



Technische Universität Darmstadt

Department of Electrical Engineering and Information Technology
Department of Computer Science (Adjunct Professor)
Multimedia Communications Lab
Prof. Dr.-Ing. Ralf Steinmetz

Referenzszenarien für Peer-to-Peer-Systeme

Technical Report

von

**Ralf Steinmetz, Oliver Heckmann, Nicolas Liebau, Alejandro
Buchmann, Claudia Eckert, Jussi Kangasharju, Max Mühlhäuser,
Andreas Schürr**

am
6. Mai 2006

KOM-TR-2006-04

Inhaltsverzeichnis

| | |
|--|----|
| Inhaltsverzeichnis..... | 2 |
| 1 Einführung..... | 3 |
| 2 P2P-basiertes Kommunikationssystem für den Katastropheneinsatz..... | 4 |
| 2.1 Motivation und Übersicht..... | 4 |
| 2.2 Charakteristika des Referenzszenarios..... | 7 |
| 3 Integration von Softwareentwicklungswerkzeugen mit P2P-Techniken..... | 11 |
| 3.1 Motivation und Übersicht..... | 11 |
| 3.2 Skizze eines Prototypen..... | 16 |
| 4 Zusammenfassung..... | 18 |
| Referenzen..... | 19 |
| | |
| Abbildung 1: Kommunikationssystem für den Katastropheneinsatz..... | 4 |
| Abbildung 2: Integration von Softwareentwicklungswerkzeugen mit P2P-Techniken..... | 14 |
| Abbildung 3: Angaben zu den Qualitätsanforderungen der Referenzszenarien..... | 18 |

1 Einführung

Peer-to-Peer-Systeme (P2P-Systeme) gewinnen in der Forschung zunehmend an Bedeutung. Bislang sind Dateitauschbörsen die häufigste Anwendung für P2P-Systeme. Dateitauschbörsen haben nur einen geringen gesellschaftlichen Nutzen und stellen keine hohen Qualitätsansprüche an die ihnen zugrunde liegenden Systeme. Um den breiteren Einsatz von P2P-Technologie zu bewerten und voranzutreiben sind daher „seriösere“ Anwendungen mit höheren Qualitätsanforderungen nötig. In diesem Technischen Report stellen wir zwei Referenzszenarien für P2P-Systeme vor, die diese Ansprüche erfüllen. Die Referenzszenarien zeichnen sich durch hohe und zueinander weitgehend orthogonale Qualitätsanforderungen aus. Das erste Szenario behandelt den Einsatz von P2P-Technologien für die Koordination von Rettungskräften bei einem Katastrophenfall. Der Nutzen von P2P ist hier leicht ersichtlich. Das zweite Szenario beschreibt den potentiellen Einsatz von P2P-Technologien zur Unterstützung eines global verteilten Softwareentwicklungsprozeß. Der Einsatz von P2P für diesen Zweck liegt auf den ersten Blick nicht unbedingt nahe, da zentralisierte Systeme (Client-Server) für diese Zwecke große Vorteile versprechen. Wir zeigen in diesem Dokument, dass dies nicht zwangsweise der Fall ist und dass aus der Umsetzung dieses Referenzszenarios besonders viel über die Einsatzmöglichkeiten von P2P gegenüber Client-Server gelernt werden kann.

Beide Referenzszenarien werden im Rahmen der DFG Forschergruppe QuaP2P (www.quap2p.de) von uns untersucht und umgesetzt.

2 P2P-basiertes Kommunikationssystem für den Katastropheneinsatz

2.1 Motivation und Übersicht

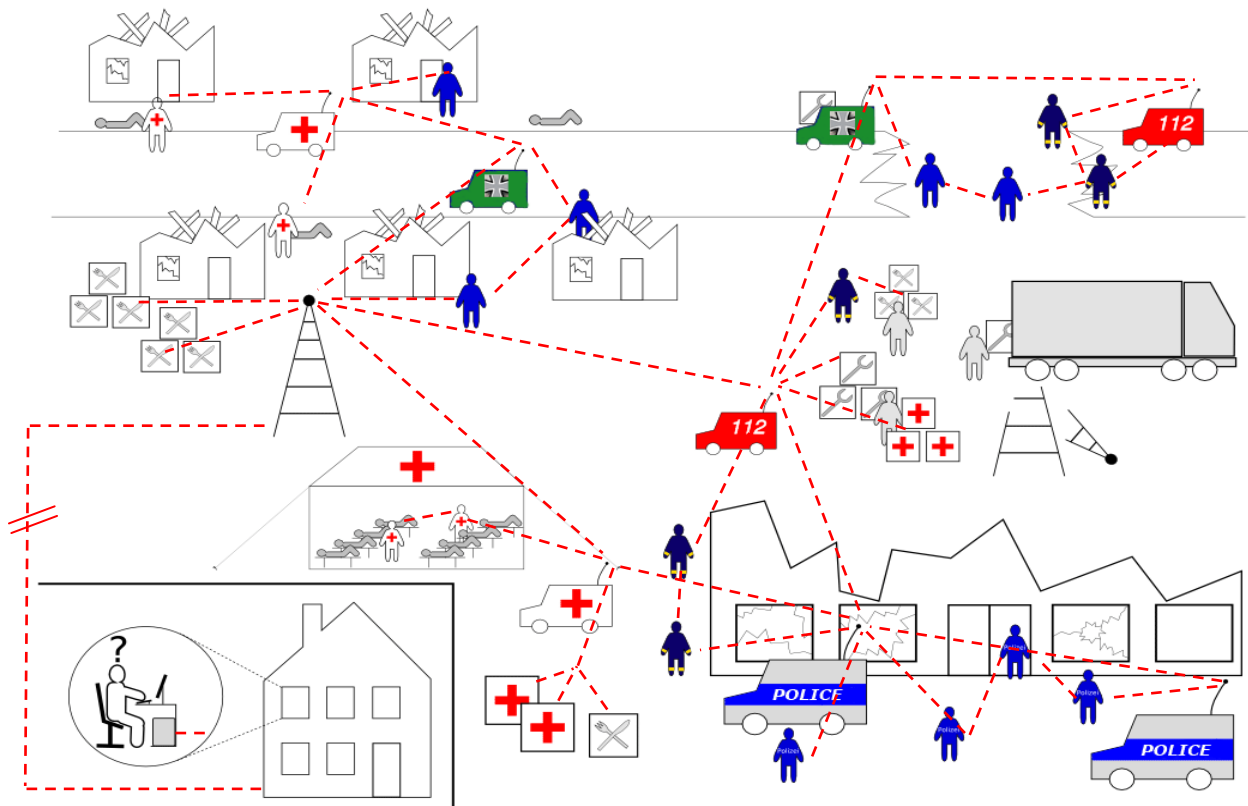


Abbildung 1: Kommunikationssystem für den Katastropheneinsatz

Für das Verständnis sei zunächst der Begriff Katastrophe näher erläutert. Katastrophe wird hier im Sinne des inzwischen breit eingeführten angelsächsischen Terms *incident* verstanden; er reicht von derjenigen Klasse von Feuerwehreinsätzen, bei denen mehrere Züge (und andere beteiligte Rettungskräfte) koordiniert werden müssen, bis zu Großereignissen wie Naturkatastrophen (Erdbeben, Orkane, Tsunamis etc.), industriell verursachten Katastrophen (Grubenunglücke, Chemie-, Reaktorunfälle, etc.) und Terroranschlägen bzw. Kriegsfolgen. Im Fokus des Szenarios steht die unmittelbare Rettung, Bergung und Sicherung der „ersten Stunden“, die betroffenen Helfer seien als Ersthelfer bezeichnet. Folgende Überlegungen führten zur Auswahl des Referenzszenarios „Katastropheneinsatz“ (engl.: *incident response*; auch, aber seltener: *emergency response*).

- *Allgemeine Bedeutung:* Die Informatik hält seit Jahren immer stärker Einzug in verschiedene Bereiche des Katastropheneinsatzes. Schon seit den 70er Jahren sind „Incident Management“-Systeme im Einsatz, seit einigen Jahren wird an für den Realzeitbetrieb tauglicheren Systemen gearbeitet. Im Zusammenhang mit den Terroranschlägen des 9. September 2001 („Nine-Eleven“) und der Tsunami-Katastrophe von Weihnachten 2004 stiegen nicht nur das Bewusstsein für das Gefahren- und Bedrohungspotenzial und für Unzulänglichkeiten der Katastrophenhilfe (insbesondere jener der „ersten Stunden“) sehr stark an, sondern - nicht unerheblich - auch die Mittel zur Förderung von Forschung und Entwicklung beim amerikanischen DHS (*Department of Homeland Security*).

- *P2P-Charakteristika:* Drei Überlegungen bzw. Entwicklungen legen nahe, dass im Referenzszenario Katastropheneinsatz Eigenschaften von P2P-Systemen und -Mechanismen wünschenswert sind:
 1. Die hinlänglich bekannte Schwierigkeit, die einzelnen gut organisierten/ eingespielten Rettungsdienste in ihrer *Gesamtheit* übergeordnet zu organisieren und zielführend zu koordinieren: die teilweise extremen Echtzeitbedingungen (brennende Häuser, eingeklemmte Verletzte etc.) legen hier schon allgemein den IT-Einsatz nahe; spezieller gilt: die immer wieder neu zusammengewürfelte Besetzung (beeinflusst durch den Ort des Geschehens und - bei ggf. eingeschränkter Infrastruktur - Nähe und möglicher Zeitpunkt des Eintreffens von Rettungsdiensten), je nach Schwere und Besonderheiten (d.h. erforderlicher Expertise) auch unterschiedliche Anzahl und fachliche Zusammensetzung. All das sind Charakteristika hoch dynamischer *Anwendungs*-Konfigurationen, die denen herkömmlicher P2P-Systeme sehr nahe kommen.

 2. Die hohe Wahrscheinlichkeit hochdynamischer *Infrastruktur*-Konfigurationen: häufig werden bei den genannten Ereignissen Kommunikationsnetze beschädigt, wobei meist - als Bestandteil des Katastropheneinsatzes - die sukzessive Reparatur schon in den „ersten Stunden“ beginnt; breit einsetzbare IT-Lösungen sollten daher ohne Festnetz-Infrastruktur (ad-hoc verbunden) auskommen, eine sukzessiv verbesserte oder intermittierend fehleranfällige Infrastruktur aber auch möglichst gut ausnutzen - auch dies sind Eigenschaften, welche den Eigenschaften von P2P-Systemen nicht unähnlich sind, denn diese sind auf Betrieb ohne „Server“ und ohne verlässlich verfügbare Partner ausgerichtet.

3. Die zunehmende Bedeutung von Subnetzen mit intrinsischem P2P-Charakter: die Zuverlässigkeit und Leistungsfähigkeit vernetzter Mikrosysteme (tragbare Endgeräte als Teil der Rettungsausrüstung, Sensornetzwerke sowie *smart tags*, also kommunizierende Identifikatoren z.B. auf der Basis von Ultra-WideBand UWB, etc.) hat einen Stand erreicht, der ihren Einsatz sowohl im Vorfeld (Gebäude-/Geländeausstattung) als auch im unmittelbaren Rettungseinsatz sehr attraktiv macht. Typisch für diese Systeme ist erstens die kurze Kommunikationsreichweite bei großer Zahl von Knoten (Sensoren), welche P2P-artiges Verhalten bei der Informationsvermittlung nahe legt, zweitens kooperative Ableitung höheren „Wissens“ durch Propagation und Aggregation, wobei veränderliche und unzuverlässige Kommunikationsstrukturen toleriert werden müssen. Besonders attraktiv - und z.B. über das Kerngebiet der Sensornetzwerk-Forschung hinausgehend - ist im Antragszusammenhang die Herausforderung, die schon für sich genommen P2P-„artigen“ Subnetze in weitergehende vernetzte Anwendungen auf höheren Ebenen zu integrieren, welche gemäß Punkt 1 und 2 ebenfalls P2P-Charakteristika aufweisen.
- *Vielseitige Herausforderungen an Qualitätsmerkmale:* Bei näherer Betrachtung der Einsatzszenarien lassen sich hohe Qualitätsanforderungen an P2P-Systeme für den Katastropheneinsatz unmittelbar ableiten [TRQm]. Alle von der Forschergruppe vorgeschlagenen Qualitätsmerkmale sind betroffen, wenn auch nicht in jedem denkbaren Szenario gleich stark. Folgende bereits genannten Charakteristika führen unmittelbar zu hohen Qualitätsanforderungen: Realzeit-Bedingungen, hochdynamische Knoten-/Verbindungs-Topologie mit teils intermittierend verfügbaren Komponenten, anwendungsseitig ebenfalls hohe Dynamik hinsichtlich Informationsquellen und -verarbeitungselementen, verknüpften Teilsystemen (Rettungsorganisationen) sowie verfolgten Zielen.

Im Folgenden wird über die offensichtliche Bedeutung von Sprachkommunikation hinaus behauptet, dass auch andere Medienstromklassen für den Katastropheneinsatz wichtig seien. Da dies nicht unmittelbar einsichtig sein mag, sei - beispielhaft - der Abdruck eines Kommissionsberichtes zum Ablauf der Ersthelfer-Einsätze beim Terroranschlag auf das World Trade Center als Argumentationshilfe herangezogen [Nat02]. Bekanntlich haben dabei allein von der New Yorker Feuerwehr fast 350 Helfer ihr Leben verloren. Im Bericht wird neben der „so gut wie nicht existierenden“ übergeordneten Koordination der Rettungskräfte (vgl. Pkt. 1 oben) und dem gravierenden Informationsmangel aufgrund zusammengebrochener Netze (vgl. Pkt. 2) ganz besonders beklagt, dass Videoverbindungen fehlten, weshalb die Helfer vor Ort wesentlich schlechtere Gesamtsicht auf das Geschehen hatten als das weltweite Fernsehpublikum. Die durch Rauch und Staub verheerend schlechte Umgebungssicht machte dieses

Fehlen von Videoverbindungen teilweise lebensgefährlich. Die Forderung nach Videokommunikation kommt hier also unmittelbar von Betroffenen. Schließlich wird berichtet, dass nach dem Einsturz der Türme die Hand nicht vor den Augen zu sehen gewesen sei. In solchen Situationen wären 3D-Gelände-/Gebäudedarstellungen, in Realzeit an Datenbrillen übertragen, eine ggf. lebenswichtige Neuerung, wodurch Live-3D-Simulationen als dritte Medienstrom-Klasse elementare Bedeutung gewinnen. Dieselbe Situation rückt auch die Bedeutung von Lokationskontext ins Zentrum, denn etliche Forschungsaufbauten haben inzwischen gezeigt, dass von Rettungsleitständen aus via Sprechverbindung ein „Lotsendienst“ sinnvoll eingerichtet werden kann, wenn z.B. an Displays im Leitstand die Gelände-/Gebäudetopologie und der aktuelle Aufenthaltsort der zu lotsenden Person (Rettungshelfer) dargestellt werden kann. Entsprechende Systeme sind also so wenig Fiktion wie Datenbrillen für die Feuerwehr, vgl. den Abschnitt zu verwandten Arbeiten.

Nach allen vorgenannten Erläuterungen sollte klar sein, dass P2P-Mechanismen an vielen Stellen als viel versprechend angesehen werden können, z.B. als Grundlage des Kommunikationssystems für Ersthelfer, wo es dezentral, unabhängig von zentraler Infrastruktur und bei geeigneter Konzeption selbst organisierend realisiert werden kann. Die Forderung nach Unterstützung von mindestens Text-, Sprach- und Videokommunikation in Echtzeit, ggf. 3D-Simulationen, wurde motiviert. Bezieht man die zunehmende Bedeutung von vernetzten Mikrosystemen ein, welche die Forderung nach Skalierbarkeit nach sich ziehen, so sind alle vier Untermerkmale von Adaptivität im Lichte der obigen Ausführungen von offensichtlicher Bedeutung. Wegen der Echtzeitanforderung interaktiver Sprachkommunikation, den im Vergleich zur reinen Datenkommunikation relativ hohen Datenmengen bei Videokommunikation und des ggf. stark schwankenden Bandbreitenbedarfs von 3D-Simulationen werden hohe Ansprüche an die *Effizienz* gestellt. Ferner sind hohe Anforderungen an die Dienstvermittlung/Lokalisierung als Untermerkmal des Qualitätsmerkmals *Validität* gestellt, da z.B. hochgradig instabile bzw. sich stark verändernden Topologien den Zugriff auf (valide) Daten/Objekte erschweren.

2.2 Charakteristika des Referenzszenarios

Als nächstes werden einige wesentliche sowie einige optionale Bestandteile bzw. Charakteristika eines P2P-Systems für den Katastropheneinsatz festgelegt. Wesentliche Punkte sind: i) eine übergeordnete Koordination, welche wie alle anderen Teilstrukturen an fehlende, intermittierende und sukzessive verbesserte Infrastruktur anpassbar ist; ii) eine beliebige, a priori unbekannte und dynamisch variable Zahl von Vor-Ort-Netzen für die einzelnen Rettungsdienste, welche mit anderen Vor-Ort-Netzen und den logisch darüber und darunter liegenden Netzen (vgl. i) und iii)) kooperieren; iii) mehrere Typen vernetzter Mikrosysteme, welche Einsatzfeld, Personal und Hilfsmittel zu koordinieren und lokalisieren helfen und

hohe Ansprüche an Skalierbarkeit induzieren; darüber hinaus werden hohe Ansprüche an die Datenverbindungen insbesondere von den zu unterstützenden Medienstrom-Klassen abgeleitet. Abbildung 1 verdeutlicht das Zusammenwirken dieser Punkte.

Die hohe Dynamik in Anwendung und Netz wirkt sich auf alle Komponenten aus und legt P2P-Ansätze besonders nahe. Zur Konkretisierung der Echtzeitanprüche wird das Szenario im Rahmen der Arbeiten eingegrenzt auf konkrete Katastrophen (Zugunglück, Erdbeben, etc., inkl. Zahlen und Randbedingungen) und weiter heruntergebrochen auf Aspekte der Dynamik (Ausfälle/Wiederanläufe, erlaubte Zeitintervalle, etc.) bis hin zu konkreten Problemstellungen („Trupp 14 benötigt in Gebäude A zwei Atemschutzgeräte der Klasse AS4“ o.ä.).

Die folgende Auflistung orientiert sich wesentlich an den eingangs beschriebenen drei Dimensionen der P2P-„Artigkeit“ von Katastropheneinsatz- Systemen, nämlich der hochdynamischen Anwendung, der hochdynamischen Topologie und den vernetzten Mikrosystemen als Sub-Komponenten.

- Hochdynamische Anwendung: Einsatzkoordination. Wie oben bereits erwähnt, geht der Einsatz von IT-Systemen zur Koordination von Rettungskräften in die 70er Jahre zurück. Systeme aus dieser Zeit waren auf einen Rettungsdienst beschränkt (Polizei, Feuerwehr, ...) und bezogen ihre Realzeit-Fähigkeit im Wesentlichen aus der menschlichen Systemkomponente: das Bedienpersonal, d.h. Mitarbeiter in der Einsatzleitung, interpretierten ggf. Systemergebnisse in Echtzeit auf das gebotene Einsatzszenario. Unter Schlagwörtern wie „Unified Incident Command and Decision Support (UICDS)“ wird inzwischen an Systemen geforscht und entwickelt, welche sowohl die (gemäß vorigen Erläuterungen besonders spannende) Koordination mehrerer, ad hoc kooperierender Rettungsdienste ermöglichen als auch Realzeit- Bedingungen einhalten. Der Verzicht auf verlässliche Festnetz-Infrastrukturen, der P2P-Ansätze besonders nahe legen würde, ist in diesen Systemen allerdings erst in Ansätzen angedacht; diese Thematik wird sehr viel stärker in militärischen Anwendungen betrachtet, nämlich der Einsatzleitung bei Kampfeinsätzen.
- Hochdynamische Topologie: Vor-Ort-Vernetzung. Im Gegensatz zur eben genannten übergreifenden Einsatzkoordination wird im Bereich der Vor-Ort-Vernetzung in jüngster Zeit sehr wohl über infrastrukturlose Netze nachgedacht. Daher erstaunt nicht, dass hier bereits der Einsatz von P2P-Technologie untersucht wird, wie er z.B. in einer Fallstudie für ein Erdbebenszenario [HIST93] beschrieben ist. Alle aus der Literatur bekannten Versuche, die Vor-Ort-Vernetzung im Katastropheneinsatz mittels P2P-Mechanismen zu unterstützen, sind allerdings in zwei wesentlichen Punkten nicht mit dem hier vorgeschlagenen Vorgehen vergleichbar: erstens werden – aufgrund des Anwendungsfokus der Projekte kaum verwunderlich - spezifisch zugeschnittene Systeme entwickelt, und zwar weitgehend von Grund auf neu, anstatt wie hier vorgeschlagen

zuschneidbare Mechanismen einzusetzen, welche in wieder verwendbare Komponenten eingebettet sind; zweitens werden nicht drei Ebenen logischer Netze betrachtet (Gesamtkoordination, Vor-Ort-Vernetzung innerhalb und zwischen Rettungsdiensten, Einbeziehung untergeordneter vernetzter Mikrosysteme) und über P2P-Mechanismen kombiniert.

- Vernetzte Mikrosysteme, Typ 1: im Gebäude/Gelände installierte, drahtlos kommunizierende Sensoren. Vor allem die Gruppe um das NEST-Labor an der Universität Berkeley (UCB) hat die an der UCB entwickelten, inzwischen kommerziell erhältlichen und vielseitig einsetzbaren „Mote“-Sensorsysteme in etlichen Zusammenhängen untersucht, die mit Katastropheneinsatz in unmittelbarem Zusammenhang stehen. Bekannt sind vor allem Arbeiten zur Untersuchung von Gebäudestrukturveränderungen, welche u.a. auf verringerte Stabilität bzw. erhöhte Einsturzgefahr hinweisen [KSRS04]. Die Systeme wurden im Mai 2005 im Rahmen des „Golden Gate“-Projekts an der gleichnamigen Brücke angebracht. Werden solche Systeme zum Zwecke der Langzeit-Beobachtung (Einfluss von Erdbeben, Verwitterung, etc.) angebracht, so können sie im Katastrophenfall auch Hinweise auf besonders einsturzgefährdete Gebäude/-teile geben; die verbleibende Zeit zur Rettung kann unter Umständen relativ zuverlässig aus vernetzten Motes ermittelt werden; so können Ersthelfer einerseits aus überkritischen Bereichen abgezogen, andererseits in hochkritische Bereiche vordringlich verlagert werden. Studentische Arbeiten an der UCB haben auch untersucht, wie ein mit Motes überzogenes Gebäude „intelligenten“ Brandschutz erhalten kann, wenn auch Rauch- bzw. Feuersensoren auf den Motes angebracht sind: ohne zentrale Infrastruktur können sie kürzeste Rettungswege, ggf. mit verschiedenen Gefahrenstufen (grün und gelb, im Gegensatz zu rot: gesperrt) kooperativ ermitteln, welche die Evakuierung des Gebäudes ermöglichen.
- Vernetzte Mikrosysteme, Typ 2: um Lokationskontext erweiterte, flächendeckende Mobilkommunikation. Auf der innerstädtischen Ebene gewinnt inzwischen in den USA die flächendeckende Einrichtung von Drahtlos-Netzwerken (wLan oder „wiMax“ [WiMax]) zunehmende Bedeutung. Als Router fungieren beispielsweise „Access Points“ auf Laternenmasten. In Corpus Christi, Texas, wird die schon in anderen Städten pilotierte Ausrüstung aller Rettungsfahrzeuge mit entsprechenden Endgeräten im „Digital Cities“-Projekt weitergedacht: so wird z.B. der in solchen Netzen relativ leicht zu ermittelnde Lokationskontext (von Rettungsfahrzeugen) nicht erst beim eigentlichen Einsatz (Ampelsteuerung, Navigationssystem, etc.) genutzt, sondern bereits im Vorfeld, wenn Muster im Straßenverkehr Hinweise auf deutlich erhöhte Unfallgefahr geben. Firmen wie Eckerhau vertreiben Ergänzungen für tragbare Endgeräte, welche die Signalstärke-basierte Positionierung der Geräte deutlich

erleichtern und verbessern. Heute sind solche Systeme insbesondere unter Tage (Tunnelbau) und in Kliniken im Einsatz, wo sie neben der Lokalisierung von Personen vor allem diejenige von (meist teuren, daher nicht in hoher Zahl vorhandenen) Geräten erlauben (Stichwort: *asset tracking*).

- Vernetzte Mikrosysteme, Typ 3: UWB-basierte kommunizierende Etiketten. Allgemein befinden sich diese Systeme, in der Größe einer Geldmünze, von allen genannten noch am stärksten im Forschungsstadium, so dass der Einsatz im Katastropheneinsatz nicht erforscht ist, sondern im Wesentlichen den Visionen von Firmen wie Aetherwire (www.aetherwire.com) entnommen werden müssen. Es zeigt sich aber deutlich, dass der Einsatz über die für Typ 2 diskutierte (dort außerdem: kommunizierende Endgeräte *ergänzende*) Ermittlung von Lokationskontext hinausgeht. UWB-Etiketten lassen sich beispielsweise „wie die Brotkrumen bei Hänsel und Gretel“ bzw. wie der Ariadnefaden verwenden, um Pfade von Personen oder beweglichen Objekten nach- oder zurückzuverfolgen. Das eröffnet weitere Katastropheneinsatz-Felder wie die Suche von Lawinopfern (entsprechende, z.B. automatisierte „Ausstreuung“ vorausgesetzt). Wesentlich vielseitigere Möglichkeiten ergeben sich aber vor allem dadurch, dass sehr geringe Kosten und Größe einen Umfang von UWB-Tag-Netzwerken realistisch erscheinen lassen, der vorher nur für passiven Etiketten (RFID-Tags) denkbar schien - welche von einer Infrastruktur von Lesegeräten abhängen. Der Katastropheneinsatz geht dadurch ebenfalls weit über das oben genannte Beispiel von „Spurensuche“ hinaus.
- Vernetzte Mikrosysteme, Typ 4: Tragbare Endgeräte als Teil von Rettungsausrüstungen. Das Potenzial vernetzter „Wearables“ im Katastropheneinsatz wird unter anderem in der Studie [Sla03] untersucht und ist Gegenstand vieler aktueller Projekte. Wie so oft kann auch ein Blick in die Landschaft der Spielkonsolen einen Eindruck vorstellbarer Möglichkeiten geben, beispielsweise das PS2-Spiel „Firefighter F.D. 18“ [Onl04], welches eine Datenbrille für den „Spielhelden“ zugrundelegt.

3 Integration von Softwareentwicklungswerkzeugen mit P2P-Techniken

3.1 Motivation und Übersicht

An der Durchführung großer Softwareentwicklungsprojekte sind heutzutage oft hunderte bis tausende von Personen beteiligt, die weltweit verteilt arbeiten und verschiedenen Organisationen angehören. Mehr und mehr wird dabei sogar in Schichten „rund um die Uhr“ an denselben Entwicklungsdokumenten gearbeitet [BDV01]. Bei ihrer Arbeit setzen die beteiligten (Software-) Ingenieure oft dutzende verschiedener Werkzeuge (*CASE-Tools*) ein. Aus nahe liegenden Gründen ist in einem solchen Szenario weniger die korrekte Durchführung einzelner Entwicklungsaufgaben, als vielmehr die Koordination der verschiedenen Teilaktivitäten und die Konsistenzhaltung der entstehenden Dokumente (in ihren verschiedenen Versionen und Varianten) eines der Hauptprobleme bei der Etablierung eines „reifen“ Softwareentwicklungsprozesses. Unterstützung versprechen hierbei so genannte *kollaborative Softwareentwicklungsumgebungen*, die die Funktionalitäten „klassischer“ integrierter Softwareentwicklungsumgebungen (Project Support Environments) mit CSCW-Techniken kombinieren. Besonders augenscheinlich wird die zunehmende Bedeutung dieses Themenfeldes durch jüngst ins Leben gerufene Workshops wie dem „International Workshop on Distributed Software Development (DiSD)“ und dem „Workshop on Cooperative Support for Distributed Software Engineering Processes (CSSE)“. Die auch in den „Call for Papers“ dieser Workshops zum Ausdruck kommenden Anforderungen an solche Umgebungen werden beispielsweise in [ALG+04, RBS+03], dem Entwurf der kollaborativen Softwareentwicklungsumgebung GENESIS aus Benutzersicht dargestellt. Gewünscht werden u.a. Dienste zur globalen und lokalen Entwicklungsprozesskontrolle, zur Behandlung von Konflikten im Entwicklungsprozess, zur Qualitätsüberwachung auf Basis von Metriken, zur Verwaltung von Dokument-Repositories, zur Kommunikation und Koordination sowie zur Aufzeichnung aller Entwicklungsaktivitäten. Ähnliche Ausführungen findet man in Arbeiten, die sich mit CSWC- bzw. Hypermedia-Unterstützung für Engineering-Prozesse in anderen Anwendungsdomänen auseinandersetzen [Ind, BDV01, CSN98].

Betrachtet man nun den „State of the Art“ der Integration von Softwareentwicklungswerkzeugen sowie der Unterstützung geographisch verteilter Entwicklungsmannschaften - siehe etwa [Bou04, DS04, DS05] - so stellt man fest, dass die dabei zum Einsatz kommenden Integrationsmechanismen wie gemeinsame Repositories oder Konfigurationsmanagementsysteme und Publish-Subscribe-Mechanismen fast ohne Ausnahme auf *zentralen Servern* basieren. Gute Beispiele hierfür liefern die Serverfarmen und Dienste, wie sie etwa von Sourceforge (www.sourceforge.net) und Collabnet (www.collab.net) zur

Verwaltung von „Open-Source-Software-“, und kommerziellen Software-Projekten angeboten werden. Solche allein auf zentralen Serverdiensten aufsetzenden Entwicklungsplattformen stoßen jedoch bei der Abwicklung großer kommerzieller Projekte an ihre Grenzen, wie beispielsweise der Entwicklung der gesamten eingebetteten Software eines Kraftfahrzeugs oder eines Flugzeugs. In einem solchen Projekt beauftragt ein Systemführer eine Reihe oft weltweit verteilter Zulieferfirmen mit der Entwicklung von Teilkomponenten auf Basis entsprechender Anforderungsdokumente (Lastenhefte). Für die Produktion dieser Teilkomponenten greifen die einzelnen Zulieferer ggf. wieder auf die Dienste anderer Unternehmen oder einzelner freiberuflicher Entwickler zurück (Out-Sourcing). Fertiggestellte Komponenten werden vom Systemführer oder einer dritten Partei zu einem Gesamtsystem integriert. Die einzelnen an einem solchen Softwareentwicklungsprojekt beteiligten Unternehmen koordinieren dabei ihre jeweiligen Entwicklungsaktivitäten innerhalb eines Intranets durch Einsatz zentraler Dienste (File-Server, Konfigurations-Management-Systeme, Dokument-Management-Software, ...), verwenden aber darüber hinaus auch meist dutzende auf verschiedene Teilaufgaben spezialisierte und nicht integrierte Systementwicklungswerkzeuge. Deshalb gibt es sowohl innerhalb einzelner Entwicklungsmannschaften als auch insbesondere an den Unternehmensgrenzen die wohlbekannten Kommunikations- und Koordinationsprobleme [ADS02]. Die Weitergabe und der Abgleich aktualisierter Versionen von Entwicklungsdokumenten ist dabei in aller Regel mit einer Vielzahl manueller und damit fehleranfälliger Tätigkeiten verbunden.

Die Lösung der skizzierten Problematik durch die Einführung *unternehmensübergreifender Entwicklungsplattformen* mit zentralen Koordinationsdiensten scheitert in aller Regel an folgenden Punkten:

1. Meist skalieren die eingesetzten Integrationslösungen nicht und lassen sich kaum für die Koordination der Aktivitäten vieler hundert, weltweit verteilter „nomadisierender“ Mitarbeiter mit sehr unterschiedlichen Anbindungen an zentrale Server einsetzen.
2. Für die Bearbeitung bestimmter Dokumente benötigte Werkzeuge stehen nur an ausgewählten Standorten zur Verfügung und können aus verschiedenen (sowohl aus technischen wie auch aus rechtlichen) Gründen nicht unbeschränkt in einem unternehmensübergreifenden Netzwerk überall genutzt werden.
3. In aller Regel sind die beteiligten Unternehmen nicht willens, (bis zu einem gewissen Zeitpunkt) vertrauliche Dokumente auf Servern außerhalb ihres Intranets zur Verfügung zu stellen, sondern es werden zu wohl definierten Zeitpunkten Sichten auf Entwicklungsdokumente an bestimmte Partner weitergegeben.

4. Darüber hinaus ist es für ein Unternehmen auch aus Verfügbarkeitsgründen meist nicht akzeptabel, für die eigenen Entwicklungsprozesse kritische Dokumente (nur) auf von dritter Seite verwalteten Servern zu speichern.

Hinzu kommt, dass in Zukunft vermehrt auch bei der kommerziellen Softwareentwicklung Open-Source-Software-Komponenten (OSS-Komponenten) eingesetzt werden, deren Entwicklung mit einem Minimum an zentralen Koordinationsdiensten und per se global verteilt mit sich kontinuierlich ändernden Entwicklungsmannschaften vonstatten geht. Zwar werden solche Projekte heutzutage in aller Regel auf einem zentralen Server verwaltet (allein Sourceforge beherbergt über 90.000 solcher Projekte), aber Beispiele wie die erfolgreichen Angriffe im November/Dezember 2003 auf die Server der OSS-Projekte Debian, Gentoo und GNU [Sch03a] zeigen sehr deutlich auch hier die Schwächen solcher zentralistischen Lösungen. Schließlich wird in der einschlägigen Literatur argumentiert, dass man sich von P2P-Lösungen eine deutliche Reduktion des Administrationsaufwandes, eine effizientere Nutzung von Bandbreiten zum Austausch von Entwicklungsdokumenten, eine deutliche Reduktion der bereitzustellenden zentralen Betriebsmittel (Server) und eine geringere Anfälligkeit gegenüber Ausfall einzelner Verbindungen und Knoten erhofft [BGM02]. Oft wird auch in der einschlägigen Literatur argumentiert, dass die bisherigen serverzentrierten Lösungen keine Unterstützung für Szenarien liefern, bei denen sich Entwicklergruppen fern ihrer eigenen Infrastruktur zur gemeinsamen Bearbeitung von Aufgabenstellungen treffen. In diesen Fällen sind Ansätze gesucht, die den direkten aber dennoch kontrollierten Austausch von Dokumenten zwischen den Laptops der versammelten Entwickler unterstützen, ohne dass dieser Austausch über weit entfernte und oft nur über unzuverlässige, schmalbandige Verbindungen erreichbare Server durchgeführt werden muss [BM02, PJG+97].

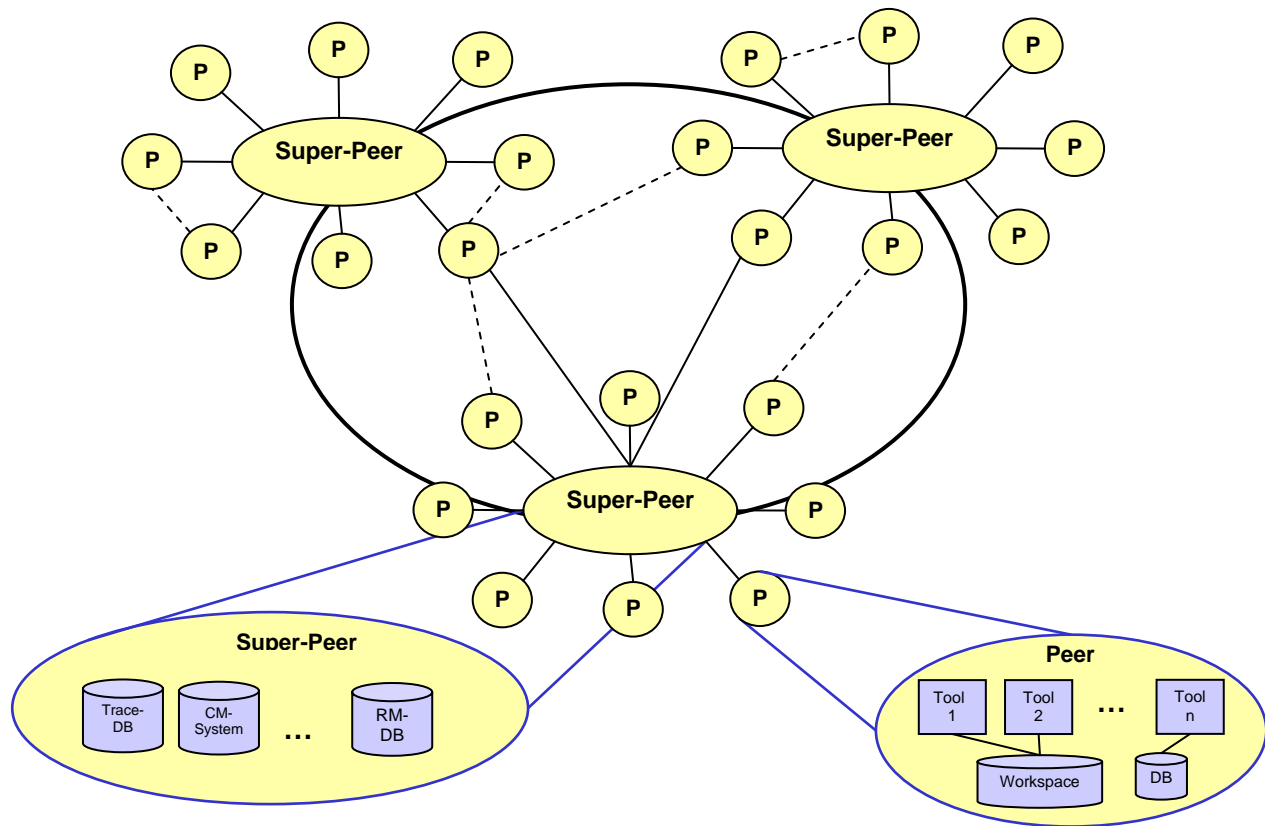


Abbildung 2: Integration von Softwareentwicklungswerkzeugen mit P2P-Techniken

Aus diesen Gründen wurden bereits Mitte bis Ende der 90er Jahre erste Vorschläge zum Einsatz von P2P-Mechanismen für die Realisierung verteilter dezentraler Konfigurationsmanagementsysteme [vdHHW96, KDJY97] diskutiert. Neuere Arbeiten aus dem akademischen Umfeld zur Entwicklung verteilter *P2P-Softwareentwicklungsumgebungen* wie [BGM02, BH04, BHSS04, Han04, BCM04] greifen diese Ideen auf und betonen dabei vor allem die Unterstützung der OSS-Entwicklung durch dynamische Entwicklungsmannschaften mit „nomadisierenden“ Entwicklern, die wie oben diskutiert oft offline und an verschiedenen Standorten arbeiten. Darüber hinaus zeigt aber die Existenz erfolgreicher kommerzieller Produkte wie Co-op [Rel05] oder Groove [Gro05], dass tatsächlich auch im kommerziellen Umfeld echter Bedarf an verteilten Softwareentwicklungsplattformen auf Basis von P2P-Mechanismen existiert. Die genannten Systeme sind dabei einerseits gute Beispiele für den erfolgreichen Einsatz von P2P-Mechanismen in diesem Umfeld, demonstrieren andererseits aber auch den Bedarf an weiteren Forschungsaktivitäten. Beispiele für interessante Fragestellungen sind etwa:

- Welche Ansätze zur Konstruktion adaptiver P2P-Overlay-Netzwerke sind in diesem Szenario sinnvoll, in dem sich auf der einen Seite die Topologie der Netze (zumindest bei kommerziellen Projekten) mit eher geringer Dynamik ändert, während man auf der anderen Seite Peers mit sehr unterschiedlichen Fähigkeiten und Rollen (File-Server als Super-Peers, Entwickler-PCs als

einfache Peers, etc.) unterscheiden muss?

- Wie kann man die zur Verfügung stehenden Ressourcen effizienter als beispielsweise in Co-op nutzen, wo alle an einem P2P-Netzwerk beteiligten Versionsverwaltungssysteminstanzen alle Versionen aller Dokumente der betrachteten Projekte spiegeln? Hier sind beispielsweise selbst adaptierende Strategien zum Replizieren und Caching von Dokumenten gefordert, wie sie in zentral administrierten, verteilten Konfigurationsverwaltungssystemen zum Einsatz kommen.
- Wie garantiert man in einem solchen Szenario, dass benötigte Entwicklungsdokumente jederzeit in der gewünschten Version verfügbar sind und durchgeführte Änderungen nicht verloren gehen, auch wenn nicht alle Peers Kopien aller Dokumente besitzen, keine pessimistischen Sperrkonzepte zur Durchführung von Änderungen verwendet werden und trotzdem ggf. automatisches Verschmelzen nebenläufig durchgeführter Änderungen nicht möglich ist?
- Wie garantiert man darüber hinaus in einem Szenario ohne zentrales Repository, dass nicht nur - insbesondere bei der Verschmelzung von Änderungen verschiedener Peers/Personen - die interne Korrektheit einzelner Dokumentversionen erhalten bleibt, sondern auch die externe Korrektheit von Versionen verschiedener Dokumente und der dazwischen existierenden Beziehungen (Traceability Links)?
- Und schließlich gilt es zu untersuchen, wie man die gerade bei der kommerziellen Softwareentwicklung sehr hohen Ansprüche an Sicherheit erfüllen kann, die in punkto Integrität replizierter Dokumente, Verbindlichkeit über Organisationsgrenzen hinweg übermittelter Änderungen, Zugriffsrechte etc. weit über das hinaus gehen, was heute verfügbare P2P-Systeme in diesem Umfeld anbieten.

3.2 Skizze eines Prototypen

In Anlehnung an die Untersuchungen im GENESIS-Projekt [ALG+04, RBS+03] und die Anforderungen an verteilte Entwicklungsumgebungen in [ECM93, ECM94] wird im Projekt QuaP2P eine auf P2P-Mechanismen basierende verteilte Softwareentwicklungsumgebung prototypisch realisiert, die in ihrer Endausbaustufe folgende Dienste anbieten wird:

1. Integration einer exemplarischen Menge einzelner Entwicklungswerkzeuge (jeweils ein Requirements-Engineering-Werkzeug, ein Modellierungswerkzeug sowie eine IDE).
2. Bereitstellung von Publish-/Subscribe- bzw. event-basierten Diensten zur Integration weiterer Entwicklungswerkzeuge und als Basis für die inkrementelle Propagation von Änderungen an Entwicklungsdokumenten.
3. Verwaltung und Überwachung der Korrektheit der von diesen Werkzeugen erstellten Dokumente und ihrer Konsistenzbeziehungen (traceability links) durch Integration der Dienste verschiedener Versionsverwaltungswerkzeuge anstelle der Realisierung eines homogenen Netzwerks von Versionsverwaltungssystemen.
4. Projektmanagement-Dienste für die Zuordnung von Organisationen und Mitarbeitern zu Projekten sowie zur Verwaltung der dazugehörigen rollen- und kontextbasierten Zugriffsrechte auf Dokumente.
5. In späteren Ausbaustufen können erweiterte Konfigurations-, Projekt- und Qualitätsmanagement-Komponenten zum Bug-Tracking, Verwaltung von Feature-Listen, etc. sowie CSCW-Dienste und Funktionen zur Durchführung regelmäßiger Analysen und Berechnung von Metriken oder zur Durchführung von Ad-hoc-Abfragen hinzugenommen werden, soweit dies für die Untersuchung von Qualitätsmerkmalen Erkenntnisgewinne verspricht.

Alles in allem stellt dieses Referenzszenario bewusst eine sehr große Herausforderung für den P2P-Ansatz dar, weil gerade für die Anforderung einer konsistenten und verlässlichen Datenverwaltung Systeme basierend auf zentralen Servern ihre natürlichen Stärken haben. Aber gerade deshalb eignet sich dieses Szenario besonders als Referenzszenario, um die Grenzen des P2P-Paradigmas aufzuzeigen. Es stellt - wie oben ausgeführt - im Unterschied zum vorherigen Referenzszenario besondere Ansprüche an die

Qualitätsmerkmale *Validität* und *Vertrauenswürdigkeit*. Alle in einem Projekt anfallenden Daten müssen innerhalb einer P2P-Softwareentwicklungsumgebung zunächst mit sehr hoher Wahrscheinlichkeit verfügbar und valide gehalten werden. Des Weiteren müssen Datenänderungen nachvollziehbar sein und rückgängig gemacht werden können. Darüber hinaus spielt der Schutz von Daten gegen unberechtigte Zugriffe und Änderungen auch auf böswilligen Knoten eine Schlüsselrolle.

Es ergeben sich jedoch auch für die Qualitätsmerkmale *Adaptivität* und *Effizienz* neue Herausforderungen, die sich weniger aus der Anzahl der Peers und der Dynamik der betrachtenden Netzwerke, sondern vielmehr aus der hohen Anzahl zu verwaltender Datenobjekte (Dokumente mit interner Struktur) mit zugehörigen Beziehungen und Integritätsbedingungen resultieren. Damit bietet dieses Szenario die Voraussetzungen für grundlegende Untersuchungen der Vor- und Nachteile von P2P-Architekturen im Vergleich zu Client-/Server-Architekturen aus Sicht der verschiedenen Qualitätsmerkmale, wie sie etwa in [BCM04] für bestimmte Teilaspekte von verteilten Konfigurationsmanagementsystemen durchgeführt wurden.

4 Zusammenfassung

In diesem Dokument sind zwei Referenzszenarien zur Evaluierung von Peer-to-Peer-Systemen motiviert und beschrieben worden. Das erste Szenario beschreibt den Einsatz von Peer-to-Peer Technologien für die Koordination von Rettungskräften bei Katastrophen. Das zweite Szenario handelt von einem Peer-to-Peer-System zur verteilten Entwicklung von Software.

Beide Referenzszenarien unterscheiden sich grundlegend in den Qualitätsanforderungen, die sie stellen. Abbildung 3 fasst die unterschiedlichen Qualitätsanforderungen schematisch zusammen. Die Abbildung orientiert sich an den in [TRQm] beschriebenen Qualitätsmerkmalen von P2P-Systemen.

Im der DFG Forschergruppe QuaP2P (www.quap2p.de) werden beide Referenzszenarien genauer untersucht und prototypisch umgesetzt.

| | | Referenzszenario | |
|----------------------|--|------------------|-----|
| | | A | B |
| Adaptivität | Skalierbarkeit | ++ | + |
| | Stabilität | ++ | ++ |
| | Flexibilität | +++ | + |
| Effizienz | Leistungsfähigkeit - Diensterbringung | +++ | + |
| | Leistungsfähigkeit – Overlay Operationen | ++ | (+) |
| | Aufwand - individueller Knoten | +++ | (+) |
| | Aufwand - gesamtes System | +++ | (+) |
| | Aufwand - IP Infrastruktur | +++ | + |
| Validität | Lokalisierbarkeit | +++ | ++ |
| | Kohärenz | ++ | +++ |
| | Konsistenz | ++ | +++ |
| | Korrektheit | + | +++ |
| Vertrauenswürdigkeit | Verlässlichkeit – Verfügbarkeit | ++ | +++ |
| | Verlässlichkeit – Zuverlässigkeit | + | +++ |
| | Verlässlichkeit – Robustheit | +++ | ++ |
| | Sicherheit – Integrität | ++ | +++ |
| | Sicherheit – Vertraulichkeit | + | +++ |
| | Sicherheit – Authentizität | + | +++ |
| | Sicherheit - Verbindlichkeit | + | ++ |

Abbildung 3: Angaben zu den Qualitätsanforderungen der Referenzszenarien

Referenzen

- [ADS02] F. Altheide, H. Dörr und A. Schürr: Requirements to a Framework for sustainable Integration of System Development Tools. In: H. Stoewer und L. Garnier (Herausgeber): Proc. of the 3rd European Systems Engineering Conference (EuSEC'02), Seiten 53-57. AFIS PC Chairs, 2002.
- [ALG+04] L. Aversano, A. D. Lucia, M. Gaeta, P. Ritrovato und M. Villani: Managing Distributed Projects in GENESIS, UPGRADE. The European Journal for the Informatics Professionals, 5(5), 2004.
- [BCM04] C. Bellettini, L. Capra und M. Monga: A comparative assessment of peer-to-peer and server-based configuration management systems. In: A. D. Lucia, H. C. Gall und S. Dustdar (Herausgeber): Proceedings of the Workshop on Cooperative Support for Distributed Software Engineering Processes (CSSE'04), Seiten 15-26, 2004.
- [BDV01] L. Bendix, A. Dattolo und F. Vitali: Software Configuration Management In Software And Hypermedia Engineering: A Survey. In: S. K. Chang (Herausgeber): Handbook of Software Engineering and Knowledge Engineering, Band 1, Seiten 523-548. World Scientific Publishing, 2001.
- [Ber90] B. Berliner: CVS II: Parallelizing Software Development. In: Proceedings of the USENIX Winter 1990 Technical Conference, Seiten 341-352. USENIX Association, 1990.
- [BGM02] D. Balzarotti, C. Ghezzi und M. Monga: Supporting configuration management for virtual workgroups in a peer-to-peer setting. In: SEKE '02: Proceedings of the 14th international conference on Software engineering and knowledge engineering, Seiten 507-511, 2002.
- [BH04] L. Bischofs und W. Hasselbring: A Hierarchical Super Peer Network for Distributed Software Development. In: A. de Lucia, H. Gall und S. Dustdar (Herausgeber): Proceedings of the Workshop on Cooperative Support for Distributed Software Engineering Processes (CSSE 2004), Seiten 99-106, 2004.
- [BHSS04] L. Bischofs, W. Hasselbring, J. Schlegelmilch und U. Steffens: A Hierarchical Super Peer Network for Distributed Artifacts. In: M. Agosti, H.-J. Schek und C. Türker (Herausgeber): DELOS Workshop: Digital Library Architectures, Seiten 105-114, 2004.

- [BM02] S. Bowen und F. Maurer: Using peer-to-peer technology to support global software development - some initial thoughts. In: Proc. of the ICSE Int. Workshop on Global Software Development, Orlando, FL, USA, May 2002, 2002.
- [Bou04] N. Boulila: Computer Supported Cooperative Software Engineering: A framework for supporting distributed concurrent group modeling of software. In: Proceedings of the International Conference on Applied Computing, 2004.
- [CSN98] F. L. Chan, M. D. Spiller und A. R. Newton: WELD - An Environment for Web-based Electronic Design. In: Design Automation Conference, Seiten 146-151, 1998.
- [CW98] R. Conradi und B. Westfechtel: Version Models for Software Configuration Management. ACM Computing Surveys, 30(2):232-282, 1998.
- [DS04] H. Dörr und A. Schürr (Herausgeber): Special Section on Tool-integration Applications and Frameworks, in: International Journal on Software Tools for Technology Transfer, Seiten 183-255. Springer Verlag, 2004.
- [DS05] H. Dörr und A. Schürr (Herausgeber): Special Section on Model-based Tool Integration, in: Journal of Software&Systems Modeling, Seiten 109-170. Springer Verlag, 2005.
- [ECM93] ECMA: TR/55 - Reference Model for Frameworks of Software Engineering Environments. Technischer Bericht, ECMA, 1993. <http://www.ecmainternational.org/publications/files/ecma-tr/TR-055.pdf>.
- [ECM94] ECMA: TR/69 - Reference Model for Project Support Environments. Technischer Bericht, ECMA, 1994. <http://www.ecmainternational.org/publications/files/ecma-tr/TR-069.pdf>.
- [Gro05] Groove Networks: Groove Virtual Office, 2005. <http://www.groove.net/home/index.cfm>.
- [Han04] K. Hansen: Thoth - A publish/subscribe architecture for peer-to-peer tool integration. Special Section on Tool-integration Applications and Frameworks in International Journal on Software Tools for Technology Transfer, 6(3):219-230, 2004.

- [HIST93] A. Halevey, Z. Ives, D. Suciu und I. Tatarinov: Schema Mediation in Peer Data Management Systems. In: Proceedings of ICDE Conference 2003, Seiten 505-516, 1993.
- [IBM] IBM: Rational ClearCase. <http://www-306.ibm.com/software/awdtools/clearcase>.
- [WiMax] IEEE Microwave Theory and Techniques Society IEEE Computer Society. 802.16: Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems. IEEE Standard, 1 October 2004.
- [Ind] L. S. Indrusiak: A Review on the Framework Technology Supporting Collaborative Design of Integrated Systems. <http://www.citeseer.ist.psu.edu/indrusiak02review.html>.
- [KDJY97] G. E. Kaiser, S. E. Dossick, W. Jiang und J. J. Yang: An Architecture for WWW-based Hypercode Environments. In: International Conference on Software Engineering, Seiten 3-13, 1997.
- [KSRS04] N. Kurata, B. Spencer und M. Ruiz-Sandoval: Application of Wireless Sensor Network Mote for Building Risk Monitoring. In: Proceedings of INSS 2004, Poster Supplement, Poster No. 8, 2004.
- [Lyo00] D. Lyon: Practical CM - Best Configuration Management Practices. Butterworth-Heinemann, 2000.
- [Nat02] National Commission On Terrorist Attacks Upon The United States: Emergency Preparedness and Response: Statement No13, 2002. <http://www.mipt.org/pdf/NCTAUTUSStaff-Statement13.pdf>.
- [Onl04] G. Online: Review of Firefighter F.D. 18, 2004. <http://www.gamespot.com/ps2/action/firefighterfd18/review.html>.
- [PJG+97] T. W. Page, Jr., R. G. Guy, J. S. Heidemann, D. H. Ratner, P. L. Reiher, A. Goel, G. H. Kuenning und G. Popek: Perspectives on Optimistically Replicated Peer-to-Peer Filing. Software - Practice and Experience, 11(1),1997.

- [RBS+03] Ritrovato, Ballarini, Santucci et al.: Cooperative Software Development in GENESIS: Requirements, Conceptual Model and Architecture. Cooperative Methods and Tools for Distributed Software Processes, Seiten 67-86, 2003.
- [Rel05] Reliable Software: Code Co-Op, 2005. http://www.relisoft.com/co_op/index.htm.
- [Sla03] G. Slack: Smart Helmets Could Bring Firefighters Back Alive. Forefront, 2003.
- [Tel05] Telelogic: DOORS, 2005. <http://www.telelogic.com/products/doorsers/doors>.
- [Tig05] Tigris.org: Subversion, 2005. <http://subversion.tigris.org>.
- [TRQm] O. Heckmann, R. Steinmetz, N. Liebau, A. Buchmann, C. Eckert, J. Kangasharju, M. Mühlhäuser, A. Schürr: Qualitätsmerkmale von Peer-to-Peer Systemen. Technischer Bericht, 2006.
- [vdHHW96] A. van der Hoek, D. Heimbigner und A. L. Wolf: A generic, peer-to-peer repository for distributed configuration management. In: ICSE '96: Proceedings of the 18th international conference on Software engineering, Seiten 308-317. IEEE Computer Society, 1996, ISBN 0-8186-7246-3.
- [WMC01] B. Westfechtel, B. P. Munch und R. Conradi: A Layered Architecture for Uniform Version Management. Software Engineering, 27(12):1111-1133, 2001.