

# Infrastrukturen zur Aneignungsunterstützung

## Ein Konzept zur Integration von produkt- und prozessorientierter Flexibilisierung

Gunnar Stevens<sup>1</sup>; Volker Wulf<sup>1,2</sup>; Volkmar Pipek<sup>2</sup>

<sup>1</sup>Fraunhofer-Institut für Angewandte  
Informationstechnik FIT  
53754 Sankt Augustin  
{gunnar.stevens@fit.fraunhofer.de}

<sup>2</sup>Wirtschaftsinformatik  
Universität Siegen  
57068 Siegen  
{volker.wulf, volkmar.pipek}@uni-siegen.de

### Abstract

Die Aneignung und Einbettung von Computersystemen in den Nutzungskontext stellt einen zentralen, nichtsdestotrotz von der bisherigen Forschung ungenügend adressierten Aspekt der Softwarenutzung dar. Es muss dabei klar sein, dass Aneignung ein kollektiver und kreativer Prozess gesehen werden muss. In diesem Beitrag stellen wir das Konzept der Aneignungsinfrastruktur vor. Dabei handelt es sich um eine sozio-technische Methode, die auf die Kombination hoch anpassbaren Softwaredesigns mit der Nutzung einer Kommunikations- und Kooperations-Infrastruktur innerhalb der Nutzercommunity sowie zwischen Benutzern und Entwicklern aufbaut. Die Kommunikations- und Kooperationsinfrastruktur ermöglicht es, nicht antizipierte Anforderungen des Nutzungskontexts zu fassen und in einen agilen Entwicklungsprozess einfließen zu lassen. Die dadurch möglich gemachte technische Flexibilität unterstützt den Umgang mit differenzierten und dynamisch wechselnden Anforderungen. Anhand des BSCWeasel-Projektes zeigen wir, wie eine entsprechende Aneignungsinfrastruktur realisiert werden kann.

## 1 Einführung

Die Forschung, die durch das *Produktivitätsparadox* angeregt wurde, hat das Verständnis für die wirtschaftlichen Effekte von Informationstechnologien enorm verbessert. So zeigen

Brynjolfsson und Hitt, dass die meisten Produktivitätseffekte nicht direkt auf Computer-Anwendungen zurückzuführen sind, sondern auf die spezifische Aneignung der Artefakte in der organisationalen Umwelt [Bryn98]. Aneignung von Technologien wird dabei als kreativer Prozess des Anwenders – sowohl Organisation, als auch Endbenutzers - verstanden, der auch über die zugehörigen intendierten Nutzungsmöglichkeiten der Softwareentwickler hinausgehen kann [Pipe05]. Der erfolgreiche Prozess der Technologie-Aneignung führt dabei zu neuen Arbeitsweisen, innovativen Geschäftsprozessen oder veränderten organisatorischen Strukturen. Da die organisationale Umgebung sich typischerweise während des Aneignungsprozesses ändert und eine erfolgreiche IT-Aneignung zusätzliche Auswirkung auf die Arbeitspraxis, Geschäftsprozess oder organisatorische Strukturen hat, sind die IT-Anforderungen im Prozess nicht stabil [Orli96, Pip+99]. Deshalb muss sich ein Ansatz, der die IT-Aneignung unterstützt, mit den Dynamiken des Anwendungskontexts auseinandersetzen. Daneben muss die Ausdifferenzierung von Anforderungen aus verschiedenen Anwendungsfeldern berücksichtigt werden. Darüber hinaus sollte das Potential der Benutzer berücksichtigt werden, die aufgrund ihrer Praxiskenntnis und ihrer Nutzungserfahrung im kreativen Handeln innovative Lösungen entwickeln [vHip88].

Im Folgenden zeigen wir einen Ansatz, der einen Brückenschlag zwischen entwicklungsorientierter und nutzungsorientierter Flexibilisierung schlägt. Dabei werden wir zunächst den aktuellen Stand in der Diskussion innerhalb von HCI und SE aufarbeiten. Dann werden wir ein allgemeines Framework für die Aneignungsunterstützung vorstellen und anschließend anhand des Groupwaresystems BSCWeasel zeigen, wie sich das darlegte Konzept in die Praxis umsetzen lässt.

## **2 Aneignung und Flexibilisierung**

Im Folgenden wird der Zusammenhang zwischen Aneignungsprozessen und den technischen Konzepten zur Flexibilisierung darlegt. Das Studium der Literatur zeigt, dass das Thema der Flexibilisierung dabei zumeist unter zwei unabhängig voneinander betrachteten Perspektiven – der produktorientierten und der prozessorientierten Flexibilisierung - angegangen wird. Doch aufgrund der unter der jeweiligen Perspektive erzielten Fortschritte scheint es nun angebracht, die beiden Perspektiven wieder zu synthetisieren.

## 2.1 Aneignungsprozesse

Aneignungsprozesse bei Groupware wurden in verschiedenen Langzeitstudien untersucht [Orli96, Pip+99, Kar+98, Wulf99, Tör+03]. Diese Fallstudien geben ein gutes empirisches Bild der Aneignungsprozesse und den daraus resultierenden Änderungen in der Arbeitspraxis, Geschäftsprozessen und organisatorischen Strukturen.

Orlikowski und Hofmann bieten ein konzeptionelles Modell an, um die organisatorischen Änderungen, die aus der Aneignung von Groupware resultieren, zu verstehen [Orl+97]. Sie unterscheiden drei verschiedene Typen von im Prozess auftretenden Änderungen: *Anticipated changes* sind organisatorische Transformationen, die geplant sind und während der Groupware-Nutzung in der Organisation auch umgesetzt werden. Im Gegensatz dazu stehen die *opportunity-based changes*, die bei der Einführung nicht vorhergesehen waren, sondern auf der Entdeckung von Potentialen des Systems im Verlauf der Aneignung und darauf aufbauender zielgerichteter Weiterentwicklung beruhen. Demgegenüber sind *emergent changes* nicht nur in der Einführungsphase nicht vorhersehbar, sie finden insbesondere unter der Hand in dezentralen und ungeplanten Aktivitäten statt.

Die gelegheitsbasierten, und emergenten Änderungen bedürfen einer Aneignungsinfrastruktur, um sich als Potential lokal sich entwickelnden Innovation entfalten zu können. Diese Formen nicht vorhergesehener Aneignung geschehen immer dann, wenn ein Einzelner oder eine Gruppe von Benutzern eine technische Funktionalität auf innovative Weise in Praxis integrieren kann. Oft löst dabei eine externe Irritierung die Reflektion über die Anwendung und die herrschende Praxis aus. Typischerweise lernen dabei die Benutzer voneinander, wobei sich benutzer-getriebene Innovationen über lokale Wissensnetzwerke verbreitern.

Mit dem Aufkommen vernetzter, kooperativer Applikationen ist der Bedarf an flexiblen, gruppenanpassbaren Anwendungen gestiegen [Wul+95, Ben+95]. Zugleich stellt die Verteiltheit der Akteure stellt auch neue Anforderung an die Gestaltung der Anpassbarkeit der Systeme [Ober94]. Einige dieser Herausforderungen wurden durch Pipek mit seinem Konzept der Diskursinfrastrukturen angegangen [Pipe05]. Ein integrierter Ansatz zur Aneignungsunterstützung steht jedoch noch aus.

## 2.2 Produkt-orientierte Flexibilisierung

Die HCI-Forschung betrachtet Flexibilität hauptsächlich als ein Produktmerkmal, das es erlaubt, Anwendungen innerhalb ihres Nutzungskontextes anzupassen [Hen+91, Teeg00]. Das Anpassen findet nach der eigentlichen Design- und Implementierungsphase statt. Typischerweise beginnt es direkt nach der Installation oder während der Installation einer Anwendung. Das Anpassen erfolgt normalerweise als Zusammenarbeit von normalen Benutzern, lokalen Experten, System Support oder Helpdesk-Personal, wobei man drei Ebenen der Komplexität -Parametrisierung, Erweiterung und Neuprogrammierung - unterscheidet [Mørre97].

Die Forschung zu produkt-orientierter Flexibilisierung fokussiert dabei vor allem auf den Entwurf von geeigneten Softwarearchitekturen, der Gestaltung von Anpassungswerkzeugen und der kognitiven Anforderungen an den Endbenutzer. Verschiedene hoch anpassbare Systeme - kommerzielle wie Forschungsprototypen - wurden hierzu entwickelt, wie z.B. OVAL [Mal+95], oder FreEvolve [Stiem00].

Die Frage, wie Flexibilitätsanforderungen in geeigneter Weise erhoben werden, wird jedoch selten bis gar nicht adressiert. Dieser blinde Fleck liegt unter anderem daran, dass die Ansätze zur Gestaltung hoch anpassbarer Systeme Fragen der Flexibilität von Produkten nicht mit Fragen der Flexibilisierung von Entwicklungsprozessen verbinden.

### **2.3 Prozess-orientierte Flexibilisierung**

Selbst die Gestaltung hoch anpassbarer Systeme kann nicht sämtliche Nutzungssituation und die daraus abgeleiteten Flexibilitäts-Anforderungen vorhersehen. Deshalb kann es immer wieder vorkommen, dass die bereitgestellten Anpassungsmöglichkeiten der Systeme nicht ausreichen. Deshalb sollte auch der Entwicklungsprozess auf nicht antizipierte Anforderungen vorbereitet sein. Insbesondere bedarf es hierfür einer Infrastruktur, die es erlaubt, solche nicht antizipierten Situationen im Nutzungskontext überhaupt erst einmal sichtbar zu machen.

Agile Vorgehensmodelle, wie eXtreme Programming (XP) [Beck00], gehen davon aus, dass Anforderungen sich permanent ändern können und haben deshalb verschiedene, aufeinander abgestimmte Techniken entwickelt, die es erlauben, den Änderungswünschen mit vertretbaren Kosten nachzukommen [Beck00].

Der Ansatz des Radical Tailoring [Mal+95] versucht, sämtliche zukünftige Anforderungen mittels hoch anpassbarer Produkte zu erschlagen. Demgegenüber verfolgt XP die Strategie, hierauf mit hoch agilen Entwicklungsprozessen zu reagieren. Damit stellt XP sicherlich die bekannteste und extremste Form einer prozess-orientierten Flexibilisierungsstrategie dar.

Da sich aber XP als reines Ausführungsorgan der Kundenanforderung versteht, kümmert sich XP nicht selbst um Aneignungsprozesse im Nutzungskontext. Insbesondere findet man keine Überlegungen zu einer geteilten Infrastruktur, die einen gemeinsamen Lernprozess von Benutzern und Entwicklern unterstützt. Stattdessen erhalten Entwickler nur indirekt Feedback über das *Customer on Side*-Prinzip [Beck00]. Erschwerend kommt hinzu, dass dieses Prinzip in der Praxis schwer zu realisieren ist [Rum+01].

Von partizipativ orientierten evolutionären Prozessmodellen wie STEPS [Flo+89] unterscheidet sich XP insbesondere im Verständnis der Beziehung von Entwicklern und Nutzern. Die Kernidee von STEPS beruht diesbezüglich darin, Softwareentwicklung als ein fortdauernden gegenseitigen Lernprozess aufzufassen, bei dem sich Software in enger Zusammenarbeit von Benutzern und Entwicklern in diversen Releases entwickelt. Die kontinuierliche Verbesserung sollte dabei auf Grund von konkreten, alltagsweltlichen Nutzungserfahrungen erfolgen. STEPS ist jedoch ein sehr allgemeines Modell. Die Realisierung der unterschiedlichen Teile des Modells wurde innerhalb von STEPS auch nicht weiter ausgearbeitet.

Insbesondere auf der konzeptionellen Ebene unberücksichtigt blieb in STEPS jedoch der Aspekt heterogener Umwelten, an die das Produkt im Nutzungskontext jeweils individuell angepasst werden muss. Eine Erweiterung in diese Richtung findet man aber im erweiterten STEPS Modell von Wulf und Rohde [Wul+95]. Auf der softwaretechnischen Ebene fanden Ideen des erweiterten STEPS-Modells Eingang in den Forschungsprototypen FreEvolve [Stie00]. Jedoch ist auch bei FreEvolve keine Aneignungsinfrastruktur integriert, die es erlaubt, nicht antizipierte Aneignungsphänomene und Anpassungsbedarfe zu erfassen.

Andere relevante Arbeiten sind Fischers konzeptionellen Überlegungen zur der Erweiterung seines Konzepts der End-User Modifiability [Fis+90]. Ähnlich dem erweiterten STEPS Modell schlägt er in den Arbeiten zum Prozessmodell SER vor, dass die produktorientierte Flexibilisierung Teil eines evolutionären Entwicklungsprozesses sein sollte [Fis+02]. Er geht dabei von einem Phasenmodell aus, bei dem sich Nutzung und kooperative Gestaltung abwechseln. Jedoch finden sich in seinen Arbeiten nur vereinzelt Hinweise darauf, wie eine solche Zusammenführung beider Flexibilisierungsstrategien aussehen könnte.

Zusammenfassend lässt sich feststellen, dass in den neueren Arbeiten ein Trend zu erkennen ist, Ansätze zur produkt-orientierter Flexibilisierung mit prozess-orientierter Flexibilisierung zu kombinieren. Jedoch ist in den Arbeiten kein klarer Ansatz herausgearbeitet, der die Lücke zwischen den beiden Perspektiven überbrückt.

### 3 Vermittlung zwischen prozess- und produkt-orientierter Flexibilisierung: Das Konzept einer Infrastruktur zur Aneignungsunterstützung

Im Folgenden wollen wir unser Konzept einer technischen Infrastruktur zur Unterstützung von Aneignungsprozessen in Organisationen umreißen. Der Begriff Infrastruktur adressiert dabei genau jenes offene Problem bisheriger Ansätze, produkt-orientierte mit prozess-orientierter Flexibilisierung zu verbinden.

Das Konzept verfolgt dabei zwei grundsätzliche Ziele:

- die Förderung des Wissensaustausch innerhalb der Nutzercommunity
- die Förderung einer reziproken Partizipation am Nutzungs- bzw. Entwicklungskontext

Durch die Förderung einer reziproken Partizipation soll dabei die Chance erhöht werden, innovative Nutzungen als auch eventuelle Nutzkrisen zu erkennen und aufgrund geeigneter Produktmodellierung (wie z.B. durch komponenten-basiertes Software Engineering) und geeigneter Prozessmodellierung (wie z.B. eXtreme Programming) auf eine effiziente Art und Weise reagieren und die Anwendung an den lokalen Kontext anpassen bzw. erweitern zu können.

Bei der Gestaltung der Infrastruktur gilt es zu beachten, dass sowohl die Nutzercommunity, als auch Benutzer und Entwickler üblicherweise räumlich sowie zeitlich verteilt sind, da Aneignung ein kontinuierlicher, nicht auf einen bestimmten Zeitraum beschränkter Prozess ist. Zum anderen gilt es zu berücksichtigen, dass hier zwischen Akteuren mit deutlich unterschiedlichen Arbeitspraxen, Interessen, Wissenshorizonten und Kulturen vermittelt werden soll.

Bei der Vermittlung zwischen heterogenen Kulturen, haben Star und Griesemer anhand einer Fallstudie gezeigt, dass sogenannten *Boundary Objects* eine besondere Rolle zur Lösung des Vermittlungsproblem zukommt [Sta+89]. *Boundary Objects* zeichnen sich dabei dadurch aus, dass sie auf der einen Seite eine gemeinsam geteilte Entität in den verschiedenen Kulturen darstellen, dass sie aber auf der anderen Seite von den jeweiligen Akteuren anders betrachtet und genutzt werden können. Strübing erläutert anhand eines Betatests des Programms ATLAS/ti<sup>1</sup>, wie das Artefakt selbst die Rolle eines *Boundary Object* innerhalb des Gestaltungs- und Nutzungsdiskurs einnehmen kann [Strü99].

Im Falle der Aneignungsunterstützung von Softwareartefakten gilt es unseres Erachtens darum, dass Konzept von *Boundary Objects* mit dem softwareergonomischen Konzept der *Direct*

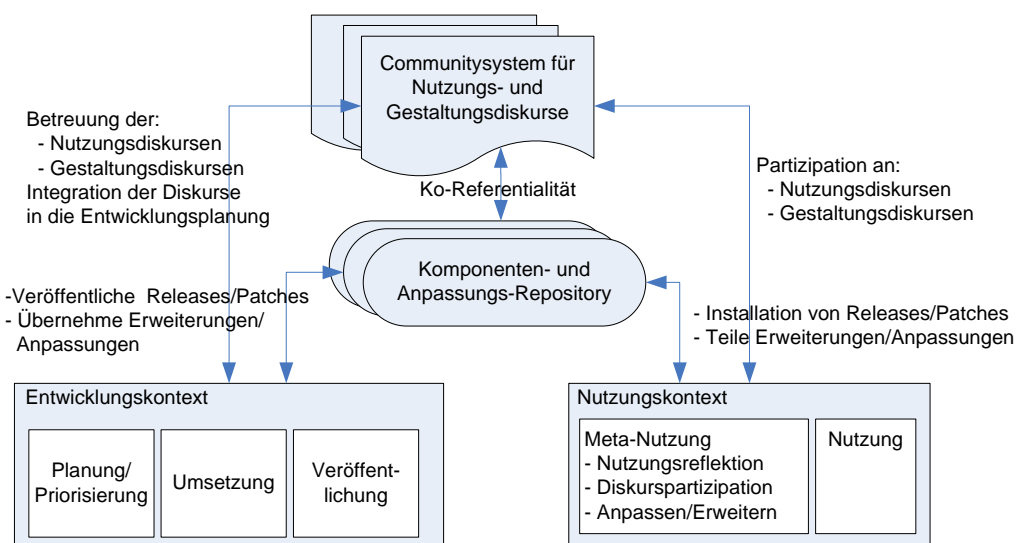
---

<sup>1</sup> Vgl. <http://www.atlasti.de/>

*Manipulation* zu verbinden. Das von Ben Shneidermann eingeführte Konzept besagt im Kern, dass die für den Nutzungskontext relevanten Dinge, die so genannte *Objects of Interest*, am User Interface visuell präsentiert werden und dort auch direkt bearbeitbar sein sollten [Shne83]. Umgekehrt zeigte sich in unserer Evaluation, dass die symbolische Interpretation des Artefakts durch das an der Benutzerschnittstelle sichtbare Verhalten geprägt wird und die visuelle Repräsentation in der Kommunikation zwischen den Akteuren eine besondere Bedeutung zukommt.

Für die technische Umsetzung der Aneignungsunterstützung lassen sich aus der Verbindung beider Konzepte folgende Anforderungen ableiten:

- sollten die Nutzungs- und Gestaltungsdiskurse am User Interface sichtbar gemacht werden und direkt aus der Nutzung zugänglich sein [Wul+01]
- sollte bei der Rolle des Artefakts als *Boundary Object* insbesondere seine visuelle Repräsentation und den Aspekt des *Direct Manipulation* mit einbeziehen
- sollte deshalb das direkte Referenzieren auf die *Objects of Interest* innerhalb der Nutzungs- und Gestaltungsdiskurse unterstützt werden



**Abbildung 1:** Schaubild einer geteilten, technischen Infrastruktur, die zwischen Nutzungs- und Entwicklungskontext vermittelt.

Abbildung 1 zeigt das Schaubild einer von Benutzern und Entwicklern geteilten Infrastruktur, die die verschiedenen Aspekte der Aneignungsunterstützung integriert und den Akteuren, eine an ihre Bedürfnisse angepasste Sicht auf die Meta-Nutzung erlaubt. Dabei sollten die Komponenten samt deren Anpassungen und die darauf sich beziehenden Diskurse einander ko-

referenziell verweisen [Wul+01], umso den Übergang zwischen Nutzungskritik und Nutzungsanpassung zu erleichtern.

Weite Teile dieser Architektur konnten im BSCWeasel-Projekt auf Basis von Eclipse umgesetzt werden. Die Realisierung des *Component Repository* wurde dabei von Eclipse mitgeliefert. Die *Discourse Infrastructure* wurde durch die im Folgenden beschriebenen Module *CHiC* und *PaDU* realisiert. Der Übergang zwischen Nutzungskritik und Nutzungsanpassung ist jedoch noch eine Herausforderung für zukünftige Arbeiten.

#### **4 Umsetzung einer Aneignungsunterstützung innerhalb des BSCWeasel Projekts**

Um unser Konzept zu erforschen, haben wir ein Groupware-System namens BSCWeasel entwickelt, das die hier vorgeschlagene Infrastruktur zur Aneignungsunterstützung als integralen Bestandteil besitzt. BSCWeasel ist ein Rich Groupware Client für das BSCW System, das an der Gesellschaft für Mathematik und Datenverarbeitung (GMD – heute integriert in die Fraunhofer-Gesellschaft) Mitte der 1990er Jahre entwickelt wurde [Ben+95].

BSCW ist eine erfolgreiche, vollständig browserbasierte Groupware-Anwendung. Die zugrunde liegende Architektur hat bestimmte Vorteile für die Benutzer, so z.B. bedarf sie keines zusätzlichen Installationsaufwands. Jedoch hat die *Thin Client*-Lösung auch verschiedene Nachteile gegenüber einer modular aufgebauten *Rich Client*-Lösung, insbesondere was die eigenständige Erweiterbarkeit durch den Nutzer anbetrifft [Ste+04].

Das BSCWeasel – ein Open Source Projekt<sup>2</sup>, das im Frühjahr 2004 begonnen wurde –ergänzt deshalb das BSCW um solch einen modular aufgebauten *Rich Client* auf Eclipse-Basis (siehe Abbildung 2). In der ersten Version des BSCWeasel wurden dabei einige Hauptfunktionen des BSCW umgesetzt, die über die XML RPC API des BSCW Systems angesprochen wurden. Die erste Realisierung konnte dabei fast vollständig durch die von der *Eclipse Rich Client*-Plattform (RCP) bereitgestellten Komponenten und einige weitere schon vorhandene Komponenten realisiert werden. Später wurde der *Rich Client* durch das Hinzufügen und das Adaptieren bestehender Komponenten um neue Funktionalität erweitert (z.B. ein Chat auf XMPP/Jabber Basis).

---

<sup>2</sup> Vgl. [www.bscweasel.de](http://www.bscweasel.de)



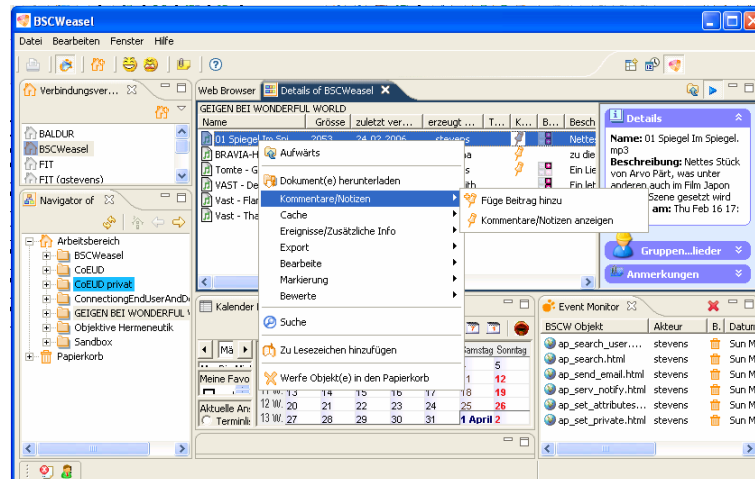


Abbildung 2: Snapshot des BSCWeasel Client

In das BSCWeasel wurden zusätzlich zwei (aktuell noch getrennte) Teilsystemen – ChiC und PaDU - integriert, die den Aneignungs- und Gestaltungsdiskurs zwischen den Benutzern bzw. Benutzern und Entwicklern verbessern sollen.

#### 4.1 ChiC - Community Help In Context

Das eine Teilsystem ist die *Community Help In Context* (ChiC) [Ste+06a]. Die Kernidee von ChiC besteht dabei darin, das Konzept kontext-sensitiver Hilfe mit dem Wiki-Konzept zu verbinden. Das integrierte Hilfesystem wird so von einem unidirektionalen Medium zwischen Hersteller und Benutzer zu einem Community-orientierten Medium. Er erlaubt den Benutzern zu jeglicher Funktion des Anwendungssystems einen Beitrag zu erzeugen, zu ändern, zu kommentieren oder zu Bewerten und mit anderen Benutzern auszutauschen. Das entwickelte System macht sich dabei zu Nutze, dass heutige Anwendungssysteme nach dem *Direct Manipulation* Prinzip [Schne83] aufgebaut sind. Durch die algorithmische Inspektion der Widgetstruktur der laufenden Anwendung erlaubt das ChiC-System, die Nutzungsdiskurse an die betreffenden *Objects of Interest*, den Widgets, zu heften. Das hierbei angewendete Verfahren, das auf Prinzipien der Metaprogrammierung basiert, hat darüber hinaus den Vorteil, dass es auch in beliebigen Anwendungssystemen auf Eclipse-RCP-Basis eingesetzt werden kann. Die technischen Details von ChiC sind in [Ste+06a] genauer dargelegt.

#### 4.2 PaDU – Participatory Design in Use

Um die reziproke Partizipation zwischen Benutzern und Entwicklern zu fördern, haben wir ein professionelles *Issue-Tracking-System* in die BSCWeasel-Applikation integriert, in dem

Benutzer und Entwickler in gleicher Weise partizipieren können [Ste+06b]. Um den Einstieg für den Endbenutzer zu vereinfachen, haben wir das System dabei mit einer speziell für Endbenutzer angepassten Benutzerschnittstelle ausgestattet. Dies erlaubt Benutzern des BSCWeasel-Systems nun einen speziellen Zugang aus ihrem unmittelbaren Nutzungskontext heraus.

Die Nutzerbeteiligung wird durch ein Formular zur Beschreibung des Nutzungsproblems unterstützt: Das Formular ist dabei eine Abwandlung des ursprünglichen *Critical Incident-Dialog* (Castillo), so wie er in der Studie von Hartson, Castillo et al. benutzt wurde [Har+96]. Im Gegensatz zu Hartson, Castillo et al. erlaubt PaDU auch das graphische Annotieren des Nutzungskontexts. Hierzu wird vom aktuellen Zustand automatisch ein Screenshot angelegt, in den Dialog eingefügt und mit einem speziellen Annotationswerkzeug verknüpft. Dies erlaubt dem Benutzer, nicht nur textuell, sondern auch graphisch auf die *Objects of Interest* zu verweisen.

Da Systementwicklung ein im hohen Maße kommunikativer Aushandlungsprozess ist, war es bei der Gestaltung von PaDU wichtig, die Kommunikationsprozesse für alle Beteiligten transparent zu machen. Deshalb wird z.B. der Benutzer nach dem Senden eines Feedback-Reports direkt an die Stelle in *Issue-Tracking-System* geführt, wo sein Beitrag samt aller Ergänzungen Dritter verwaltet wird. Dieses Feature erlaubt es dem Benutzer, den Fortgang seines Beitrags weiterzuverfolgen, die Kommentare von anderen Benutzern/Entwicklern zu lesen bzw. selber neue Kommentare zu verfassen.

Die technischen Details und die Einordnung von PaDU in die Landschaft anderer Systeme zur Beteiligung von Anwendern während der Nutzung ist in [Ste+06b] dargelegt.

## **5 Diskussion der Aneignungsunterstützung im BSCWeasel Projekt**

Das BSCWeasel wurde an der Universität Siegen entwickelt. Die Kerngruppe bestand aus zwei Entwicklern, die zu verschiedenen Zeitpunkten durch Studententeams unterstützt wurden. Eine erste anfängliche Version des BSCWeasel wurde ab Mai 2005 von den Entwicklern und den Studententeams benutzt. Spätere Versionen wurden innerhalb der Forschungsgruppe (ca. 15 Mitglieder) und einer befreundeten Forschungsgruppe (ca. 15 Forscher) ca. 100 km von Siegen entfernt veröffentlicht.

Das BSCWeasel-Projekt folgt einen Aktionsforschungsansatz. Hieraus ergibt sich, dass primär formative Evaluationen betrieben werden, um die Ergebnisse in die Verbesserung der Gestaltung einfließen zu lassen. Sie beruht auf der Untersuchung von Paper-Mockups, semi-strukturierten und informellen Interviews, gesammelten Feldnotizen und BSCWeasel bezogener Email-Korrespondenz. Zusätzlich wurden zwei Usabilitystudien nach dem DaTech-Verfahren durchgeführt. Die erste Studie wurde ohne die Aneignungsinfrastruktur im April 2005 mit 9 Nutzern durchgeführt. Die zweite wurde im Januar 2006 mit der Aneignungsinfrastruktur mit 6 Nutzern durchgeführt.

Hinsichtlich des Nutzungsgrads des BSCWeasel an der Universität Siegen und dem Forschungsinstitut haben wir 11 regelmäßige Nutzer gezählt. Alle diese Nutzer waren zuvor auch intensive BSCW Nutzer und haben im BSCWeasel spezielle Funktionen entdeckt, die ihnen die alltägliche Arbeit vereinfachen.

Von September 2005 bis 2006 wurden 74 Gestaltungsvorschläge mittels der Aneignungsinfrastruktur gemeldet. Die meisten der dort adressierten Problem verweisen auf Nutzungsprobleme, die zwar die Nutzer irgendwie gemeistert haben, bei denen jedoch eine Erweiterung bzw. ein Redesign der Groupware das Nutzerverhalten ändern oder die Effizienz steigern könnte.

Basierend auf unseren Studien innerhalb des BSCWeasel-Projekts konnten wir drei Dimensionen identifizieren, die bei der Gestaltung einer Infrastruktur zur Aneignungsunterstützung berücksichtigt werden sollte. Wir wollen die einzelnen Dimensionen im Einzelnen darlegen. Die Ausarbeitung konkreter Gestaltungsrichtlinien ist jedoch noch eine zu leistende Aufgabe.

## **5.1 Die epistemologische Dimension: Krisensituationen und Erforschungsprozesse**

Die heutige Forschung betrachtet die verschiedenen Aspekte von Krisensituation<sup>3</sup> in der Nutzung und damit einhergehende Reflektionsprozesse meist noch getrennt. So wird die Konsultierung des Hilfesystems, das Anpassung des Systems oder die Partizipation am Gestaltungsdiskurs in Open-Source-Projekten getrennt voneinander studiert. In unseren Interviews ist uns jedoch aufgefallen, dass sich diese Trennung aus Nutzersicht nicht so scharf darstellt. Vielmehr fanden es die Nutzer in einer Krisensituation häufig schwierig, die richtige Unterstützungsfunktionalität auszuwählen.

---

<sup>3</sup> Im Folgenden soll hier der englische Begriff der *breakdown situation* mit Krisensituation übersetzt werden.

Einen Hinweis auf das zugrunde liegende Problem fanden wir in einem Kommentar eines Nutzers. Er kommentierte sein Erweiterungsvorschlag mit den Hinweis: „*Funktion nicht vorhanden oder nicht gefunden, d.h. ich bin ein DAU ;-)*“. Im ersten Abschnitt des Satzes wird das fehlende Feature als ein Mangel des Systems konstatiert, was einen Kontakt zu den Entwicklern legitim erscheinen lässt. Im zweiten Abschnitt wird das Problem jedoch auf mangelndes eigenes Wissen über das System zurückgeführt, was die Konsultation eines Hilfesystems legitimieren würde. Der springende Punkt hier ist aber, dass aus der Perspektive des Nutzers in der Krisensituation nicht entschieden werden kann, welcher Fall hier tatsächlich vorliegt.

Um die Struktur von Krisensituation und deren Umgang durch den Nutzer besser verstehen zu können, haben wir deren Manifestationen in den ‚Thinking-Aloud‘-Protokollen, die während der Usability Studien angefertigt wurden, eingehender studiert. Dabei haben wir auf die von Ullrich Oevermann entwickelte Methode der Sequenzanalyse zurückgegriffen [Oeve97]. Aus Platzgründen kann an dieser Stelle die Ergebnisse nur komprimiert anhand einer Passage illustriert werden.

„*Na, komm!* [das routinisierte Handeln gerät in eine Krise] - *Ja, jetzt wäre die Frage,* [der Nutzer fängt an über die aktuelle Situation zu reflektieren] *jetzt bin ich hier im Navigator und wenn ich den Inhalt jetzt da drüben sehen wollte, ne* [im Sinne von: „nicht wahr?“], *ganz einfach, also dann, vielleicht ... weiß ich nicht, vielleicht kann man das da reinschieben.* [er entwickelt eine Hypothese über die Problemsituation] *Da geht irgendwas nicht. (T: zustimmend Hmm)* [experimentiert herum]. *Nee, so! O.k., prima, kriege ich jetzt nicht hin. Ich hatte jetzt nur gedacht, vielleicht könnte man - sozusagen wie im Windows Explorer ...* [er kommt zu den Schluss, dass seine Annahme nicht stimmt und fängt an sie durch eine neue zu ersetzen] [er probiert verschiedene Hypothesen aus] *so wie ich mir das vorstelle, scheint das nicht zu gehen* [kommt zu einer Überzeugung] *aber das ist egal, macht nix.*[beschließt die Sache nicht weiter zu erforschen]“ (Passage aus einem Thinking-Aloud Protokoll).

Wir können unsere Interpretation einer Krisensituation in der Softwarenutzung wie folgt zusammenfassen: Das Hauptmerkmal einer Krisensituation ist der Zweifel, der in aktuellen Situation gründet<sup>4</sup>. In seiner allgemeinen Form kann er durch den höchst indexikalischen Ausdruck charakterisiert werden: „*Hier und jetzt ist irgendwas ungewöhnlich*“. Dies löst einen Forschungsprozess aus, in welchem Hypothesen entwickelt werden, die auf der einen Seite über

---

<sup>4</sup> Anhand dieses Beispiel wird auch klar, dass mit Krisensituation keine existentielle Krise gemeint ist. Vielmehr bedeutet eine Krisensituation zunächst einmal nur, dass die üblichen Handlungsroutinen nicht greifen.

die Nutzungsroutine hinausgehen müssen (da diese ja gerade in die Krise geraten ist), auf der anderen Seite an diese zurückgebunden bleiben (da nur von den bestehenden Überzeugungen die neue Situation interpretiert werden kann). Dieses Spannungsverhältnis kann nur aufgelöst werden, wenn man die entwickelten Hypothesen irgendwie an der Realität prüft. Dabei ergibt sich ein offener zyklischer Prozess von Reflektion und Aktion, der strukturell viele Ähnlichkeiten zu dem aufweist, was der Pragmatist H. Joas mit "Kreativen Handeln" bezeichnet hat [Joas96]. Insbesondere kann zu Beginn dieses Prozesses nicht entschieden werden, welche Handlungsstrategie letztendlich zielführend ist. Entsprechend kann es vorkommen, dass der Akteur innerhalb dieses Prozesses verschiedene Handlungsstrategien ausprobiert.

Auf dieser Ebene kann das Artefakt *epistemisch* die Funktion eines *Boundary Objects* übernehmen, wenn im Rahmen des Umgangs mit dem Artefakt gemachte Erfahrungen als Bindeglied zwischen seiner symbolischen Interpretation und der in die Krise geratenen Handlungsroutine fungieren kann. Als Folge des Forschungsprozesses kann dabei sowohl die symbolische Interpretation als auch die Handlungsroutine transformieren werden.

## **5.2 Die mediale Dimension: Visuelle Annotation und die Referenzierung des Nutzungskontext**

Wie oben beschrieben gründet eine Krisensituation in einer momentanen Situation und verweist höchst indexikalisch auf die aktuellen *Objects of Interests*. Dies hat auch Konsequenzen für die mediale Dimension einer Aneignungsunterstützung. So zeigte sich in der Evaluation der Beiträge, dass sich die Nutzer in ihren Beiträgen implizit oder explizit auf das Softwareartefakt und die aktuelle Nutzungssituation beziehen. Dabei wird häufig auf die aktuelle Benutzerschnittstelle Bezug genommen, in der die Krisensituation auftrat. Des Weiteren haben wir festgestellt, dass die Funktionalität, einen Screenshot vom aktuellen Zustand der Applikation zu erstellen, ein sehr hilfreiches Werkzeug dafür ist, ein ‚Etwas‘ des ‚Hier und Jetzt‘ einer aktuellen Krisensituation einzufangen. Des Weiteren wird das eingebaute graphische Annotierungswerkzeug nicht nur dazu verwendet, auf die Problemsituation zu verweisen, sondern auch dazu, ein alternatives Design zu illustrieren.

Auf dieser Ebene kann das Artefakt *medial* die Funktion eines *Boundary Objects* einnehmen, indem es eine gemeinsame Interpretationsbasis herstellt, mittels derer sich die indexikalischen Bezüge innerhalb Kommunikation zwischen den verschiedenen Akteuren verorten lassen.

### 5.3 Die politische Dimension: Ich, meine Community und der Hersteller

Die Entwicklung und Nutzung von Technologien ist eingebettet und beeinflusst von den vorhandenen Machtstrukturen. Als Folge davon besitzt die Aneignung und kreative (Um-)Nutzung auch eine politische Dimension. Dies ist zwar ein allgemein bekanntes Faktum, trotzdem wird dieser Aspekt selten im Kontext von User Innovation diskutiert.

In unseren Evaluationen haben wir verschiedene Indikatoren für die Wichtigkeit der politischen Dimension und den Aspekt des Vertrauens gefunden. Dabei haben wir aus Nutzersicht drei verschiedene Akteure identifiziert, die eine kritische Rolle im Aneignungsprozess spielen: die Gruppe von Freunden und Kollegen, die weltweite Nutzercommunity und den Hersteller. Dabei wird die Rolle des Herstellers von den Nutzern unterschiedlich beurteilt. Auf der einen Seite vertrauen einige Nutzer den Informationen der Hersteller, da sie von offizieller Seite geprüft worden sind. Jedoch haben wir auch die gegenteilige Meinung gefunden. Exemplarisch wird diese Haltung in den Kommentar eines Nutzers ausgedrückt: *„Ich würde den Hilfetexten von anderen Nutzern mehr vertrauen, insbesondere weil der Text von jemand geschrieben wurde, der sich in einen ähnlichen Nutzungskontext befindet.“* Gegenüber großen Herstellern fehlt auch das Vertrauen, dass ihr Feedback Ernst genommen wird und dass dieser etwas bewirken kann.

Ein anderer wichtiger Punkt bei den Nutzern ist die Angst, dass ihre Beiträge von irgendjemand falsch interpretiert werden. Hierbei kann eine zu große Transparenz auch problematisch werden. Ein Ausschnitt aus einer Email Korrespondenz soll diesen Aspekt verdeutlichen. Hierbei haben wir eine Benutzeranfrage per Email bekommen: *“Ein ‚Killer-Feature‘ wäre m.M.n. sicher weiter eine einfache Backup / Transfer-Möglichkeit von Server zu Server.“* Auf Grund unseres Entwicklungsansatzes haben wir die Anfrage auf das öffentliche Design-Diskussionsforum gestellt und den Nutzer per Email darüber informiert: *„Ich habe Deine Anregung auch mal unter [URL] aufgenommen.“* Da die Anfrage jedoch eine politische Implikationen hat, hat uns zweite Email erreicht, in der der Nutzer folgende äußerte *“Eine Bitte: Kannst Du meinen Request nochmal löschen und ich erstelle ihn neu, dann kann ich es etwas präziser formulieren und wir vermeiden ggf. Konflikte mit [Hersteller], die den ‚Server-to-Server‘-Transfer meines Wissens als kommerzielle Dienstleistung anbieten.“* Dieses Beispiel zeigt auf, dass die Kommunikation und einhergehend damit auch die Entwicklung von Gestaltungsideen von dem jeweiligen politischen Kontext abhängen.

Insbesondere kann auf dieser Ebene das Artefakt *politisch* die Funktion eines *Boundary Object* einnehmen, auf die sich die verschiedenen Gestaltungs- und Nutzungsinteressen beziehen und

miteinander ausgehandelt werden können. Der Erfolg eines Artefakts ergibt sich dabei aus seiner Fähigkeit, die unterschiedlichen Interessen der einzelnen Akteure im ausreichenden Maße zu integrieren.

## 6 Zusammenfassung

Der Flexibilisierung von IT-Landschaften kommt eine immer größere Bedeutung zu. In der Forschung werden jedoch meist produkt-orientierte und prozess-orientierte Flexibilisierungsstrategien noch getrennt von einander betrachtet.

Synthetisiert man jedoch beide Richtungen, erkennt man, dass ein integrierendes Glied fehlt, das in der praktischen Umsetzung zwischen beiden Konzepten vermittelt. Bei der Vermittlung kommt dem Endbenutzer eine entscheidende Rolle zu, da er zum einen mit den (geänderten) Anforderungen im Nutzungskontext unmittelbar konfrontiert und zum andern in der Lage ist, auch eigenständig Innovation zu entwickeln.

Das Konzept der Infrastrukturen zur Aneignungsunterstützung erweitert die Ideen der *User Innovation* [vHip88, vHi+02] um das Konzept anpassbarer Software [Tege00, Hen+91], wobei es sich die Eigenschaft der *Direct Manipulation* [Shne83] heutiger Anwendungssysteme zunutze macht, um Meta-Nutzungsfunktionalitäten direkt in das User Interface einzuweben.

Auf der analytischen Ebene konnte das Konzept der *Boundary Objects* [Sta+89] fruchtbar gemacht werden um die einzelnen Dimensionen einer Aneignungsunterstützung zu thematisieren. Dabei zeigt sich, dass die materielle Struktur des Artefakts nicht nur in der Nutzung, sondern auch im Diskurs hierüber eine nicht zu vernachlässigende Rolle spielt. Insbesondere zeigt die genaue Analyse von Krisensituation, dass die Trennung der verschiedenen Aspekte der Nutzungsreflektion aus der Perspektive des Nutzers problematisch ist.

Im BSCWeasel-Projekt konnte gezeigt werden, dass sich weite Teile des Konzepts der Infrastrukturen zur Aneignungsunterstützung mit Hilfe der durch Eclipse bereitgestellten Möglichkeiten auf eine effiziente Art und Weise umsetzen lassen [Ste+04, Ste+06a, Ste+06b].

Erste praktische Erprobungen der Infrastruktur beim BSCWeasel-Projekt deuten auch an, dass die hier vorgestellte technische Infrastruktur schon heute dazu beitragen kann, das Potential der Nutzer in einem kontinuierlichen Entwicklungsprozess besser zu nutzen. Jedoch hat sich auch bei der technischen Umsetzung gezeigt, dass die Architektur von Eclipse nicht darauf anlegt ist,

den Übergang von Nutzung zur Meta-Nutzung zu fördern. Inwieweit dies nachträglich durch geeignete Erweiterung behoben werden kann, ist Teil der zukünftigen Arbeiten der vorgestellten, prototypischen Umsetzung der Infrastruktur.

## References

- [Beck00] Beck, K.: Extreme Programming Explained: Embrace Change. 2000, Pearson Education: Addison-Wesley.
- [Ben+95] Bentley, R., Dourish, P.: Medium versus mechanism: Supporting collaboration through customisation. In: Proceedings of ECSCW'95, 1995. Sweden: Kluwer.
- [Bryn98] Brynjolfsson, E., Hitt, L.M.: Beyond the productivity paradox. Communications of the ACM, 1998. 41(8): S. 49-55.
- [Fis+02] Fischer, G., Ostwald, J.: Seeding, Evolutionary Growth, and Reseeding. In: Proc. of PDC 02, CPSR, 2002, S. 292-298.
- [Fis+90] Fischer, G., Girgensohn, A.: End-user modifiability in design environments. In: Proceedings of the SIGCHI. 1990, ACM Press, 1990, S. 183 - 191.
- [Flo+89] Floyd, C., Reisin, F.-M., Schmidt, G.: STEPS to Software Development with Users Source. In: Proc. of the European SE Conf., Springer-Verlag, 1989, S. 48-64.
- [Har+96] Hartson, H.R., Castillo, J.C., Kelso, J., Kamler, J., Neale, W.: Remote Evaluation: The Network as an Extension of the Usability Laboratory. In Proc. of CHI'96, 1996, S. 228-235
- [Hen+91] Henderson, A., Kyng, M.: There's No Place Like Home: Continuing Design. In: Use, in Design at Work, 1991, S. 219 - 240.
- [Joas96] Joas, H., Die Kreativität des Handelns, 1996, Suhrkamp.
- [Kar+98] Karsten, H., Jones, M.. The long and winding road: Collaborative IT and organisational change. In: Int. Conference on CSCW'98. ACM Press, 1998, S. 28-38.



- [Mal+95] Malone, T.W., Lai, K.-Y., Fry, C.: Experiments with Oval: a radically tailorable tool for cooperative work. In: ACM TOIS. 13 (2), 1995, S. 177 - 205.
- [Mørch97] Mørch, A.: Three Levels of End-user Tailoring: Customization, Integration and Extension. In: Computers and Design in context, MIT Press, 1997, S. 51 – 76.
- [Ober94] Oberquelle, H., Situationsbedingte und benutzerorientierte Anpassbarkeit von Groupware. In: Menschengerechte Groupware, Teubner, 1998, S. 31-50.
- [Orl+97] Orlikowski, W.J., Hofman, J.D.: An Improvisational Model for Change Management.. In: Sloan Management Review, 1997, S. 11-21.
- [Orli96] Orlikowski, W.J.: Evolving with Notes: Organizational change around groupware technology. In Groupware & Teamwork, J. Wiley, 1996, S. 23 – 60.
- [Oeve97] Oevermann, U.: Thesen zur Methodik der werkimmanenten Interpretation vom Standpunkt der objektiven Hermeneutik, Manuskript, gehalten auf der 4. Arbeitstagung der Arbeitsgemeinschaft objektive Hermeneutik e.V. 1997: Frankfurt a. M
- [Pipe05] Pipek, V.: From Tailoring to Appropriation Support: Negotiating Groupware Usage. PhD Thesis, University of Oulu: Oulu, Finland, 2005.
- [Pip+99] Pipek, V., Wulf, V.: A Groupware's Life. In: Proc. of ECSCW '99, Kluwer, 1999, S. 199 – 219.
- [Rum+01] Rumpe, B., Schröder, A.: Quantitative Untersuchung des Extreme Programming Prozesses, Technischer Bericht TUM-I0110 and ViSEK/006/D2001, 2001.
- [Shne83] Shneiderman, B.: Direct manipulation: a step beyond programming languages. In: *IEEE Computer* 16(8), 1983, S. 57-69.
- [Sta+89] Star, S.L., Griesemer, J.R.: Institutional Ecology, Translations and Boundary Objects. In: *Social Studies of Science*, 19, 1989, S. 387-420.
- [Ste+04] Stevens, G., Budweg, S., Pipek, V.: The "BSCWeasel" and Eclipse-powered Cooperative End User Development. In: Workshop at the CSCW'04, 2004.

- [Ste+06a] Stevens, G., Wiedenhöfer, T.: CHIC - A pluggable solution for community help in context. In: Proc. of the NordCHI 2006 (in Druck).
- [Ste+06b] Stevens, G. Draxler, S.: Partizipation im Nutzungskontext. In: Konferenzband Mensch & Computer 2006 (in Druck).
- [Stiem00] Stiemerling, O.: Component-Based Tailorability. Dissertation, Institut für Informatik III., Rheinische Friedrich-Wilhelms-Universität Bonn, 2000.
- [Strü99] Strübing, J.: Digging the Gold Mine. Vortragsmanuskript, gehalten an der Spring School on STS, Zurich, March, 3, 1999.
- [Teeg00] Teege, G.: Users as Composers: Parts and Features as a Basis for Tailorability. In: Computer Supported Cooperative Work, 9(1), 2000, S. 101-122.
- [Tör+03] Törpel, B., Pipek, V., Rittenbruch, M.: Creating Heterogeneity - Evolving Use of Groupware in a Network of Freelancers. In: Computer Supported Cooperative Work, 12(1-2), 2003.
- [vHip+02] von Hippel, E., Katz, R.: Shifting Innovation to Users via Toolkits, In: Management Science, 48(7), 2002, S. 821-834.
- [vHip88] von Hippel, E.: The sources of innovation, Oxford University Press, 1988.
- [Wul+01] Wulf, V., Golombek, B.: Exploration environments: concept and empirical evaluation. In: Proc. of the GROUP. 2001, S. 107-116.
- [Wul+95] Wulf, V., Rohde, M.: Towards an Integrated Organization and Technology Development. In: Proc. of Designing Interactive Systems. 1995, S. 55-64.
- [Wulf99] Wulf, V.: Evolving Cooperation when Introducing Groupware – A Self-Organization Perspective. In: Cybernetics and Human Knowing, 6(2), 1999, S. 55 – 75.