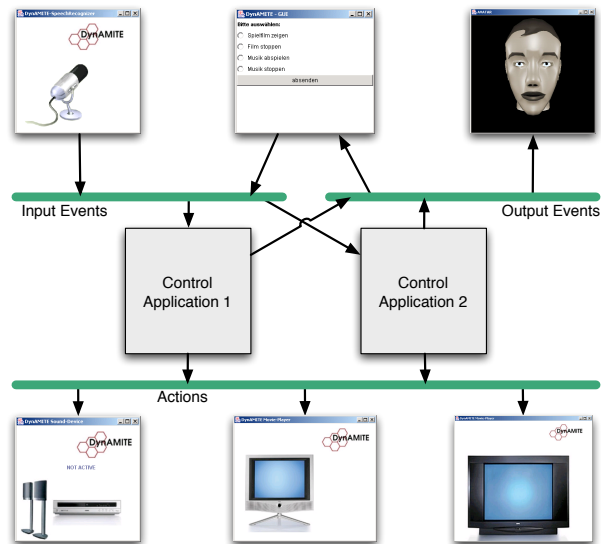


Abbildung 5: Szenario „Entertainment“ (komplettes Ensemble)

MPEG-Player; die Basistopologie aus Abb. 2 bildet hier die Grundlage. Abb. 5 zeigt die konkrete Topologie einer möglichen Instanz dieses Ensembles.

4 Ausblick

In der Zukunft werden zahlreiche Gerätesysteme insbesondere im Privathaushalt miteinander vernetzt sein. Um solche „intelligenten Umgebungen“ allerdings praktikabel zu gestalten sind flexible Plattformen notwendig, die eine Vielzahl an verschiedenen Geräten verschiedener Nutzer unterstützen können. So gehören laut [15] wiederverwendbare Plattformen und Tools für multimodale Dialogsysteme zu den wesentlichen Meilensteinen der Forschung in den nächsten drei Jahren. Aus diesem Grund birgt das hier vorgeschlagene Framework für selbstorganisierende, verteilte Systeme die Chance, verbunden mit den vorgeschlagenen Aktivitäten innerhalb der wissenschaftlichen Community einen Weg zu einem solchen Standard zu schaffen.

Wir sind sicher, dass wir mit unserer Aktivität ein großes Bedürfnis des Marktes adressieren: ohne ein standardisiertes und wiederverwendbares Softwareframework für die Selbstorganisation von Geräte-Ensembles lassen sich mittel- und langfristig die Bedürfnisse des Verbrauchers nicht erfüllen. Das Echo auf den ersten DYNAMITE-Workshop (Mai 2004) in Presse und einschlägigen Internet-Foren belegt klar das Interesse der Öffentlichkeit an diesem Thema.

Unsere Entwicklung und der Community-Prozess, den wir initiiert haben, wird dazu beitragen, die Existenz einer solchen gemeinsamen Plattform sicherzustellen. Auf Basis dieser Plattform können dann neuartige Geschäftsmodelle umgesetzt und neue Märkte eröffnet werden; gleichzeitig wird ein Bedarf an Appliances entstehen, die auf Basis einer Plattform wie DYNAMITE interoperabel sind. Die Offenheit der Plattform stellt dabei insbesondere sicher, dass alle interessierten Institutionen wirksam am Markt partizipieren können.

Der nächste Schritt im Projekt ist die Entwicklung / Verfeinerung der Modelle die den Konfliktlösungsstrategien zu Grunde liegen. Die Ergebnisse hierzu werden auf dem nächsten DYNAMITE-Workshop (geplant für November 04) vorgestellt und diskutiert werden.

Eine Bereitstellung der vollständigen DYNAMITE Middleware ist für Mitte 2006 vorgesehen.

Danksagung

Der Autor möchte sich herzlich bei den Mitgliedern des DYNAMITE-Projektteams von EML, IGD und Loewe für die tatkräftigen Unterstützung bedanken: Yun Ding, Christian Elting, Michael Hellenschmidt, Rainer Malaka, Ali Nazari, Jens Neumann, Sybille Peters, Dennis Pfisterer, Thomas Schmitz.

Literatur

- [1] Davies, N., Gellersen, H.-W. Beyond Prototypes: Challenges in Deploying Ubiquitous Systems. *IEEE Pervasive Computing*, January–March 2002.
- [2] DYNAMITE – Dynamisch Adaptive Multimodale IT Ensembles. www.dynamite-project.org, 2003.
- [3] Elting, C., Möhler, G. Modeling output in the embassi multimodal dialog system. In *Proceedings 4th IEEE International Conference on Multimodal Interfaces*, Pittsburgh, USA, October 14–16 2002.
- [4] EMBASSI – Elektronische Multimediale Bedien- und Service-Assistenz (BMB+F Leitprojekt). www.embassi.de, 1999.
- [5] FhG-IPSI et al. Roomware. www.roomware.de, 2001.
- [6] HAVi, Inc. The HAVi Specification – Specification of the Home Audio/Video Interoperability (HAVi) Architecture – Version 1.1. www.havi.org, May 2001.
- [7] Heider, T., Kirste, T. Supporting goal-based interaction with dynamic intelligent environments. In *Proc. 15th European Conference on Artificial Intelligence (ECAI'2002)*, Lyon, France, July 22–26 2002.
- [8] MIT Laboratory for Computer Science. MIT Project Oxygen – Pervasive, Human-Centered Computing. www.oxygen.lcs.mit.edu, 2000.
- [9] MITRE corporation. Galaxy communicator. <http://communicator.sourceforge.net/>, 2001.
- [10] Shafer, S. et al. Microsoft Project “EasyLiving”. <http://research.microsoft.com/easyliving/>, 1999.
- [11] SRI International AI Center. The Open Agent Architecture. <http://www.ai.sri.com/~oaa/>, 2000.
- [12] Sun Microsystems, Inc. Jini Technology Core Platform Specification – Version 1.1. www.jini.org, October 2000.
- [13] University, S. Stanford interactive workspaces project. graphics.stanford.edu/projects/iwork/, 2000.
- [14] Universal plug-and-play. www.upnp.org.
- [15] Wahlster, W. Research Roadmap of Multimodality 2002-2005, 2001. Dagstuhl Seminar 2001 on Coordination and Fusion in Multimodal Interaction, http://www.dfki.de/~wahlster/Dagstuhl_Multi_Modality.
- [16] Wahlster, W., Reithinger, N., Blocher, A. Smartkom – multimodal communication with a life-like character. In *Proc. Eurospeech 2001*, Aalborg, Denmark, 2001.