

---

## DIANE: Dienste in Ad-hoc-Netzen

Teilprojekt des DFG Schwerpunktprogramms 1140

“Basissoftware für selbstorganisierende Infrastrukturen für vernetzte mobile Systeme”



---

Zwischenbericht Februar 2004

## Effektive und effiziente Dienstsuche und -nutzung in Ad-hoc-Netzen

Dr. Birgitta König-Ries  
Dipl.-Inform. Philipp Obreiter  
Dipl.-Inform. Michael Klein

Universität Karlsruhe  
Institut für Programmstrukturen und Datenorganisation  
Am Fasanengarten 5  
D-76131 Karlsruhe



## Kurzfassung

Der folgende Bericht beschreibt die im Projekt DIANE (Dienste in Ad-hoc-Netzen) in den vergangenen anderthalb Jahren im Rahmen der Förderung als Teilprojekt des SPP 1140 erzielten Ergebnisse. Das Ziel des DIANE-Projektes ist es, die in Ad-hoc-Netzen verteilt vorliegenden Ressourcen integriert nutzbar zu machen. Basierend auf dem dienstorientierten Paradigma gehen die Arbeiten davon aus, dass Ressourcen in Form von Diensten verfügbar gemacht werden. Dabei werden Mechanismen benötigt, um diese Dienste *effektiv* und *effizient* nutzen zu können.

Nach einer Analyse der in Ad-hoc-Netzen relevanten Dienste wurde die Entscheidung getroffen, in der ersten Projektphase ausschließlich Basisdienste zu betrachten. Basisdienste sind Dienste, die grundlegende Funktionalität bereitstellen. Sie sind nicht aus Teildiensten zusammengesetzt, können vollständig automatisch, d.h. ohne Benutzerinteraktion ausgeführt werden und basieren auf einem klassischen Client-Server-Kooperationsmodell. Basisdienste sind die Bausteine aus denen komplexe Dienste kombiniert werden – für eine sinnvolle Nutzung von Ad-hoc-Netzen sind also Möglichkeiten zur effektiven, effizienten und automatischen Suche nach und Nutzung von Basisdiensten ein unabdingbarer erster Schritt. Für diese Klasse von Diensten wurden konkret folgende Ergebnisse erzielt:

- *Beschreibung von Dienstangeboten.* Damit Dienste automatisch gefunden werden können, müssen sie beschrieben sein. Wir haben hierfür eine auf geschichteten Ontologien beruhende Angebotsbeschreibungssprache, die *DIANE Service Offer Description (DSOD)*, entwickelt. Zwei wesentliche Charakteristika dieser Sprache, die sie gegenüber existierenden Ansätzen auszeichnen, sind ihre reine Zustandsorientierung und ihre Anreicherung mit Variablen ([KK02, KK03, KKO03d]).
- *Beschreibung von Dienstanfragen.* Damit nach Diensten gesucht werden kann, muss eine Sprache zur Formulierung von Dienstanfragen bereit gestellt werden. Soll diese Suche automatisch erfolgen, muss diese Sprache es ermöglichen, Anfragen sehr präzise zu formulieren und in der Anfrage Auswahlkriterien für Dienste eindeutig zu hinterlegen. Beide Forderungen werden von existierenden Ansätzen nicht erfüllt. Wir entwickelten daher die *DIANE Service Request Description (DSRD)* vor, die gemeinsam mit der DSOD eine automatische Auswahl und Konfiguration von Diensten erlaubt ([KK04]).
- *(Halb-)semantische Overlays zur Dienstvermittlung.* Damit Dienstnachfragen und -angebote abgeglichen werden können, müssen sie zunächst zueinander gebracht werden. Gängige Ansätze, wie die Einrichtung von Verzeichnisdiensten, das Fluten des Netzes oder auch hashbasierte Ansätze sind für die semantische Dienstsuche in Ad-hoc-Netzen nicht geeignet. Benötigt wird eine logische Struktur oberhalb des Netzwerkes, mit deren Hilfe Dienstsuchen und -anfragen im Netz verteilt werden können, und die gleichzeitig gut an das sich dynamisch verändernde Netz angepasst bleibt. Wir haben hierzu mehrere semantische (Ringe, Cluster) und halbsemantische Overlays (Mesh, Lanes) entwickelt. Als am besten für Ad-hoc-Netze geeignet hat sich die Lanes-Struktur herausgestellt ([KK02b, KKO03c, KKO03b]).
- *Diensterbringung.* Um einen gefundenen Dienst nutzen zu können, muss dieser zunächst konfiguriert, dann angestoßen werden. Hierzu wurde ein spezielles *Konfigurations- und Ausführungsprotokoll* entwickelt. Die konkrete Einbindung des Dienstes in die DIANE-Struktur wird durch einen *Containermanager* vorgenommen, in dem der Dienst abläuft.
- *Anreizmechanismen.* Die an Ad-hoc-Netzen teilnehmenden Geräte sind autonom und haben daher keine inhärente Motivation, Dienste zu erbringen oder sich an der Dienstvermittlung zu beteiligen. Wir haben gezeigt, welche Arten von Anreizen gegeben werden können, um dennoch kooperatives Verhalten zu motivieren. Basierend darauf haben wir Anreizschemata entwickelt, die für eine faire Nutzung der Ressourcen sorgen. Dabei kommt ein selbstentwickeltes verteiltes Reputationssystem zum Einsatz ([OKK03b, ON03, FO04, OKP04, Obr04]).
- *Implementierung und Evaluierung.* Da momentan noch keine Multihop-Ad-hoc-Netze vorhanden sind, die man als Basis zur Evaluierung der entwickelten Protokolle und Mechanismen nutzen könnte, haben wir uns für eine Implementierung in einem Simulator entschieden. Es stellte sich heraus, dass existierende Produkte die für uns interessanten oberen Schichten des ISO/OSI-Modells nur ungenügend berücksichtigen. Wir haben daher unseren eigenen Simulator, DIANE<sub>mu</sub>, entwickelt und darin die oben genannten Komponenten implementiert. Besondere Charakteristika sind eine komfortable Möglichkeit zur Generierung realitätsnaher Benutzermodelle, die Bewegung und Dienstnutzung sinnvoll koppeln, und eine Softwarearchitektur, die eine direkte Übertragbarkeit der entwickelten Protokolle auf reale Geräte ermöglicht ([Kle03]).



## Anmerkung

Der folgende Zwischenbericht des Projektes DIANE (Dienste in Ad-hoc Netzen) wurde nach einer Laufzeit von 17 Monaten (Oktober 2002 bis Februar 2004), bei einer bewilligten Projektdauer von 24 Monaten (Oktober 2002 bis September 2004), angefertigt. Anlass für die Erstellung ist die Einreichung eines Folgeantrags durch die Antragstellerin.

Ausgewählte Veröffentlichungen aus dem Projekt wurden dem Fortsetzungsantrag als Anlage beigelegt; sämtliche Veröffentlichungen sowie die entstandenen Studien- und Diplomarbeiten finden sich in elektronischer Form auf der beiliegenden CD.

## Einleitung

In den letzten Jahren wurden erste technische Grundlagen für die Bildung von Multihop-Ad-hoc-Netzen, dynamischen Verbänden mobiler Rechner ohne feste Infrastruktur, geschaffen. Für Ad-hoc-Netze gilt jedoch, genauso wie für Rechnernetze in Festnetzen, dass eine rein technische Vernetzung alleine noch keine sinnvolle gemeinsame Nutzung der verbundenen Ressourcen ermöglicht. Während für Rechner in infrastrukturgestützten Netzen geeignete Konzepte und Mechanismen zu dieser integrierten Nutzung existieren, fehlten diese im jungen Forschungsgebiet Ad-hoc-Netze bislang.

Das Projekt DIANE hat sich in den letzten anderthalb Jahren mit der Entwicklung und Evaluierung von Konzepten zur effizienten und effektiven Nutzung der in einem Ad-hoc-Netz bereitgestellten Ressourcen befasst. Basierend auf dem dienstorientierten Paradigma gehen die Arbeiten davon aus, dass Ressourcen in Form von Diensten zur Verfügung werden. Mechanismen zur *effektiven* Nutzung von Ressourcen stellen sicher, dass Ressourcen bereitgestellt werden und die Ressourcennutzung das gewünschte Ergebnis liefert; konkret heißt das, dass Dienste angeboten werden und dass diejenigen Dienste gefunden und genutzt werden, die die geforderte Leistung erbringen. Mechanismen zur *effizienten* Nutzung von Ressourcen stellen sicher, dass die gewünschten Ergebnisse ressourcenschonend, also mit möglichst geringem Aufwand, erzielt werden.

Um das Ziel der effizienten und effektiven Nutzung von Ressourcen zu erreichen, schlagen wir Mechanismen zur Beschreibung, Auffindung und motivierten Bereitstellung von Diensten vor.

Im Folgenden wird nun auf die Ergebnisse zu den Arbeitspunkten (AP) aus dem Arbeitsplan des Antrags zur ersten SPP-Phase eingegangen und der momentane Bearbeitungsstand geschildert.

## AP 1: Ist-Analyse, Soll-Analyse

Ziel von Arbeitspunkt 1 war es, zunächst zu ermitteln, auf welcher technischen Basis DIANE aufsetzen kann (AP 1.1) und welche Anforderungen an die Funktionalität eines Systems zur Dienstsuche und -nutzung in Ad-hoc-Netzen bestehen (AP 1.2). Darauf aufbauend sollte in AP 1.3 dann ein Lastenheft für die weitere Entwicklung erstellt werden.

### AP 1.1: Analyse der Ausstattung

Im Februar 2002 wurde eine Online-Befragung von Karlsruher Informatikstudierenden durchgeführt<sup>1</sup>. An dieser Umfrage beteiligten sich ca. 200 Studierende. Im ersten Teil des Fragebogens wurde gefragt, welche mobilen Geräte die Studierenden schon besitzen bzw. in nächster Zeit anschaffen wollen. Es zeigte sich, dass zumindest in der befragten Zielgruppe eine hohe Durchdringung mit netzwerkfähigen mobilen Geräten gegeben ist (ca. 75%). Dominierend sind dabei Laptops mit W-LAN-Karten, PDAs spielen eine eher untergeordnete Rolle. Ebenfalls weit verbreitet waren

---

<sup>1</sup>Details sowie Ergebnisse der Umfrage finden sich auf der beiliegenden CD.

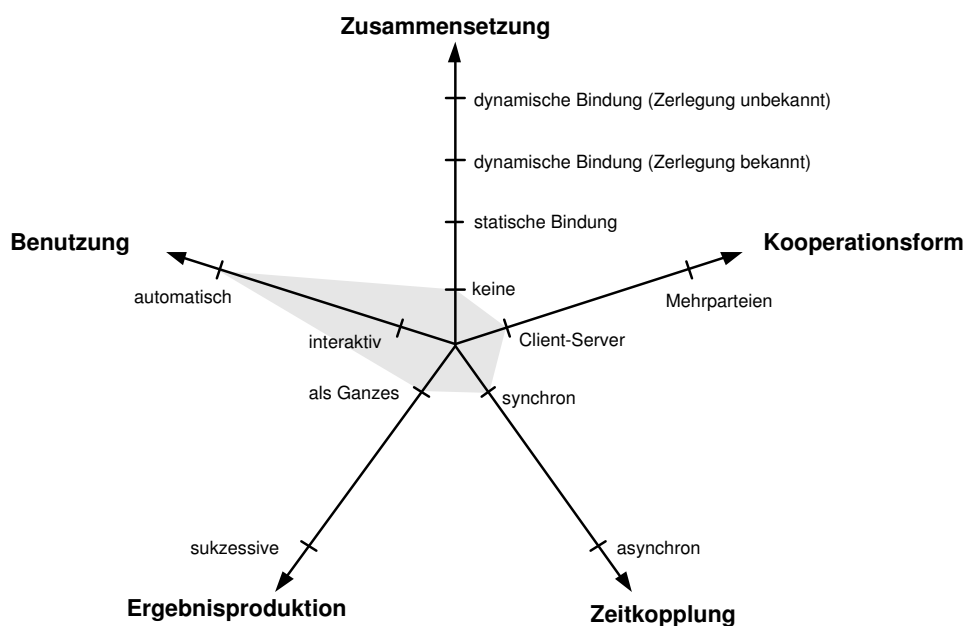


Abbildung 1: Charakterisierung der Dienste entlang fünf Dimensionen. Grauer Bereich: Basisdienste

WAP-fähige Handys, zum Befragungszeitpunkt besaß aber nur eine verschwindend kleine Minderheit der Befragten Java-fähige Handys und es wurde kein Interesse an Beschaffungen solcher Geräte geäußert. Unsere Schlussfolgerung aus diesen Ergebnissen ist, dass zumindest zum jetzigen Zeitpunkt Anwendungen auf die Verwendung mit einem Laptop ausgerichtet sein sollten. Betrachtet man aber generell die technische Entwicklung, so ist zu erwarten, dass Smartphones in der Zukunft wohl das Kommunikationsmittel der Wahl sein werden – auch, wenn sich dies in unseren Umfrageergebnissen noch nicht widerspiegelt.

Bei unseren Anstrengungen zum Aufbau von "echten" Ad-hoc-Netzen zeigte sich schnell, dass hierzu von den unteren Netzwerkschichten noch nicht ausreichend ausgereifte Techniken zur Verfügung gestellt werden. Konsequenterweise entschieden wir uns, zumindest vorläufig auf eine weit reichende Evaluierung im Realbetrieb zu verzichten und – wo immer möglich – stattdessen eine Simulationsumgebung zu verwenden.

## AP 1.2: Analyse der Wünsche

Im Rahmen der oben schon erwähnten Umfrage wurde auch nachgefragt, welche Dienste Studierende in einem Ad-hoc-Netz für sinnvoll halten und verwenden würden. Hier wurde die Nützlichkeit sämtlicher existierender Dienste zur Lernunterstützung (wie Lernserver, Online-Verfügbarkeit von Vorlesungsmaterialien, Mailinglisten usw.) in einem Ad-hoc-Netz als höher eingestuft als dies jetzt der Fall ist.

Besonders stark nachgefragt wurden Informationsdienste, die Zusatzinformationen über Lehrveranstaltungen etc. zugänglich machen, sowie komplexe Gruppendienste (virtuelle Arbeitsgruppen, semi-automatische Gruppenbildung usw.) und Dienste zur Erledigung administrativer Tätigkeiten (Rückmeldung, Prüfungsanmeldung usw.).

Bei genauerer Betrachtung der gewünschten Dienste stellt sich heraus, dass diese ein breites Spektrum umfassen und sich zum Teil erheblich voneinander unterscheiden. Wir klassifizierten die Dienste daher entlang fünf Dimensionen (siehe Abbildung 1):

- **Zusammensetzung:** Ein Dienst kann so aufgebaut sein, dass er für die Erbringung seiner Leistung *keine* weiteren Dienstgeber benötigt. Ist er zusammengesetzt, so kann er die benötigten Teildienstgeber *statisch* also fix

einbinden oder *dynamisch* nach geeigneten Dienstgebern suchen, die unter den aktuellen Gegebenheiten optimal sind. Im Falle der dynamischen Bindung kann die Zerlegung des Gesamtdienstes in Teildienste *bekannt* oder *unbekannt* sein.

- **Kooperationsform:** Bei *Client-Server-Diensten* kooperieren genau zwei Partner in den Rollen Dienstnehmer und Dienstgeber. Bei *Mehrparteien-Diensten* kooperieren mehr als zwei Partner, deren Rollenaufteilung in Dienstbringer und -nutzer nicht mehr eindeutig vorgenommen werden kann.
- **Zeitkopplung:** Bei *synchronen* Diensten besteht eine zeitliche Kopplung zwischen der Anstoßung des Dienstes und dessen Ergebnislieferung, d.h. der Dienstnehmer muss explizit auf das Eintreffen des Ergebnisses warten. *Asynchrone* Dienste entkoppeln diese Aktionen, so dass der Dienstnehmer vom Dienstgeber informiert wird, wenn das Ergebnis vorliegt.
- **Ergebnisproduktion:** Ein Dienst kann seine Dienstleistung in einzeln nutzbaren Teilen erbringen, indem er die Ergebnisse *sukzessive* bereitstellt. Andernfalls liefert der Dienst seine Leistung *als Ganzes*.
- **Benutzung:** *Interaktiv* zu nutzende Dienste sind so implementiert, dass sie von einem menschlichen Benutzer über eine graphische Schnittstelle konfiguriert und ausgeführt werden müssen. *Automatisch* nutzbare Dienste sind so beschaffen, dass sie vollständig von einem nicht-menschlichen Agenten verwendet werden können.

### AP 1.3: Erstellung des Lastenhefts

Die Analyse der Wünsche in AP 1.2 zeigte, dass Benutzer an einer Fülle sehr unterschiedlicher Dienste interessiert sind. Um die zu bearbeitende Aufgabe handhabbar zu halten, entschieden wir uns bei der Erstellung des Lastenhefts, zunächst nur solche Dienste zu betrachten, die zur Erbringung der grundlegenden Funktionalität in einem Ad-hoc-Netz benötigt werden. Dies sind insbesondere die Dienste, die die in einem Ad-hoc-Netz vorhandenen Ressourcen für alle Teilnehmer effizient und effektiv nutzbar machen, und zwar in einer möglichst transparenten Art und Weise. Bezeichnend für solche **Basisdienste** ist zum einen ihre Einfachheit: d.h. ihre klassische Client-Server-Kooperationsform, ihre Nicht-Zusammengesetztheit, ihre Synchronität und ihre Ergebnisproduktion als Ganzes. Andererseits müssen sie jedoch für menschliche Netzteilnehmer quasi unsichtbar zur Verfügung stehen, um so die Nützlichkeit des Ad-hoc-Netzes zu erhöhen. Eine wichtige Anforderung war daher die *automatische Nutzbarkeit*. Die Klasse dieser Basisdienste ist in Abbildung 1 grau hinterlegt.

Ausgehend von der Entscheidung, uns bei der Entwicklung von Techniken auf die Menge der Basisdienste zu konzentrieren, fassten wir im Lastenheft folgende grundsätzlichen Anforderungen an die zu modellierenden Lösungen in Arbeitspunkt 2 zusammen:

- *AP 2.1: Dienstbeschreibung.* Diese muss so beschaffen sein, dass sie automatisch verarbeitet werden kann, d.h. der Rechner muss deren Inhalt eindeutig verstehen, um auf dieser Basis sinnvolle Entscheidungen treffen zu können.
- *AP 2.2: Benutzerkontext.* Auch dieser muss automatisiert nutzbar sein.
- *AP 2.3: Dienstvermittlung.* Diese muss so funktionieren, dass sie (neben hoher Effizienz und Effektivität) mit Basisdiensten und deren semantischen Beschreibungen aus AP 2.1 umgehen kann.
- *AP 2.4: Dienstbringung.* Auch diese muss automatisiert durchführbar sein, d.h. die Vorgehensweise zur Konfigurierung und zur Anstoßung des Dienstes muss für einen Rechner selbstständig durchführbar sein. Da das Bereitstellen von Ressourcen als Dienste nicht automatisiert ablaufen wird (zumindest die Einwilligung des Dienstgebers wird nötig sein), sollte das Anbieten von Diensten für den Dienstgeber möglichst einfach sein.
- *AP 2.5: Motivation.* Diese muss Anreize für die Bereitstellung von Basisdiensten und für die Teilnahme an der Dienstvermittlung aus AP 2.3 bereitstellen. Die Anreize müssen automatisiert wirken und keiner Interaktion mit dem Benutzer bedürfen.

## AP 2: Modellierung

### AP 2.1: Dienstbeschreibung

Das Erstellen einer geeigneten Dienstbeschreibungssprache stellte sich als komplexer heraus als zunächst erwartet. Dies lag vor allem an den mehrschichtigen und zum Teil gegensätzlichen Anforderungen, die aus den Zielen des Projekts abgeleitet werden konnten:

- Für *Benutzer* sollen Beschreibungen in der Sprache *leicht erstellbar* und *universal* für alle möglichen Dienste geeignet sein.
- *Rechner* sollen Beschreibungen in der Sprache *automatisch verarbeiten* können, d.h. für sie muss die Beschreibung *semantisch eindeutig* sein und einen klar definierten *Bezug zur realen Welt* besitzen.

Als Nebenbedingung musste beachtet werden, dass die Sprache vorwiegend in mobilen Umgebungen zum Einsatz kommen soll, d.h. sie muss auch auf Geräten mit beschränkten Ein-/Ausgabemöglichkeiten, geringem Speichervermögen und Rechenkapazität verwendbar sein.

Zunächst wurden existierende Sprachen untersucht. Es stellte sich heraus, dass eine sinnvolle automatische Verarbeitung nur dann möglich ist, wenn zur Erstellung der Beschreibung ein gemeinsames Vokabular eingesetzt wird. Dies bieten nur ontologiebasierte Sprachen. Geeignet erschien insbesondere DAML-S [DAMLS], eine Sprache zur Erstellung von Dienstbeschreibungen unter Verwendung hochexpressiver Ontologien in DAML [DAML]. Zur Zeit wird DAML vom W3C als Web Ontology Language (OWL) [OWL] standardisiert, entsprechend gibt es OWL-S.

Die direkte Verwendung von DAML-S erwies sich als problematisch. Zwar ist die Sprache sehr flexibel und universell einsetzbar, dadurch jedoch auch zu allgemein gehalten und nicht direkt für konkrete Beschreibungen der betrachteten Dienste einsetzbar. In [KK03] entwickelten wir daher einen Prozess, mit dessen Hilfe ein Benutzer konkrete DAML-S-Beschreibungen erstellen kann. Die erzeugten Beschreibungen zeichnen sich einerseits durch eine hohe Strukturierung aus, sie sind andererseits aber nach wie vor flexibel und universal verwendbar. Möglich wurde dies durch das Prinzip der *Ontologieschichtung*: Beginnend mit einer oberen Dienstontologie, die wie DAML-S den grundsätzlichen Aufbau einer Beschreibung festlegt, wählt der Benutzer anschließend eine Ontologie für die Dienstkategorie, welche die Art der Aktionen des Dienstes (wie beispielsweise Dokumente verarbeiten, Wissen beschaffen, Handlungen in der realen Welt durchführen etc.) beschreibt. Diese können dann durch Konzepte aus Ontologien mit spezialisiertem Domänen-vokabular konkretisiert werden. Eine wichtige Kategorie von Diensten stellen *Informationsdienste* dar. Dienste dieser Art verändern den Zustand von Dokumenten und wurden in [KK02, KK03] genauer betrachtet.

Dienstbeschreibungen mit gemeinsamem Vokabular und hoher Strukturierung stellen zwar einen ersten Schritt zur automatischen Verarbeitung dar, sind jedoch häufig semantisch nicht eindeutig. Hauptursache hierfür ist, dass Dienste konfigurierbar sind. Hierin unterscheiden sie sich von anderen Ressourcen. Existierende Sprachen drücken diese Möglichkeiten der Konfigurierung durch die Angabe von eingehenden und abgehenden Nachrichten aus. Diese werden jedoch getrennt von der Wirkung des Dienstes aufgelistet, so dass für einen Rechner nicht ersichtlich ist, wie sich das Senden dieser Nachrichten auf die Leistung des Dienstes auswirkt. In [KKO03d] schlugen wir daher eine Verbindung der beiden Teile zu einer *rein zustandsorientierten* Beschreibung mit *Variablen* vor. Hier wird ein Dienst nur noch durch den Zustand vor und nach erfolgreicher Dienstauführung beschrieben, wobei die Konfigurationsmöglichkeiten dieser Zustände durch Einsetzen von Variablen klar ersichtlich werden. Die Klassifizierung der Variablen in *Variablenkategorien* ermöglicht es weiterhin, die Ausfüllreihenfolge der Variablen in die Beschreibung zu integrieren und so den Aushandlungsprozess um iterative Schätzungsphasen zu erweitern, die der eigentlichen Ausführungsphase vorausgehen können.

Neben der Eindeutigkeit der Semantik im Bezug auf die Konfigurierung der Beschreibung muss auch die Semantik des Vergleichs zwischen Dienstanfrage und Dienstangeboten für den Rechner eindeutig ersichtlich sein. Existierende Ansätze gehen davon aus, dass ein solcher Vergleich von einem generellen Allzweckvergleicher durchgeführt werden

kann, in den einerseits die Beschreibungen der angebotenen Dienste, andererseits die gleich aufgebaute, beispielhafte Beschreibung des benötigten Dienstes eingehen. Der Vergleichler bestimmt dann mittels eingebauter Ähnlichkeitsmaße eine Liste möglicher Dienste. Dieser Ansatz hat jedoch das Problem, dass der Vergleichler nicht weiß, welche Abweichungen vom Wunschkdienst für den Anfrager noch akzeptabel sind und er mitunter eigene Annahmen treffen muss, da er die genauen Präferenzen des Nutzers nicht kennt. Diese können nur dann ausgedrückt werden, wenn dem Benutzer eine spezielle Anfragebeschreibungssprache zur Verfügung steht, die solche Präferenzen genau erfassen kann. In [KK04] wurde daher eine solche *präferenzbeinhaltende Sprache* vorgeschlagen, welche Anfragen durch einen wohldefinierten Aufbau aus unscharfen Objektmengen erstellt. Neu ist, dass aus einer solchen Beschreibung ein spezieller, persönlicher Vergleichler abgeleitet werden kann, der ohne zusätzliche externe Annahmen den Vergleich exakt und automatisch nach den Wünschen des Anfragers durchführen kann.

Zur Zeit findet die Formalisierung der Angebots- und Anfragebeschreibungssprache in Form der DIANE Service Offer bzw. Request Description (DSOD bzw. DSRD) [HSS04] statt. Um sowohl die Anforderungen menschlicher Benutzer als auch die von Rechnern zu erfüllen, wird die Syntax der Sprache in einer formalen *Basisrepräsentation* hinterlegt. Diese orientiert sich an Schreibweisen aus F-Logic [KLW95] sowie der Objektorientierung. Ziel dieser Basisrepräsentation ist die formale Vollständigkeit und Eindeutigkeit der Sprache – diese Form ist jedoch nicht für die direkte Verarbeitung durch Mensch oder Maschine bestimmt. Hierfür kommen Transformatoren zu Einsatz, die sie in geeignetere Darstellungsformen, so genannte *Nebenrepräsentationen*, überführen: Für *menschliche Benutzer* steht eine graphische, UML-ähnliche Nebenrepräsentation zur Verfügung (siehe [KK03, KKO03d]) sowie ein visuelles Tool, welches den Benutzer schrittweise durch den Erstellungsprozess einer solchen Beschreibung leitet. [Zhe03]. Für *Rechner* steht zur Durchführung der Konfiguration und des Vergleichs eine Nebenrepräsentation in Form objektorientierten Java-Programmcodes zur Verfügung. Hierdurch kann die Beschreibung leicht in anderen Programmen verwendet werden und ist serialisier- und kompilierbar. So repräsentierte Anfragebeschreibungen können durch Hinzufügen von Hilfscode in einen persönlichen Vergleichler übersetzt werden. *Externe Nutzer* können auf die Beschreibungen in Form einer OWL-Nebenrepräsentation zugreifen, um so mit dem eigenen System kompatibel zu sein.

Bis zum Ende der Projektlaufzeit soll die Formalisierung der verschiedenen Repräsentationsformen sowie die Möglichkeit zur Generierung des Vergleichlers abgeschlossen werden.

## AP 2.2: Benutzerkontext

Da das Beschreiben und Verwalten von Benutzerkontexten Kernpunkt der SPP-Teilprojekte in Frankfurt und Marburg ([ELAN, MARB]) ist, wurde auf eine eigene Bearbeitung dieses Punktes zugunsten der anderen Aufgaben verzichtet. Die im Projekt entwickelte Anfragesprache (die DIANE Service Request Language, siehe AP 2.1) bietet die Möglichkeit die aus dem Kontext resultierenden Präferenzen exakt anzugeben, so dass diese automatisch von einem generierten Vergleichler berücksichtigt werden. Somit ist die Möglichkeit gegeben, die andernorts entwickelten Mechanismen in DIANE zu nutzen (siehe auch AP 5.1).

## AP 2.3: Dienstvermittlung

Die effiziente Dienstvermittlung in Ad-hoc-Netzen stellt ein komplexes Problem dar. Zunächst wurden daher existierende Verfahren für Infrastrukturnetze betrachtet und auf ihre Eignung überprüft. Diese können generell in drei Gruppen eingeteilt werden: Ansätze mit *zentralem Dienstverzeichnis* scheiden aufgrund der hohen Dynamik des Netzes aus, Ansätze mit *Anfrageflutung* verbieten sich aufgrund ihres verschwenderischen Umgangs mit Bandbreite und Energie, die in den meisten mobilen Geräten nicht unbegrenzt zur Verfügung stehen, *hashbasierte Ansätze* unterstützen hingegen nur semantisch schwache Dienstbeschreibungen und sind daher für die komplexen, ontologiebasierten Beschreibungen nicht geeignet. Details dieser Untersuchung finden sich in [KKO03a].

Wir folgerten, dass das Problem der betrachteten Verfahren darin besteht, dass sie nur unzureichend zwischen der vom Benutzer erwarteten Funktionalität (also effizienter, semantischer Dienstsuche) und den Netzwerkeigenschaften

(also hochdynamische Topologie und leistungsschwache Geräte) vermitteln. Es erschien daher sinnvoll, zur Überbrückung dieser Lücke ein *Overlay*, eine logische Netzstruktur, zwischen Benutzer und Netzwerk aufzubauen. Eine solche Struktur bringt dann wesentliche Vorteile, wenn sie zum einen die Dienstvermittlung (also Ankündigung und Suche nach Diensten mit Hilfe von Dienstbeschreibungen) optimal unterstützt und sich andererseits effizient an die ständig verändernde Netzwerktopologie anpasst. Diese Anforderungen sind konkurrierend: Die Anwendung verlangt nach stark strukturierten und globalen Strukturen mit vielen nutzbaren Overlayverbindungen, während aus Sicht des Netzes wenig strukturierte und lokale Strukturen mit möglich wenig zu wartenden Overlayverbindungen günstig sind. Daher wurden verschieden stark strukturierte Overlaystrukturen konzipiert, zugehörige Algorithmen entwickelt und untersucht, wie sich diese zur Dienstvermittlung in Ad-hoc-Netzen eignen.

Eine Gruppe stark strukturierter Ansätze stellen *semantische Overlays* dar. Bei diesen wird einerseits Wissen, welche Dienste angeboten werden, zur Definition der Struktur verwendet, andererseits Dienstsuchnachrichten abhängig von ihrem Inhalt gezielt durch die Struktur geleitet. In diese Gruppe fallen die von uns entwickelten *mehrschichtigen Cluster* [KK02b] sowie *Dienstringe* [Koo02, Pon02, KKO03c]. Ein Cluster ist dabei eine Sammlung von Geräten, die ähnliche Dienste anbieten und sich gleichzeitig in räumlicher Nähe zueinander befinden. Diese können dann selbst wieder zu Clustern höherer Ordnung gruppiert werden, indem die Bedingungen nach lokaler und semantischer Ähnlichkeit abgeschwächt werden. Im zweiten Verfahren werden Geräte zu logischen Ringen verbunden, wenn sie ähnliche Dienste anbieten. Ringe haben dabei ausgezeichnete Zugangspunkte, die selbst wieder Ringe bilden können. Es stellte sich heraus, dass sich diese komplexen Strukturen nur schwer in dynamischen Topologien warten lassen. Gerade die Hierarchisierung brachte Probleme, da lokale Anpassungen an der Struktur oft auch rekursiv an höhere Ebenen weitergeleitet werden mussten.

Im Folgenden konzentrierten wir uns daher auf schwächer strukturierte Ansätze in Form *halbsemantischer Overlays*. Bei diesen geht die Information über angebotene Dienste nicht mehr in die Definition des Overlays ein, allerdings werden die Dienstsuchnachrichten abhängig von ihrem Typ durch die Struktur geleitet. Entwickelt wurden im Wesentlichen zwei Ansätze: *Mesh*, bei dem ein zweidimensionales Gitter von Geräten aufgebaut wird, und *Lanes* [KKO03b]. Bei Lanes wurde speziell auf eine schlanke, gut anpassbare Struktur Wert gelegt. Dazu werden in Lanes Geräte zu parallelen, logischen Bahnen verbunden. Innerhalb einer Bahn gehen die Geräte eine feste Nachbarschaftsbeziehung ein, außerhalb der Bahnen existiert diese nicht – hier werden Nachrichten per Anycast übertragen. Durch diesen Aufbau sind zwei unterschiedliche Achsen definiert: eine proaktive Achse, die zum Weiterleitung der zu persistierenden Angebotsbeschreibungen verwendet werden kann, sowie eine reaktive Achse, die zur Verteilung der spontanen Anfragebeschreibungen dient. Zur Wartung der Struktur wurden eine Reihe von Algorithmen entwickelt [KKO03a], die alle als *topologiebewusst* angesehen werden können, d.h. sie reparieren und optimieren die Struktur so, dass sie stets zur Topologie des realen Netzes passt. Zudem ist die Struktur *selbstjustierend*, d.h. ihre Parameter wie beispielsweise die Länge einer Bahn werden ständig an die aktuellen Gegebenheiten im Netz angepasst. Insgesamt ergibt sich daher eine sich automatisch einstellende Zwischenstufe von loser und strenger Struktur.

Die konzeptionelle Arbeit an Dienstvermittlungs-Overlays ist abgeschlossen und wird zur Zeit evaluiert (siehe AP 3.2). Die grundlegende Anforderung zur Unterstützung der Vermittlung von Basisdiensten ist erfüllt.

## AP 2.4: Diensterbringung

In Arbeitspunkt 1.3 wurde der Fokus des Projekts auf die automatisierte Nutzung von Diensten festgelegt, d.h. wir betrachten Dienste, die ohne Interaktion des Nutzers über eine GUI ablaufen können. Weiterhin gingen wir zunächst von traditionellen Client-Server-Diensten aus, die nicht zusammengesetzt waren und ihre Ergebnisse en bloc liefern. Hauptaugenmerk dieses Arbeitspunktes lag daher auf Untersuchungen, wie man die Nutzung solcher Basisdienste für nicht-menschliche Anfrager automatisierbar machen kann.

Als erste wichtige Voraussetzung erkannten wir, dass der nicht-menschliche Anfrager die Vorgehensweise zum Aufruf des Dienstes eindeutig aus der Beschreibung ableiten können muss. Dabei stellte sich heraus, dass der eigentlichen Ausführungsphase noch Parametrisierungen und Aushandlungen in Form einer Konfigurationsphase vorangestellt werden können, in der Dienstgeber und Dienstnehmer konkretere, detailliertere oder aktuelle Werte zur Dienstleistung

austauschen. Wie bereits in AP 2.2 erwähnt ist die Semantik dieser beiden Phasen komplett in der entwickelten variablenbehafteten Dienstbeschreibung integriert und somit klar für den Dienstnehmer ersichtlich [KKO03d]. Somit konnten wir ein generisches *Konfigurations- und Ausführungsprotokoll (KA-Protokoll)* entwickeln, welches rein auf Grundlage der Dienstbeschreibung die Kommunikation zwischen Dienstnehmer und Dienstgeber in diesen beiden Phasen durchführt [Fis04].

Dennoch bestand auf Seiten des Dienstgebers nach wie vor eine Lücke: zum einen lagen die Dienste, die er anbieten wollte in Form von Programmen oder Modulen vor, die über eine Reihe von gewöhnlichen Funktions- oder Methodenaufrufe anzusprechen waren, andererseits mussten diese Funktionen, wenn sie als öffentliche Dienste bereitgestellt werden sollten, die Kommunikation des oben erwähnten KA-Protokolls verwenden. Eine Technologie, die dem Dienstgeber Unterstützung bei der Überwindung dieser Lücke bietet, sind *Containermanager*. Im Folgenden konzipierten wir einen solchen Containermanager [Fis04], mit dem der Dienstnehmer über das KA-Protokoll die offenen Variablen in der Beschreibung aushandeln und die Dienstausführung anstoßen kann. Für die Kommunikation zwischen Containermanager und realem Dienst musste die Dienstbeschreibung um einen Abschnitt *Grounding* erweitert werden, welcher eine Zuordnung der Variablen der Dienstbeschreibung zu den Parametern und Rückgabewerten der realen Dienstfunktionen festlegt. Generell sind somit derzeit beliebig implementierte Basisdienste innerhalb unseres Systems als DIANE-Dienste nutzbar.

Bei den oben beschriebenen Arbeiten wird implizit davon ausgegangen, dass eine Dienstanfrage durch Nutzung genau eines Dienstes befriedigt werden kann. In [Dom04, DK04] stellen wir erste Überlegungen an, wie die Diensterbringung aussehen muss, wenn Dienstanfragen nur durch die Ausführung mehrerer Dienste erfüllt werden können. Wir beschränken uns hierbei zunächst auf Anfragen nach Informationsdiensten.

## AP 2.5: Motivation

In diesem Arbeitspunkt wurde untersucht, wie die Netzteilnehmer motiviert werden können, ihre Ressourcen anderen Netzteilnehmern zur Verfügung zu stellen. Zunächst wurden die Rahmenbedingungen für ein solches Motivationselement analysiert [OKK03a]. Die Geräte, die am DIANE Ad hoc Netz teilnehmen, sind unter der Kontrolle ihrer jeweiligen Besitzer. Daher entfällt die Möglichkeit, Teile der entwickelten Kooperationsprotokolle auf manipulationssicherer Hardware unterzubringen. Es muss also davon ausgegangen werden, dass die Studenten die Kooperationsprotokolle manipulieren, wenn ihre Anwendung als nicht vorteilhaft erscheint. Wir leiteten daraus die Notwendigkeit eines Schemas ab, das die Befolgung der Kooperationsprotokolle durch Anreize motiviert. Ein solcher *Anreizschema* muss selbst-organisierend sein, da keine Infrastrukturkomponenten am Netz teilnehmen.

Es wurde deutlich [OKK03b], dass das Aufrufen und Erbringen von Applikationsdiensten nicht das einzige Kooperationsprotokoll darstellt. Darüber hinaus stellen verteilte Protokolle (wie zum Beispiel die zur Dienstvermittlung aus AP 2.3) auch Kooperationsprotokolle dar, die Anreizschemata bedürfen. Hierbei stellt das Dienstvermittlungs-Overlay einen Mehrparteien-Dienst im Sinne der Charakterisierung aus Abbildung 1 dar.

Um von dem jeweiligen Protokoll zu abstrahieren, gingen wir zunächst davon aus, dass Kooperation in eine Menge von Transaktionen zerlegt werden kann. Das sich ergebende Prinzipal-Agent-Modell wird in [OKK03b] beschrieben. Je nach Transaktion kann eine Protokollinstanz eine Rolle annehmen, entweder die des Agenten, der eine Aktion ausführt, oder die des Prinzipals, für den die Aktion vorteilhaft ist. Entsprechend der Vorüberlegung muss der Prinzipal dem Agenten einen Anreiz bieten, die Aktion auszuführen.

Die Untersuchung des Entwurfsraums von Anreizen ist unerlässlich für die Konzeption und Bewertung von Anreizschemata. Daher wurden in [ON03] verschiedene Muster von Anreizen identifiziert und bewertet. Prinzipiell können zwei Arten von Anreizen unterschieden werden. Bei vertrauensbasierten Anreizen vertraut der Agent darauf, dass der Prinzipal auch für ihn selbst eine Aktion ausführen wird. Solches Vertrauen rührt beim Kollektivmuster von der Identität des Prinzipals her, während beim Gemeinschaftsmuster aus dem früheren Verhalten des Prinzipals auf sein zukünftiges Verhalten geschlossen wird. Hingegen wird der Agent bei handelsbasierten Anreizen durch eine Gegenleistung des Prinzipals entlohnt. Beim Tauschhandelsmuster findet eine solche Gegenleistung zur selben Zeit wie die Aktion

des Agenten statt, so dass die Rollen des Prinzipals und Agenten gleichzeitig eingenommen werden. Dagegen wird bei wertpapierbasierten Anreizmustern die Gegenleistung nur versprochen, indem der Prinzipal ein nicht-abstreitbares Versprechen darüber (sog. Wertpapier) dem Agenten übergibt.

Die Namensgebung der Anreizmuster für Ad hoc Netze ist an den entsprechenden Anreizmustern der Wirtschaftswissenschaften angelehnt. Eine detaillierte Analyse der Beziehung zwischen diesen beiden Anwendungsbereichen von Anreizmustern findet sich in [AO03, And03]. Die Analyse zeigt auch, dass von Infrastruktureinheiten herausgegebene virtuelle Währungen nur dann in Ad-hoc-Netzen einsetzbar sind, wenn Manipulationssicherheit angenommen wird. Da dies für DIANE nicht möglich ist, wurde der Ansatz virtueller Währungen nicht weiter verfolgt.

Die Ressourcen und Bedürfnisse sind zwischen den Geräten des Ad-hoc-Netzes im Allgemeinen ungleich verteilt. In [OK03] untersuchten wir die Auswirkungen auf Anreizschemata. Wir zeigten, dass in einem solchen Fall die Anreizwirkung stark eingeschränkt ist. Die einzige Lösung für dieses Problem ist, den Gültigkeitsbereich von Anreizen auf das gesamte Gerät und seinen Benutzer auszudehnen.

Eine Klassifikation der existierenden Anreizschemata in Ad-hoc-Netzen führten wir in [OKK03b] durch. Es wurde deutlich, dass die existierenden Ansätze sich auf Netzwerkprotokolle beschränken. Dies bestärkte uns im Vorhaben, uns auf die Kooperationsprotokolle der Applikationsdienste und der Dienstvermittlung zu beschränken. Weiterhin zeigte die Untersuchung, dass nur zwei Anreizmuster angewandt werden. Die Ansätze, die die wertpapierbasierte Anreizmuster einsetzen, gehen entweder von Manipulationssicherheit der Geräte oder von der Verfügbarkeit von Infrastrukturkomponenten aus. Die übrigen Ansätze basieren dem Gemeinschaftsmuster und kommen ohne solche Annahmen aus. Das bedeutet, dass zur Zeit keine Anreizschemata existieren, die handelsbasierte Anreizmuster in Ad-hoc-Netzen einsetzen. Schließlich fanden wir bei jedem Ansatz die (manchmal implizite) Annahme, dass jedes Gerät eine Identität besitzt und sich authentifizieren kann. Diese Annahme lässt sich auch für DIANE nicht vermeiden. In [OKK03a] werden zwei Möglichkeiten vorgestellt, wie diese Annahme im DIANE-Umfeld durchgesetzt werden kann.

Verteilte Reputationssysteme werden nicht nur für das Gemeinschaftsmuster sondern auch für handelsbasierte Anreizmuster benötigt. Damit benötigt jedes Anreizschema ein verteiltes Reputationssystem. Vor der Konzeption spezieller Anreizschemata wurde daher zuerst ein verteiltes Reputationssystem (das Buddy-System [Fäh03]) entwickelt. Es ist von beliebigen Anreizschemata einsetzbar. Bei der Entwicklung des Buddy-Systems wurden die Ergebnisse berücksichtigt, die wir aus der Analyse der Einschränkungen verteilter Reputationssysteme [Obr04] erhielten. Das Buddy-System überwindet zum Teil diese Einschränkungen, indem es eine soziale Struktur zwischen den teilnehmenden Protokollinstanzen aufbaut [FO04].

Für das Dienstvermittlungs-Overlay Lanes aus AP2.3 wurde ein Anreizschema entworfen [Pap03]. Zu diesem Zweck wurden die Schritte des Kooperationsprotokolls bezüglich ihrer Vorteilhaftigkeit für den jeweiligen Ausführenden untersucht. Es zeigte sich, dass vor der Entwicklung des Anreizschemas als Vorarbeit das Lanes-Protokoll angepasst werden muss. Nur so kann das Verhalten der Protokollinstanzen für andere Teilnehmer wahrnehmbar gemacht werden. Wegen der asymmetrischen Rollen im Lanes-Protokoll wurden als Anreizmuster das Gemeinschaftsmuster und das Eigenwechselformat<sup>2</sup> gewählt. Das Gemeinschaftsmuster wird für häufige Transaktionen mit relativ geringem Transaktionswert eingesetzt. Aufwändigere Aktionen wie der Abgleich von Dienstsuchen und -angeboten wird durch den Erhalt eines Eigenwechsels belohnt. Als verteiltes Reputationssystem wird das Buddy-System eingesetzt. Das angepasste Lanes-Protokoll und das entworfene Anreizschema bilden zusammen das S-Lanes-Protokoll. Die Erfahrungen, die bei der Konzeption des S-Lanes-Protokolls gesammelt wurden, sind in [OKP04] zusammengefasst.

Bis zum Ende der ersten Phase des DIANE Projekts werden noch zwei Aufgaben im Bereich dieses Arbeitspunktes bearbeitet. Zum Einen werden Austauschprotokolle [Aso98] für Prinzipal-Agent-Transaktionen betrachtet, in denen Basisdienste genutzt werden. Die Verwendung von solchen Transaktionsprotokollen zielt darauf, das Aushändigen der Entlohnung an der Ausführung der Aktion zu koppeln. Zum Anderen wird ein Anreizschema für Basisdienste der Applikationsschicht entwickelt werden. Aufgrund der Vorarbeiten im Bereich der Anreizmuster, verteilten Reputationssysteme, S-Lanes und des Anreizschema-Rahmenwerks (siehe AP 3.4) erwarten wir, dass die Entwicklung eines solchen Anreizschemas unkompliziert ist.

---

<sup>2</sup>Ein Eigenwechsel ist ein Wertpapier, bei dem der Aussteller selber die versprochene Gegenleistung vollbringen wird.

## AP 2.6: Erstellung des Pflichtenhefts

Für die Konkretisierung der Aufgaben in Form eines Pflichtenheft stellte sich zunächst die Frage nach einer geeigneten Umgebung, um die entwickelten Ansätze implementieren und evaluieren zu können. Hierzu wurden verschiedene Plattformen untersucht:

- *Peer-to-Peer-Plattformen.* Hier wurde insbesondere die Java-basierte Plattform JXTA [JXTA] untersucht und erste Implementierungen zum Testen des Funktionsumfangs durchgeführt [Guo02].
- *Agentenplattformen.* Von den am Institut ansässigen Projekt KRASH [KRASH], welches Forschung im Bereich der Multiagentensysteme durchführt, wurden Informationen zu den weit verbreiteten Plattformen FIPA-OS [FIPA] und Jade [JADE] eingeholt.
- *Netzwerksimulatoren.* Die populären Open-Source-Simulatoren ns-2, GloMoSim und OPNet++ wurden auf verschiedenen Simulator-Workshops (siehe AP 5.1) des Schwerpunktprogramms vorgestellt. Mit dem kommerziellen Simulator QualNet [QUAL] wurden testweise selbst einige Protokolle implementiert [Koo02, Pon02].

Es stellte sich heraus, dass keine der betrachteten Plattformen sich für unsere Anforderungen eignet. Der Vorteil der Verwendung einer Agentenplattform liegt darin, dass sie komplexe Funktionalität zur Verfügung stellt, die von den Anwendungen genutzt werden kann. Der Nachteil der Verwendung einer Agentenplattform ist, dass sie mit einem nicht unerheblichen Overhead verbunden ist. Da Agentenumgebungen jedoch auf einem anderen Paradigma als unser Ansatz beruhen, ist die Verwendung der bereitgestellten Funktionalität nur sehr eingeschränkt möglich. Damit gehen die Vorteile der Plattformnutzung weitgehend verloren, der Overhead überwiegt. Ähnlich ist die Situation bei Peer-to-Peer-Umgebungen.

Andererseits sind klassische Netzwerksimulatoren prinzipiell nicht für anwendungsnahe Protokolle wie die Dienstvermittlung ausgelegt und bieten daher auch keine unterstützende Laufzeitumgebung. Weiterhin bieten sie nur wenig Möglichkeiten, realistische Benutzer zu simulieren, sondern beschränken sich auf einfache analytische Bewegungsmodelle und Traffic-Dateien. Die Modellierung der unteren vier Ebenen ist für unsere Untersuchungen weitaus zu präzise, so dass die Flut der einzustellenden Parameter oft nicht zu beherrschen ist und die Ergebnisse unverhältnismäßig stark beeinflussen. Problematisch ist auch, dass die entwickelten Protokolle nur im Simulator laufen und nicht auf reale Geräte übertragen werden können.

Wir entschieden uns daher, eine eigene Simulationsplattform zu entwerfen, die genauer an unsere Bedürfnisse angepasst ist: Abstrakte Modellierung des Netzes, gute Unterstützung bei der Entwicklung von anwendungsnahen Protokollen, Tests mit realistischen Benutzern und eine direkte Übertragbarkeit der implementierten Protokolle auf reale Geräte.

Die Plattform sollte die folgenden Anforderungen erfüllen: Die unteren vier Schichten des ISO/OSI-Modells sollten stark vereinfacht werden und über eine Schnittstelle, die die beim Workshop zum netzwerknahe Routing vorgestellten Anforderungen (siehe AP 5.1) erfüllt, mit dem Transportsystem verbunden werden. Umgekehrt soll eine detaillierte Modellierung der oberen Schichten einfach möglich sein. Insbesondere werden die Möglichkeiten, realitätsnahe Benutzermodelle einzubinden und mit geringem Aufwand Dienstvermittlungsprotokolle zu entwickeln benötigt.

Aus diesen Anforderungen ergab sich die folgende Schichtenarchitektur für den zu entwickelnden Simulator (Abbildung 2a):

- Die *Verbindungsebene* berechnet, ob zwei Geräte eine direkte Kommunikationsmöglichkeit besitzen. Dies wird aus der maximalen Funkreichweite und den Umweltgegebenheiten bestimmt.
- Die *Netzwerkebene* bietet verschiedene Methoden an, um Nachrichten durch das Netz zu senden: Unicast, Multicast, Anycast und Broadcast. Die Netzwerkschicht sendet außerdem Crosslayer-Ereignisse, z.B. wenn Nachrichten nicht zustellbar sind bzw. fremde Nachrichten mitgehört werden.

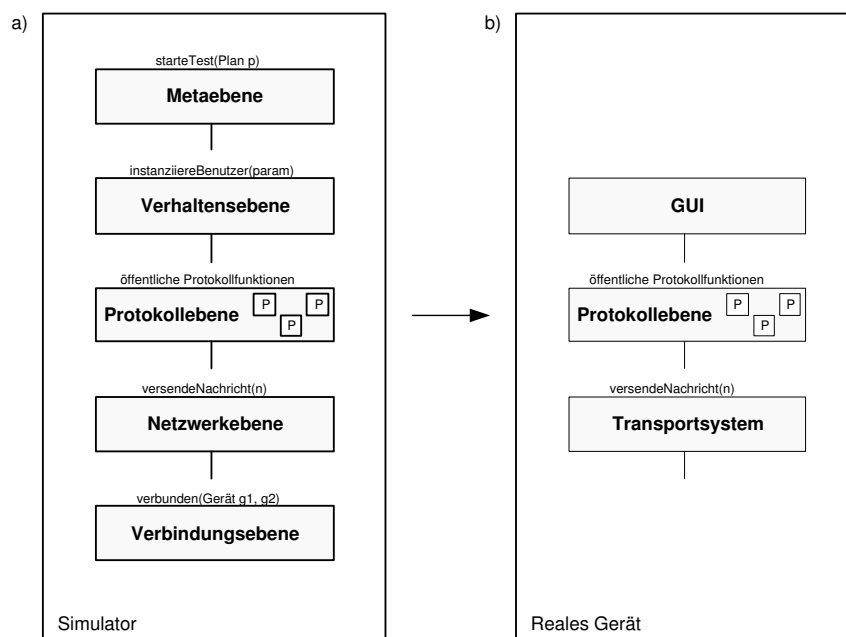


Abbildung 2: Schichten des Emulators DIANEmu: a) als Simulator, b) auf dem realen Gerät.

- Die *Protokollebene* ist die wichtigste Schicht des Simulators. Sie bietet ein Rahmenwerk, um eigene Protokolle ablaufen zu lassen, die festgelegten Spezifikationen entsprechen müssen. Protokolle sind dabei als endliche Automaten realisiert, die Benutzern, aber auch anderen Protokollen öffentliche Funktionen anbieten können (wie z.B. login, sucheDienst, kuendigeDienstAn etc.). Zu deren Realisierung versenden die Protokolle Nachrichten über die Netzwerkschicht. Die Protokollebene bietet eine Vielzahl von Hilfen, die die Implementierung von Protokollen erleichtern sollen. Beispiele hierfür sind Methoden zum automatischen Sammeln von Nachrichten, zum automatischen Verteilen von eintreffenden Nachrichten an die korrekten Protokollzustände, transparente Parallelität und automatisches Multiplexing durch Verwaltung von Sitzungen.
- Die *Benutzerebene* modelliert einzelne Benutzer durch Abarbeitung von Ablaufskripten. Diese enthalten Anweisungen zum Lebenszyklus (erzeuge, zerstöre), zur Geräteverwendung (schalteEin, schalteAus), zur Bewegung (geheNach, warte) und zum Verhalten (d.h. die Nutzung öffentlicher Protokollfunktionen).
- Die *Metaebene* schließlich ist zuständig, um komplette Testszenarien durchzuspielen. Sie generiert eine Menge von Benutzern in Form von Skripten.

Alle zu entwickelnden Protokolle sollen ohne großen Aufwand auf realen Geräten lauffähig sein. Hierzu kann die Protokollebene mit allen benötigten Protokollen aus dem Simulator herausgelöst und auf das Gerät übertragen werden (siehe Abbildung 2b). Die öffentlichen Funktionen werden dann durch eine GUI (siehe AP 3.5) verfügbar gemacht, das Versenden von Nachrichten erfolgt über ein externes Transportsystem.

### AP 3: Implementierung eines Prototyps

In den folgenden Arbeitspunkten befassen wir uns mit der Implementierung und Evaluation der Konzepte aus Arbeitspunkt 2. Wie in Arbeitspunkt 2.6 festgehalten, sollte die Implementierung der einzelnen Module als Teil eines Simulators erfolgen. Einen Überblick über diesen Simulator, *DIANEmu* genannt, bietet [Kle03]. Eine aktuelle Version findet sich auf der beiliegenden CD.

### AP 3.1: Lernmaterialbereitstellung

Wie im Zwischenbericht zu AP 1.1 beschrieben, wurde die Entwicklung eines Prototypen, der auf der Nutzung realer Ad-hoc-Netze beruht, zunächst einmal verschoben. Die erzielten Ergebnisse wurden vielmehr mittels Simulation evaluiert. Für eine solche simulative Evaluation spielt die Existenz real nutzbarer Lernmaterialien nur eine untergeordnete Rolle. Dieser Arbeitspunkt wurde daher bislang nicht intensiv bearbeitet. In laufenden Arbeiten sollen jedoch Beispieldienste (und hier unter anderem Informationsdienste, die Lernmaterialien bereitstellen) entwickelt werden. Diese Beispieldienste können auf die am Lehrstuhl im Rahmen des SCORE-Projektes [SCORE] entstandene umfangreiche Sammlung von modularisierten und mit Metadaten annotierten Lernmaterialien zugreifen.

### AP 3.2: Implementierung des Moduls “Dienstsuche”

Eine Implementierung der Dienstvermittlungsprotokolle aus AP 2.3 sollte zeigen, welche der entwickelten Verfahren unter welchen Voraussetzungen eine Effizienzsteigerung bewirken und ob die die Effektivität erhalten bleibt. Die Effizienz lässt sich für ein gewisses TestszENARIO an verschiedenen Parametern festmachen, etwa der Zahl der versendeten Nachrichten, die gesamte versendete Informationsmenge, der gesamte Energieverbrauch oder die durchschnittliche Wartezeiten der Benutzer. Die Effektivität ergibt sich aus der Anzahl der vermittelten Dienstsuchen durch die Anzahl der insgesamt vermittelbaren. Verglichen werden müssen die Protokolle gegen triviale Ansätze wie Fluten und zentralistische Ansätze sowie ausgereifere (aber meist internetbasierte) Methoden.

Zur Zeit finden umfangreiche Messungen der verschiedenen Strukturen mit DIANEmu statt. Bis zum Ende der Laufzeit sollen diese abgeschlossen und dokumentiert werden. Dazu werden die Messergebnisse laufend gesammelt und zu einem technischen Abschlussbericht zusammengefasst [KHMM04], der am Ende der Projektlaufzeit vollständig zur Verfügung stehen wird.

Bereits jetzt lassen sich einige grundlegende Aussagen über die Verwendung von Overlays zur Dienstvermittlung ablesen:

- Semantische Overlays und traditionelle internetbasierte Verfahren sind aufgrund ihrer komplexen Strukturbedingungen bereits für mäßig dynamische Ad-hoc-Netze unter realistischen Bedingungen nicht mehr zufriedenstellend wartbar. Ein Großteil der versandten Nachrichten wird für die Durchführung von Reparatur- und Optimierungsarbeiten verwendet. Mehrfachfehler, d.h. gleichzeitig auftretende Probleme in der Overlaystruktur führen oft dazu, dass die Effektivität sinkt, da die Struktur nicht mehr fehlerfrei genutzt werden kann und ein aufwändiger Neuaufbau nötig wird.
- Halbsemantische Overlays insbesondere Lanes sind aufgrund ihrer speziellen Anpassbarkeit an die Eigenschaften der Ad-hoc-Netze in realistischen Bewegungs- und Aktionsszenarien wie dem von uns betrachteten Campusszenario herkömmlichen Flutungsverfahren im Bezug auf Nachrichteneffizienz und zentralistischen Verfahren im Bezug auf Effektivität überlegen (siehe dazu auch [Pap03]). In unnatürlicheren Umgebungen, wie sie durch entsprechend parametrisierte, analytische Bewegungsmodelle erzeugt werden können, geraten auch solche Overlays an ihre Grenzen – der Overhead zur Aufrechterhaltung der Struktur überwiegt dann gegenüber dem Vorteil auf ein Verzicht auf Anfrageflutung.
- Mit Zusatzmaßnahmen kann die Effizienz der Overlays wesentlich verbessert werden. Hierzu zählt insbesondere die Verwendung von Cachingmechanismen.

Zusammenfassend lässt sich also sagen, dass sich halbsemantische Overlays, insbesondere Lanes, eignen, um in Umgebungen mit realistischen Eigenschaften effektiv und effizienter als einfache oder traditionelle Ansätze Dienste zu vermitteln.

### AP 3.3: Implementierung des Moduls “Dienstausführung”

Bei der Implementierung der Dienstausführung waren die Beschreibungserweiterungen, Komponenten und Protokolle aus AP 2.4 umzusetzen [Fis04]:

- Die Dienstbeschreibung wurde um den Abschnitt *Grounding* erweitert, der angibt, wie der Dienst konkret angesprochen werden kann. Verschiedene Groundingvorlagen ermöglichen die Einbindung beliebig implementierter Dienste. Zur Zeit werden Dienste als Javamethoden unterstützt, eine Unterstützung für mit SOAP ansprechbare Webservices ist in Arbeit.
- Die Implementierung des *Containermanagers* erfolgte in Java. Der Manager bietet Dienstgebern die Möglichkeit, eigene Dienste über eine Administrationsschnittstelle einzubringen, zu starten und zu stoppen. Die nötige Kommunikation mit der Außenwelt leitet der Manager aus dem Profilabschnitt der Dienstbeschreibung ab, die Kommunikation mit dem realen Dienst aus dem Groundingteil. Zudem setzt der Containermanager transparent Techniken ein, die es Diensten ermöglicht, gleichzeitig von mehreren Dienstgebern verwendet zu werden, wenn diese es wünschen.
- Das *Konfigurations- und Ausführungsprotokoll* wurde als DIANEmu-Protokoll implementiert und ist somit auch auf realen Geräten einsetzbar.

In der nächsten Antragsphase soll der Containermanager so erweitert werden, dass er auch zusammengesetzte Dienste anbieten kann.

### AP 3.4: Implementierung des Moduls “Dienstmotivation”

Das Anreizschema-Rahmenwerk ist in Java implementiert und mit DIANEmu gekoppelt. Es kann aber auch ohne DIANEmu betrieben werden und ist damit generell einsetzbar. Das Rahmenwerk sieht den Einsatz von Bausteinen vor, die auf jeweils eine der folgenden Aufgaben erledigen: Verwaltung der Nutzen-/Kostenbewertung (*ValuationManager*), Erbringung der kryptographischen Dienste (*CryptoManager*), Verwaltung von nicht-abstreitbaren Marken (*Evidence-Manager*), Teilnahme am verteilten Reputationssystem (*ReputationSystem*) und Benutzung von Transaktionsprotokollen (*TransactionProtocols*). Das Rahmenwerk zielt darauf ab, für die Implementierung eines Anreizschemas die jeweils passendsten Bausteine einzusetzen und somit dem Entwickler des Anreizschemas den Großteil seiner Arbeit abzunehmen. Außerdem ist es somit für Dritte möglich, ihre eigenentwickelten Bausteine im Rahmenwerk einzusetzen. So wird im Rahmen des DIANE Projekts davon abgesehen, den Baustein für die Erbringung kryptographischer Dienste zu implementieren.

Die Implementierung des Buddy-Systems ist als Baustein für das Anreizschema-Rahmenwerk ausgelegt und findet sich in [Fäh03] beschrieben. Die Messergebnisse zeigen, dass das verteilte Reputationssystem durch seine soziale Struktur insbesondere auch Komplote mehrerer Teilnehmer aufdecken kann [FO04].

Das S-Lanes-Protokoll wurde basierend auf das Anreizschema-Rahmenwerk und dem Buddy-System implementiert [Pap03]. Eine Beschreibung der Messergebnisse für unterschiedlichen Testszenarien und die Interpretation der Messergebnisse findet sich in [OKP04]. Es wird deutlich, dass S-Lanes eine hohe positive Korrelation des Kooperationsaufwands und -nutzens der jeweiligen Teilnehmer erzwingt, wo hingegen bei Lanes keine solche Korrelation existiert. Der Gesamtaufwand der Kooperation in S-Lanes ist trotz des Zusatzaufwands des Anreizschemas oft geringer als bei Lanes, da durch die Benutzung des Buddy-Systems unkooperative Teilnehmer identifiziert werden und ausgeschlossen werden.

Bis zum Ende der ersten Phase des DIANE Projekts werden noch Transaktionsprotokolle für Basisdienste implementiert und als Transaktionsprotokoll-Baustein in das Anreizschema-Rahmenwerk eingebunden. Damit kann bei der Implementierung des Anreizschemas für Basisdienste der Applikationsschicht auch auf einen passenden Transaktionsprotokoll-Baustein zurückgegriffen werden. Die Entwicklung der Bausteine für die Verwaltung der Nutzen-/Kostenbewertung und der nicht-abstreitbaren Marken wird in der zweiten Projektphase erfolgen.

### AP 3.5: Implementierung von Benutzerschnittstellen

Zwar wurde in AP 1.1 festgestellt, dass zum jetzigen Zeitpunkt Studierende Dienste wohl am ehesten mit einem Laptop-Computer anbieten und nutzen würden, wir gehen jedoch davon aus, dass mit der zunehmenden Verbreitung von Smartphones diese Geräte als zweite große Endgerätegruppe eine wesentliche Rolle spielen werden. Es wurden daher erste Überlegungen angestellt, wie man Dienstangebote und -suchen auf den doch sehr kleinen Displays dieser Geräte realisieren kann. Die Verwendung von noch weniger mächtigen Endgeräten, etwa “normalen” Handys halten wir für unwahrscheinlich.

Es wurden unterschiedliche Sprachen zur Anpassung von Benutzeroberflächen an verschiedene Geräte evaluiert. Im Rahmen einer Studienarbeit [Bre02] entstand eine prototypische Realisierung einer solchen dynamisch anpassbaren Oberfläche mittels UIML [UIML]. Die Ergebnisse dieser Untersuchung waren eher ernüchternd: Eine rein syntaktische Anpassung der Darstellung reicht offenbar nicht aus, um die Dienstsuche und -nutzung mit kleinen Geräten zu ermöglichen. Es wurde daher ein Konzept entworfen, das eine auch inhaltliche Anpassung ermöglicht. Die Grundidee hierbei ist es die Oberfläche als Dienstnehmer aufzufassen, der unterschiedliche Dienste nutzt um zu einer Darstellung zu gelangen. Eine detaillierte Ausarbeitung und Umsetzung dieses Konzeptes ist für die zweite Phase geplant.

Insgesamt spielte die Entwicklung von Benutzeroberflächen wegen der Entscheidung das System zunächst über einen Simulator zu evaluieren eine nur untergeordnete Rolle. Die Entscheidung zur Simulation brachte jedoch eine neue Anforderung mit sich: Zwar ist es nicht notwendig, dass ein einzelner realer Benutzer das System komfortabel verwenden kann, dafür ist es aber notwendig, große Zahlen solcher Benutzer realitätsnah zu simulieren. Diese Modellierung konnte in zwei Teile zerlegt werden: die Beschreibung der Bewegung der Benutzer sowie die Beschreibung ihres Verhaltens. Um solche Beschreibungen zu erhalten, entwickelten wir ein integriertes Modell (siehe [BKOK04]), welches jedem Benutzer unseres Campus-Szenarios einen Plan realweltlicher Aktivitäten zuordnet (wie z.B. Vorlesung besuchen, in die Mensa gehen, lernen, Buch ausleihen etc.) und daraus auf natürliche Weise sowohl Bewegung als auch die Verwendung des mobilen Gerätes (Ein-, Ausschalten, Nutzung von Diensten) ableitet. Die Parameter des Modells wurden aus Umfragen unter Studenten abgeleitet [Bre03].

### AP 3.6: Integration der Module

Die einzelnen Module wurden unmittelbar als Bestandteile von DIANE<sub>mu</sub> entwickelt, so dass eine nachträgliche Integration nicht notwendig war.

### AP 3.7: Dokumentation des Systems

Die Dokumentation des entstandenen Systems ist viergeteilt:

- Zunächst wurde ein *technischer Bericht* zur Simulationsumgebung DIANE<sub>mu</sub> erstellt, der den Funktionsumfang insbesondere für interessierte Dritte zusammenfassend erklärt [Kle03].
- Für Nutzer und Protokollentwickler von DIANE<sub>mu</sub> wurde zudem ein ausführliches *Benutzerhandbuch* erstellt. Dieses teilt sich in zwei Abschnitte: Ein praktischer Teil, der mittels Installationsanweisungen und Tutorials einen schnellen und praktischen Einstieg in die Simulationsumgebung bietet, sowie einen Referenzteil, der ausführlich auf den Funktionsumfang in den einzelnen Schichten eingeht. Das Handbuch ist noch in Arbeit und wird ständig erweitert [KHMM].
- Für Entwickler von Anreizschemata wird zur Zeit ein Übersicht des Anreizschema-Rahmenwerks zusammengestellt.
- Weiterhin existiert eine rein technische Dokumentation in Form von *javadoc-Seiten*. Diese beschreiben konkret die einzelnen Klassen und Methoden des Systems und sind insbesondere für Weiterentwickler des DIANE<sub>mu</sub>-Simulators und des Anreizschema-Rahmenwerks interessant.

## AP 4: Evaluierung

### AP 4.1: Einsatz und Test des Gesamtsystems

Der Simulator DIANEmu (siehe AP 2.6 sowie AP 3) ermöglicht die Übertragung der implementierten Protokolle auf reale Geräte. Vor dem Einsatz des Gesamtsystems in der Praxis sind dennoch zwei zusätzliche Schritte vonnöten. Sie betreffen das zugrundeliegende Transportsystem und die Installation der DIANE Software auf die mobilen Geräte.

DIANEmu nimmt an, dass das Transportsystem Nachrichten zuverlässig zustellt, sofern dies aufgrund der Topologie möglich ist. Weiterhin wird davon ausgegangen, dass das Transportsystem Anycast zur Verfügung stellt und passives Mithören von Nachrichten, die für andere Geräte bestimmt sind, ermöglicht. Unsere Untersuchungen in [Deh04] ergaben, dass kein solches Transportsystem für Ad-hoc-Netze existiert und auch nicht von anderen Teilprojekten des SPP bereitgestellt wird. Daher wurde ein eigenes Transportsystem entwickelt, das basierend auf einer verfügbaren Implementierung des Netzwerk-Protokolls AODV [RT99] diese Funktionalität bereitstellt. Zum Zeitpunkt dieses Zwischenberichts findet sich dieses Transportsystem in der Testphase.

Für kleinere mobile Geräte ist es aufwändig, neue Software herunterzuladen und zu installieren. Um den Benutzern der DIANE Software diese Schritte zu erleichtern, wurde die DIANE-Software-Tankstelle entwickelt. Sie ist in [Liu03] beschrieben. Die Software-Tankstelle ermöglicht es, Softwareversionen vom DIANE-Webserver über das Internet auf die mobilen Geräte herunterzuladen und automatisch zu installieren. Dabei werden die Komponenten-Versionen auf der Server-Seite mit Hilfe von CVS kontrolliert. Für die Wahl der zu installierenden Komponenten und deren jeweiligen Versionen wird ein ausführbares Installationsprogramm auf der Klientenseite zur Verfügung gestellt. Eine genaue Beschreibung der Versionskontrolle findet sich in [LO03].

Mit der Bereitstellung des Transportsystems und der Software-Tankstelle wurden die Rahmenbedingungen für den praktischen Einsatz des Gesamtsystems hergestellt. Dennoch sehen wir von einer praktischen Evaluation des Gesamtsystems ab. Dies begründet sich damit, dass der Aufwand eines solchen Vorgehens in keinem Verhältnis zu dem antizipierten Nutzen steht. Zudem sind die Geräte der Studenten zur Zeit für den Betrieb des Java-basierten Gesamtsystems zu langsam. Daher haben wir uns auf die simulative Evaluation des Gesamtsystems beschränkt, wie sie im AP3 vorgestellt wurde.

## AP 5: Querschnittsaufgaben

### AP 5.1: Austausch mit anderen Teilprojekten des SPP/Berichterstattung an die DFG

Die Ergebnisse des Projekts wurden über die Projekt-Webseite und die Vorträge bei SPP-Kolloquien und SPP-Workshops an andere Teilprojekte weitergegeben. Besucht und vorgetragen wurde bei den Workshops zum netzwerknahe Routing<sup>3</sup>, zum anwendungsnahe Routing<sup>4</sup>, zu Bewegungsmodellen<sup>5</sup> und zur Simulation<sup>6</sup>. Der Workshop zum anwendungsnahe Routing wurde von der Arbeitsgruppe des DIANE-Projekts veranstaltet.

Bei den diversen Workshops und Kolloquien zeigte sich im Laufe der Zeit immer deutlicher, dass zumindest einige der anderen Gruppen im SPP ähnliche Vorstellungen über einen sinnvollen Aufbau und eine sinnvolle Funktionalität für ein System zur gemeinsamen Ressourcennutzung in Ad-hoc-Netzen haben, wie sie dem DIANE-Projekt zugrunde

---

<sup>3</sup>Workshop: Routing, 15.01.2003, Universität Karlsruhe:

<http://www.telematik.informatik.uni-karlsruhe.de/forschung/SPP1140/workshop/routing/veranstaltungen.html>

<sup>4</sup>Workshop: Anwendungsnahe Routing, 12.-13.06.2003, Universität Karlsruhe:

[http://www.telematik.informatik.uni-karlsruhe.de/forschung/SPP1140/workshop/app\\_routing/veranstaltungen.html](http://www.telematik.informatik.uni-karlsruhe.de/forschung/SPP1140/workshop/app_routing/veranstaltungen.html)

<sup>5</sup>Workshop: Bewegungsmodelle in der Simulation, 25.04.2003, International University Bruchsal:

<http://www.telematik.informatik.uni-karlsruhe.de/forschung/SPP1140/meetings/mobility/veranstaltungen.html>

<sup>6</sup>Workshop: Simulation, 10.02.2003, Universität Bonn:

<http://www.telematik.informatik.uni-karlsruhe.de/forschung/SPP1140/workshop/simulation/veranstaltungen.html>

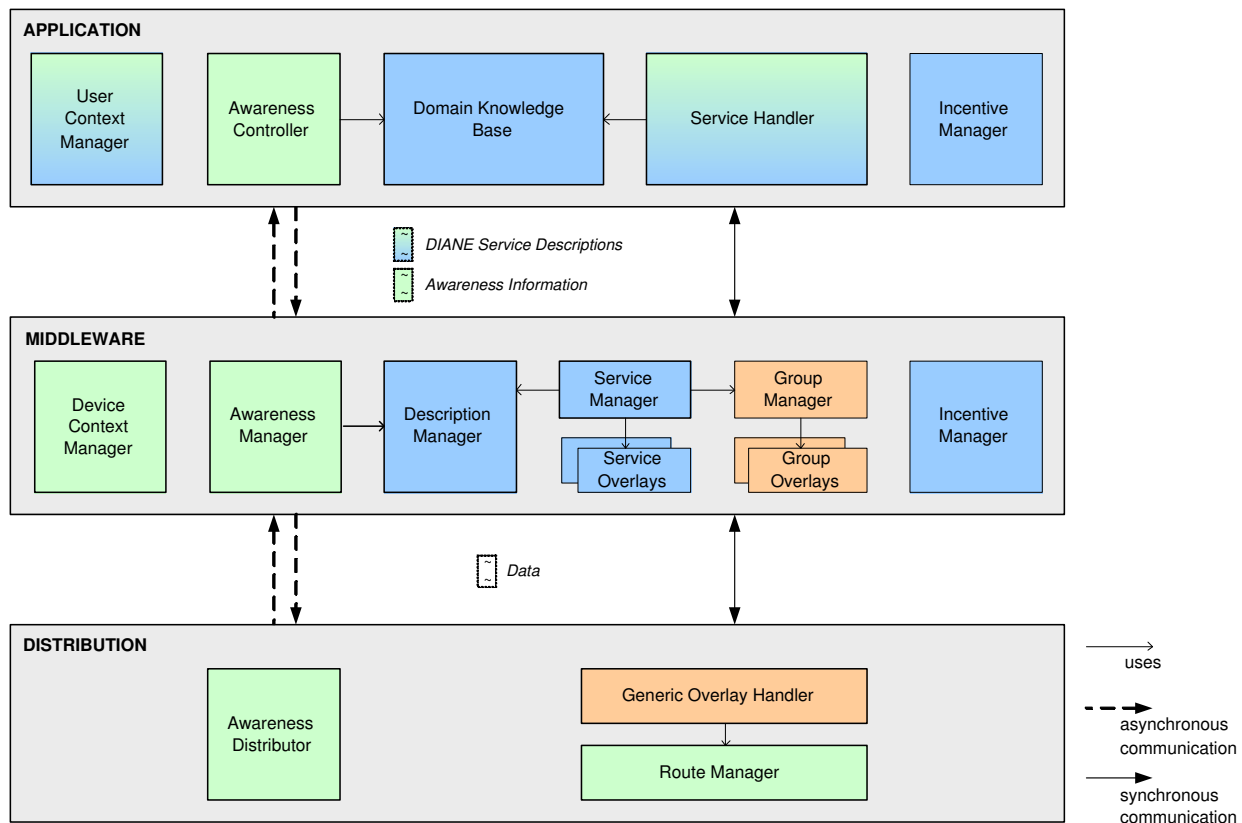


Abbildung 3: Kooperation mit ELAN und MAMAS

liegen. Es wurde daher in Zusammenarbeit mit zunächst zweien dieser Projekte, MAMAS und ELAN, eine generische Architektur zur Dienstsutzung in Ad-hoc-Netzen entwickelt (siehe Abbildung 3). Diese Architektur zeigt zum einen, welche Komponenten in einem solchen System wünschenswert sind, und zum anderen, wie sich die drei Projekte in die Thematik einordnen. Sie ermöglicht es so, Kooperationsmöglichkeiten zu identifizieren.

Die Abbildung zeigt eine Drei-Schichten-Architektur. Zwischen den Schichten findet jeweils sowohl synchrone als auch asynchrone Kommunikation statt.

Aus Benutzersicht sollte ein solches System zur verteilten Ressourcennutzung zwei Arten von Funktionalität bereitstellen: Benutzer müssen in der Lage sein, Ressourcen anzubieten und sie müssen in der Lage sein, Ressourcen anderer Teilnehmer zu nutzen. Interesse an Ressourcennutzung kann einmalig sein (“Ich suche Übungsblatt 6 zur Vorlesung Datenbankeinsatz”) oder auch langfristig (“Ich interessiere mich für alles, was es zum Thema Datenbankeinsatz gibt”). Im ersteren Fall sollten Benutzer in der Lage sein, eine konkrete Anfrage zu stellen, die dann mittels einer Dienstsuche aktiv im Netz verbreitet wird, im zweiten Fall sollten sie in der Lage sein, ihr generelles Interesse zu bekunden, diese Interessensäußerung sollte dann, wenn die Benutzer andere Geräte treffen, mit den dort gebotenen Ressourcen abgeglichen werden. Das Herzstück der *Application*-Schicht ist eine *Domain Knowledge Base*. Diese enthält Ontologien, die die betrachtete Domäne beschreiben und u.U. auch domänenspezifische Vergleichsfunktionen enthalten. Diese Wissensbasis ist somit Grundlage für die Beschreibung angebotener und gesuchter Dienste. Sie wird sowohl vom *Service Handler* als auch vom *Awareness Controller* verwendet. Aufgabe des *Service Handlers* ist es, Benutzern das Verwenden und Anbieten von Diensten zu ermöglichen. In dieser Komponente sind unter anderem Möglichkeiten zur Beschreibung und zum Anbieten von Diensten, Möglichkeiten zur Formulierung von Dienstfragen und Oberflächen und Schnittstellen zur Dienstsutzung gekapselt. Der *Service Handler* enthält somit einerseits den oben erwähnten Containermanager und unterstützt andererseits die aktive Suche nach Diensten. Längerfristiges Interesse

an Ressourcen wird durch den *Awareness Controller* unterstützt. Beide Komponenten können Zusatzinformation über den Benutzerkontext verwenden, um effizienter oder effektiver zu werden. Diese Information wird von einem *User Context Manager* bereitgestellt. Der *Incentive Manager* schließlich sorgt durch passende Anreizschema für eine faire Ressourcennutzung.

Die *Application*-Schicht gibt an die unter ihr liegende *Middleware*-Schicht DIANE Dienstbeschreibungen weiter und erhält von dieser ebenfalls Dienstbeschreibungen und Awarenessinformation. Aufgabe der *Middleware* ist es, Mechanismen bereitzustellen, mit denen Angebote und Nachfragen gegeneinander abgeglichen werden können. Hierzu muss zum einen eine geeignete Verbreitung der Information im Netz sichergestellt werden, zum anderen der eigentliche Vergleich durchgeführt werden. Im Falle der passiven Verbreitung von Awarenessinformation ist dies Aufgabe des *Awareness Handlers*. Für Dienstsuchen übernimmt diese Aufgabe der *Service Handler*. Beide Komponenten verwenden zur Durchführung des Vergleichs den *Description Manager*. Der *Service Manager* sorgt zudem für die Verbreitung von Angeboten und Nachfragen im Netz. Dies kann er entweder selber unter Zuhilfenahme der *Group Manager* erledigen, oder er kann sich zur Erfüllung dieser Aufgabe auf *Service Overlays* stützen. Auch auf dieser Ebene ermöglicht Kontextinformation eine verbesserte Durchführung und sorgt ein *Incentive Manager* für Fairness bei der Ressourcennutzung.

Die unterste Ebene, *Distribution*, setzt die von den Komponenten der *Middleware* vorgegebene logische Verteilung von Information in tatsächliche physikalische Verteilung um. Hier haben Routingprotokolle, aber auch generische Verfahren zur Optimierung und Verwaltung von *Overlays* ihren Platz. Auch hier kann Awarenessinformation eine Anpassung an das Netzwerk ermöglichen und damit zur Steigerung der Effizienz beitragen.

Die farbliche Codierung des Bildes beschreibt, welche Komponenten von welchem der drei an der Entwicklung beteiligten Projekte bereitgestellt werden können. Die blau gekennzeichneten Komponenten sind dabei diejenigen, die aus dem DIANE Projekt stammen, ELAN ist grün und MAMAS rot eingefärbt. Einige der von uns entwickelten Komponenten können vom ELAN-Projekt benötigte Funktionalität anbieten, besonders zentral sind hier die DIANE-Dienstbeschreibungen, der *Description Manager*, die *Domain Knowledge Base* und der *Service Handler*. Umgekehrt könnte DIANE von der Verwendung einzelner Komponenten aus dem ELAN oder MAMAS Projekt profitieren. Besonders interessant erscheint hier die Verwendung der im MAMAS-Projekt entwickelten *Overlays* und generischen Verwaltungsfunktionen. Eine nähere Untersuchung der Durchführbarkeit einer solchen Integration ist für die nächste Projektphase geplant. Noch in dieser Phase sollen die Architektur und insbesondere die Schnittstellen detaillierter ausgearbeitet werden.

## Zusammenfassung und Ausblick

Ziel des Projekts DIANE in der ersten Projektphase war die Entwicklung von Mechanismen für die effektive und effiziente Dienstsuche und -nutzung in Ad-hoc-Netzen. Zunächst wurde mit Hilfe einer Umfrage in Erfahrung gebracht, welche Rahmenbedingungen für unser Lernszenario gelten. Es zeigte sich, dass Benutzer an einer Fülle sehr unterschiedlicher Dienste interessiert sind. Eine Analyse des Gestaltungsraums für Dienste führte zur Fokussierung auf die Basisdienste, Dienste also, die automatisch nutzbar sind.

Bei der *Beschreibung von Diensten* stellte sich die automatische Verwendung als besonders schwierig heraus. Ein Rechner kann eine Beschreibung nur dann sinnvoll verarbeiten, wenn für ihn die Semantik eindeutig ersichtlich ist. Daher wurde ein Verfahren der Ontologieschichtung vorgeschlagen, mit dessen Hilfe das Vokabular von Beschreibungen in strukturierter und flexibler Weise vereinheitlicht werden konnte. Zudem wurden Erweiterungen entwickelt, die mit Hilfe reiner Zustandsorientierung mit Variablen die Konfigurationssemantik von angebotenen Diensten sowie mit unscharfen Objektmenge die Präferenzsemantik von benötigten Diensten eindeutig erfassbar machten. Letztere erlaubten es, automatisch einen persönlichen Vergleich abzuleiten. Zur Zeit erfolgt eine Formalisierung der Sprache in mehreren Repräsentationsformen, die mittels eines Transformators ineinander überführt werden können und für unterschiedliche Aufgaben optimiert sind.

Es wurde gezeigt, dass sich für die *Vermittlung von Diensten* internetbasierte Ansätze nicht eignen. Daher wurden Dienstvermittlungs-Overlays vorgeschlagen, die die Eigenschaften von Ad-hoc-Netzen berücksichtigen. Die semantischen Overlays erwiesen sich als zu unflexibel, da die Overlaystruktur sowohl die Semantik der zu vermittelnden Dienste als auch die Topologie des Ad-hoc-Netzes widerspiegeln muss. Daher wurde bei den halbsemantischen Overlays der Ansatz verfolgt, von der Semantik der Dienste zu abstrahieren. Das Lanes-Protokoll benutzt ein solches Overlay und kommt daher mit Topologieänderungen besser zurecht. Die Evaluation zeigte, dass in realitätsnahen Umgebungen Lanes effizienter als traditionelle Ansätze ist.

Für eine automatische *Dienstausführung* muss ein nicht-menschlicher Nutzer eindeutig erkennen, in welcher Art der Dienstgeber konfiguriert und aufgerufen werden kann. Wir entwickelten daher ein generisches Konfigurations- und Ausführungsprotokoll, welches durch die eindeutige Konfigurationssemantik der Dienstbeschreibung konkretisiert werden kann. Um einem Dienstgeber die Einbindung von Diensten zu erleichtern, wurde ein Containermanager konzipiert, der die Kommunikationslücke zwischen KA-Protokoll und realem Dienstaufwurf überbrückt. Hierzu wurde die Dienstbeschreibung um den Abschnitt des Groundings erweitert, welcher die Abbildung der Variablen und Typen in der abstrakten Beschreibung auf die konkreten, programmiersprachlichen Parameter der realen Funktionsaufrufe übernimmt.

Um Geräte zu kooperativem Verhalten zu *motivieren*, wurden selbst-organisierende Anreizschemata vorgeschlagen. Der Entwurfsraum von Anreizen wurden untersucht und basierend darauf wurden existierende Anreizschemata klassifiziert. Zunächst wurde ein verteiltes Reputationssystem entwickelt, das durch seine soziale Struktur besonders effektiv bei der Identifikation von unkooperativen Teilnehmern ist. Darauf aufbauend schlugen wir ein Anreizschema für das Dienstvermittlungs-Overlay Lanes vor. Das resultierende S-Lanes-Kooperationsprotokoll koppelt den Kooperationsaufwand und -nutzen der jeweiligen Teilnehmer und erzielt dadurch eine starke Anreizwirkung. Es wurde ein Rahmenwerk entwickelt, das die Entwicklung von beliebigen Anreizschemata durch wieder verwendbare Bausteine unterstützt. In der verbleibenden Zeit soll ein Anreizschema für Basisdienste basierend auf dem Rahmenwerk konzipiert werden.

Zur Unterstützung der *Implementierung* anwendungsnaher Protokolle wurde der DIANEmu-Simulator entwickelt. Er ermöglicht die Integration der Module und deren direkte Übertragung auf reale Geräte. Zum Einspielen bestimmter Versionen der entwickelten Module auf die mobilen Geräte wurde eine internetbasierte Software-Tankstelle entwickelt. Die praktische Evaluation erwies sich als zu wenig nutzbringend, so dass die Evaluation in der Simulation erfolgte.

Für die nächste SPP-Phase ergeben sich aus dem geschilderten Vorhaben damit vor allem zwei wichtige Aspekte: Zum einen ist der Übergang von den betrachteten Basisdiensten zu komplexeren Dienstklassen denkbar. Dieser ist notwendig, da viele der szenariorelevanten Dienste komplex sind. Bei dem Übergang zu solchen Diensten ist zu erwarten, dass die Mechanismen zur Dienstbeschreibung, -vermittlung, -nutzung und -motivation erweitert werden müssen. Zum anderen hat sich die Entwicklung von Anreizschemata als sehr komplex erwiesen. Daher sollten die Erfahrungen der ersten Projektphase zu einem strukturierten Vorgehen ausgebaut werden, welches die Komplexität der Anreizschema-Entwicklung beherrschbar macht.



## Literatur

### Begutachtete Veröffentlichungen bei Konferenzen und Workshops aus der ersten Phase des geförderten Projektes

- [BKOK04] Tobias Breyer, Michael Klein, Philipp Obreiter und Birgitta König-Ries. *Activity-Based User Modeling in Service-Oriented Ad-Hoc-Networks*. In First Working Conference on Wireless On-Demand Network Systems (WONS 2004). Trento, Italien, Januar 2004.
- [KK02] Birgitta König-Ries und Michael Klein. *Information Services to Support E-Learning in Ad-Hoc Networks*. In First International Workshop on Wireless Information Systems (WIS2002), S. 13–24. Ciudad Real, Spanien, April 2002.
- [KK02b] Michael Klein und Birgitta König-Ries. *Multi-Layer Clusters in Ad-Hoc Networks - An Approach to Service Discovery*. In Proceedings of the First International Workshop on Peer-to-Peer Computing (Co-Located with Networking 2002), S. 187–201. Pisa, Italien, Mai 2002.
- [KK03] Michael Klein und Birgitta König-Ries. *A Process and a Tool for Creating Service Descriptions Based on DAML-S*. In Proc. of the 4th VLDB Workshop on Technologies for E-Services (TES'03). Berlin, September 2003.
- [KKO03b] Michael Klein, Birgitta König-Ries und Philipp Obreiter. *Lanes – A Lightweight Overlay for Service Discovery in Mobile Ad Hoc Network*. In Proc. of the 3rd Workshop on Applications and Services in Wireless Networks (ASWN2003). Bern, Schweiz, Juli 2003.
- [KKO03c] Michael Klein, Birgitta König-Ries und Philipp Obreiter. *Service Rings – A Semantical Overlay for Service Discovery in Ad Hoc Networks*. In Proc. of the Sixth Intl. Workshop on Network-Based Information Systems (NBIS2003), Workshop at DEXA 2003, Prag, Tschechien, September 2003.
- [KKO03d] Michael Klein, Birgitta König-Ries und Philipp Obreiter. *Stepwise Refinable Service Descriptions: Adapting DAML-S to Staged Service Trading*. In Proc. of the First Intl. Conference on Service Oriented Computing, S. 178–193. Trento, Italien, Dezember 2003.
- [KK04] Michael Klein und Birgitta König-Ries. *Combining Query and Preference – An Approach to Fully Automate Dynamic Service Binding*. Submitted to IEEE International Conference on Web Services (ICWS 2004).
- [LO03] Lei Liu und Philipp Obreiter. *The Software Station - A System for Version Controlled Development and Web Based Deployment of Software for a Mobile Environment*. In Proc. of the National Data Base Conference (NDBC'03). Changsha, China, 2003. Erweiterte Version ist als Interner Bericht TR2003-16 veröffentlicht.
- [NEK03] Khaled Nagi, Iman Elghandour und Birgitta König-Ries. *Mobile Agents for Locating Documents in Ad Hoc Networks*. In Workshop on Agents and Peer-to-Peer Computing at AAMAS 2003. Melbourne, Australien, Juli 2003.
- [Obr04] Philipp Obreiter. *A Case for Evidence-Aware Distributed Reputation Systems*. In Second International Conference on Trust Management (iTrust'04). Oxford, UK, 2004.
- [OK03] Philipp Obreiter und Michael Klein. *Vertical Integration of Incentives for Cooperation - Inter-Layer Collaboration as a Prerequisite for Effectively Stimulating Cooperation in Ad Hoc Networks*. In Proc. of the Second Mediterranean Workshop on Ad-Hoc Networks (MED-HOC NET 2003). Mahdia, Tunesien, Juni 2003.

- [OKK03b] Philipp Obreiter, Birgitta König-Ries und Michael Klein. *Stimulating Cooperative Behavior of Autonomous Devices - An Analysis of Requirements and Existing Approaches*. In Proceedings of the Second International Workshop on Wireless Information Systems (WIS2003), S. 71–82. Angers, Frankreich, April 2003.
- [OKP04] Philipp Obreiter, Birgitta König-Ries und Georgios Papadopoulos. *Engineering Incentive Schemes for Ad Hoc Networks*. In First International Workshop on Pervasive Information Management (EDBT-Workshop). Heraklion, Greece, März 2004.
- [ON03] Philipp Obreiter und Jens Nimis. *A Taxonomy of Incentive Patterns - The Design Space of Incentives for Cooperation*. In Proceedings of the Second Intl. Workshop on Agents and Peer-to-Peer Computing (AP2PC'03), Springer LNCS 2872. Melbourne, Australien, Juli 2003. Erweiterte Version ist als Interner Bericht TR2003-9 veröffentlicht.

#### Technische Berichte aus der ersten Phase des geförderten Projektes

- [AO03] Ralf Anders und Philipp Obreiter. *Economic Incentive Patterns and their Application to Ad Hoc Networks*. Technischer Bericht Nr. 2003-17, Universität Karlsruhe, Fakultät für Informatik, Oktober 2003.
- [FO04] Stefan Fähnrich und Philipp Obreiter. *The Buddy System - A Distributed Reputation System Based On Social Structure*. Technischer Bericht Nr. 2004-1, Universität Karlsruhe, Fakultät für Informatik, Februar 2004.
- [KHMM04] Michael Klein, Markus Hoffman, Daniel Matheis und Michael Müssig. *Comparison of Overlay Mechanisms for Service Trading in Ad Hoc Networks*. Technischer Bericht Nr. 2004-2, Universität Karlsruhe, Fakultät für Informatik, Februar 2004.
- [KKO03a] Michael Klein, Birgitta König-Ries und Philipp Obreiter. *Lanes – A Lightweight Overlay for Service Discovery in Mobile Ad Hoc Networks*. Technischer Bericht Nr. 2003/6, Universität Karlsruhe, Fakultät für Informatik, Mai 2003.
- [Kle03] Michael Klein. *DIANEmu – A Java-Based Generic Simulation Environment for Distributed Protocols*. Technischer Bericht Nr. 2003/7, Universität Karlsruhe, Fakultät für Informatik, Mai 2003.
- [OKK03a] Philipp Obreiter, Birgitta König-Ries und Michael Klein. *Stimulating Cooperative Behavior of Autonomous Devices - An Analysis of Requirements and Existing Approaches*. Technischer Bericht Nr. 2003-1, Universität Karlsruhe, Fakultät für Informatik, Januar 2003.

#### Diplom- und Studienarbeiten aus der ersten Phase des geförderten Projektes

- [And03] Ralf Anders. *Anreizmuster und Ihre Einsatzmöglichkeiten in Mobilen Ad-Hoc-Netzen*, 2003. Diplomarbeit an der Fakultät für Informatik, Universität Karlsruhe.
- [Bre02] Tobias Breyer. *Entwicklung einer geräteunabhängigen Benutzeroberfläche für DIANE*, Juni 2002. Studienarbeit an der Fakultät für Informatik, Universität Karlsruhe.
- [Bre03] Tobias Breyer. *Modellierung der Bewegung und des Verhaltens von Dienstmutzern in mobilen Ad-hoc-Netzen*, Oktober 2003. Diplomarbeit an der Fakultät für Informatik, Universität Karlsruhe.
- [Deh04] Jan Dehner. *Ein Transportsystem für die Übertragung der Protokolle aus DIANEmu auf reale Geräte*, März 2004. Studienarbeit an der Fakultät für Informatik, Universität Karlsruhe.

- [Dom04] Roland Domagalski. *Möglichkeiten der Anfragebearbeitung in mobilen Ad-hoc-Netzen*, März 2004. Bachelorarbeit an der Fakultät für Informatik, Technische Universität München.
- [Fäh03] Stefan Fähnrich. *Entwurf eines vollkommen verteilten Reputationssystems für Ad-Hoc Netze*, November 2003. Studienarbeit an der Fakultät für Informatik, Universität Karlsruhe.
- [Fis04] Thomas Fischer. *Entwicklung eines Kontainermanagers zur Unterstützung der Konfigurierung und Ausführung semantisch beschriebener Dienste*, März 2004. Studienarbeit an der Fakultät für Informatik, Universität Karlsruhe.
- [Guo02] Chuan Guo. *Konzeption und prototypische Implementierung eines auf JXTA basierenden Systems zur integrierten Nutzung von Lernmaterialien*, Juni 2002. Studienarbeit an der Fakultät für Informatik, Universität Karlsruhe.
- [Koo02] Erik Koop. *Konzeption eines auf lokalen, hierarchisch organisierten Ringen basierenden Dienstsucheverfahrens*, Juli 2002. Studienarbeit an der Fakultät für Informatik, Universität Karlsruhe.
- [Liu03] Lei Liu. *Schaffung einer internetbasierten Softwaretankstelle zur Herstellung der technischen Voraussetzungen für die Teilnahme am DIANE Ad-hoc Netzwerk*, Mai 2003. Studienarbeit an der Fakultät für Informatik, Universität Karlsruhe.
- [Pap03] Georgios Papadopoulos. *S-Lanes - Ein schlankes Overlay zur motivierten Dienstvermittlung in mobilen Ad-Hoc-Netzen*, Dezember 2003. Diplomarbeit an der Fakultät für Informatik, Universität Karlsruhe.
- [Pon02] Eduardo Ponz. *An Approach to Support Service Discovery in Mobile Ad-hoc Networks – Concept and Simulation*, Juli 2002. Studienarbeit an der Fakultät für Informatik, Universität Karlsruhe.
- [Zhe03] Ting Zheng. *Implementierung eines webbasierten Werkzeuges zur Instanziierung von Dienstbeschreibungen*, Mai 2003. Studienarbeit an der Fakultät für Informatik, Universität Karlsruhe.

#### Weitere eigene Veröffentlichungen

- [DK04] Roland Domagalski und Birgitta König-Ries. *Möglichkeiten der Anfragebearbeitung in mobilen Ad-hoc-Netzwerken*, 2004. Workshop Grundlagen und Anwendungen mobiler Informationstechnologie, Heidelberg, März 2004.
- [HSS04] Thorsten Höllrigl, Christoph Schaa und Frank Schell. *Entwicklung einer formalen Repräsentation für die DIANE-Dienstbeschreibung*, Mai 2004. Studienarbeit an der Fakultät für Informatik, Universität Karlsruhe (in Arbeit).
- [KHMM] Michael Klein, Markus Hoffman, Daniel Matheis und Michael Müssig. *DIANEmu Manual*. Aktuelle Version unter <http://www.ipd.uni-karlsruhe.de/DIANE/docs/DIANEmu-Manual.pdf>.
- [KK02c] Michael Klein und Birgitta König-Ries. *An Ontology-Based Document Space as an Adaptable User Interface for Mobile Information Systems*. In Proceedings des Mini-Workshop "Mobile Datenbanken und Informationssysteme - Datenbanktechnologie überall und jederzeit", Otto-von-Guericke Universität Magdeburg, März 2002.

#### Weitere Referenzen

- [Aso98] N. Asokan. *Fairness in Electronic Commerce*. Dissertation, University of Waterloo, 1998.

- [DAML] Defense Advanced Research Projects Agency (DARPA). *DARPA Agent Markup Language (DAML)*. <http://www.daml.org/>.
- [DAMLS] Defense Advanced Research Projects Agency. *DARPA Agents Markup Language - Services (DAML-S)*. <http://www.daml.org/services/>.
- [ELAN] Institut für Informatik, Universität Frankfurt. *E-Learning in Ad-hoc Netzen (ELAN) (Projekt im SPP 1140)*. <http://elan.wox.org/>.
- [FIPA] FIPA OS. *The FIPA-OS Agent Platform*. <http://fipa-os.sourceforge.net>.
- [SCORE] Institute for Program Structures and Data Organization, Universität Karlsruhe. *A System for Courseware Reuse (SCORE)*. <http://www.ipd.uni-karlsruhe.de/SCORE>.
- [KRASH] Institute for Program Structures and Data Organization, Universität Karlsruhe. *KRASH Project*. <http://www.ipd.uka.de/KRASH/>.
- [JADE] Jade. *Java Agent DEvelopment Framework*. <http://sharon.cselt.it/projects/jade/>.
- [KLW95] Michael Kifer, Georg Lausen und James Wu. *Logical Foundations of Object-Oriented and Frame-Based Languages*. *Journal of the ACM*, 42(4):741–843, July 1995.
- [JXTA] Project JXTA. *Sun Microsystems*. <http://www.jxta.org/>.
- [MARB] Arbeitsgruppe Verteilte Systeme, Universität Marburg. *Skalierbare kontextabhängige Routingalgorithmen für selbstorganisierende Mobilfunknetze (Projekt im SPP 1140)*. <http://ds.informatik.uni-marburg.de/de/research/laufende.php>.
- [QUAL] Qualnet. *Network Simulation Software*. <http://www.scalable-networks.com/>.
- [RT99] E. Royer und C. Toh. *A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks*. *IEEE Personal Communications*, April 1999.
- [UIML] UIML.Org. *User Interface Markup Language (UIML)*. <http://www.uiml.org>.
- [OWL] World Wide Web Consortium. *Web Ontology Language (OWL)*. <http://www.w3.org/TR/owl-ref/>.