

# Usability Enhancement für die intelligentPad Technologie - Das Cockpit Konzept

Stefan Löw, Christian Wölfert  
intellipaxx GmbH  
Rathausplatz 4  
66111 Saarbrücken, Germany  
{stefan.loew;christian.woelfert}@intellipaxx.com

Gerhard Degel  
circle New Media & Coporate Design  
Josefstaler Str. 133  
66386 St. Ingbert, Germany  
gerhard@circle-media.de

**Abstract:** Im Internet des neuen Jahrtausends spielt die Auffindbarkeit von Informationen eine zentralere Rolle als jemals zuvor. Es gilt den Benutzer schnell, elegant und vor allem effizient genau zu den Informationen gelangen zu lassen, die er wünscht. Die IntelligentPad Technologie versucht, mit Hilfe einfach zu bedienender Mechanismen, hierzu einen Beitrag zu leisten. Dieser Beitrag befasst sich mit der intelligent Pad Technologie in der Implementierungsvariante CHIP<sup>1</sup>, welche am Meme Media Laboratory unter der Leitung von Prof. Dr. Yuzuru Tanaka erdacht [Tan03] und implementiert wurde. Diese Technologie verspricht dem Benutzer eine Reihe von Vorteilen bei der Lösung seiner Navigationsaufgaben innerhalb der Informationsflut auf dem PC, den Netzwerken und dem Internet. Doch ohne Unterstützung des Benutzers seitens der Software droht auch in der „IntelligentPad-Welt“ Navigationsverlust und somit Verlust der Kontrolle über die Technologie und damit verbunden natürlich auch keine Lösung der eigentlichen Aufgabe(n) des Benutzers. Dem entgegen stehen zwei Implementierungen, welche gemeinsam das sog. Cockpit Konzept für die IntelligentPad Technologie realisieren und die Usability nach Ansicht der Autoren maßgeblich fördern. Gerade im Bereich von Business Applikationen zeigen die Erfahrungen, dass der Benutzer kein „Computer-Freak“ ist und sich mehr und mehr Software wünscht, welche er intuitiv und sicher bedienen kann und die ihm auch soviel Arbeit wie möglich abnimmt.

## 1 Einleitung

Die IntelligentPad Technologie wurde von Yuzuru Tanaka am Meme Media Laboratory der Hokkaido University erdacht. Ihre Bausteine werden als Pads bezeichnet. Ein solches Pad beschreibt ein interaktives Meme Medium, es speichert als Software faktisches und

---

<sup>1</sup>Collaborating Host Independent Pads

prozedurales Wissen. Komplexe Anwendungen können durch die funktionale Kombination einer Vielzahl von Pads entstehen.

Das Verhalten von Pads entspricht dem von Memes. So wie die Gesamtheit der Memes einer Informationsgesellschaft einen Meme-Pool oder ein Biotop (vgl. [Jan05]) bildet, so entsteht aus einer Vielzahl von intelligenten Pads eine Art Pad-Pool. Memes werden aus dem Pool ausgewählt, verändert, mit anderen Memes zu neuen, komplexeren Memes kombiniert. Die Ergebnisse dieser Prozesse kehren wieder in den Pool zurück. So kann der Pool gut als Abbild aller zu einem bestimmten Zeitpunkt wertvollen und interessanten Informationen betrachtet werden. Für detaillierte Informationen über Memes sei der Leser auf das Buch von Susan Blackmore „The Meme Machine“ [Bla99] verwiesen.

So trivial es klingen mag, der Verwendung dieser Informationen geht zunächst die Kenntnis voran, daß sie überhaupt existieren. Für die Memes wurde und wird das Problem gut gelöst. Bücher, Bibliotheken, Fachzeitschriften, Magazine und das Internet machen auch große und vielfältige Mengen an Informationen für Auswahl und Nutzung verfügbar.

Für die Erstellung von Anwendungen mit der IntelligentPad Technologie gelten diese guten Voraussetzungen nicht. Dieser Prozess aus Auswahl und Veränderung bedeutet Evolution: Nur die erfolgreichen Pads und ihre Anwendungen werden im Pad-Pool bestehen können und der Pad-Pool wird eine Menge effizienter und an aktuelle Anforderungen angepaßter Software bilden.

Wurde eine Anwendung bereits konstruiert, kann sie durch den Austausch von Pads oder die Erweiterung mit anderen Pads an sich verändernde Anforderungen angepasst werden. Auch die veränderten Anwendungen kehren in den Pad-Pool zurück.

Existieren bereits Pads, die meine Anforderungen erfüllen? Welche Funktionen erfüllt ein Pad? Was kann dieses bestimmte Pad besser als andere? Ist es mit meiner Anwendung und dem Datenformat der beteiligten Pads kompatibel? Diese Fragen können bisher nicht beantwortet werden. Der Konstruktionsprozess einer aus Pads zusammengesetzten Anwendung kann somit leicht bereits bei der Auswahl der Komponenten scheitern.

Welche Pads benötige ich für meine Anwendungen und nach welchem Muster füge ich sie zusammen? Auch die Veränderung einer bestehenden komplexen Anwendung ist nicht trivial. Welcher Teil meiner grafischen Oberfläche gehört zu welchem Pad? Welcher Teil der Funktionalität wird abgebildet? Benötige ich neue Pads oder genügt es, wenn ich die vorhandenen Pads neu zusammensetze? Mit der Auswahl eines neuen Pads steht man wieder am Anfang.

Eine Antwort auf diese Probleme ist nach unserer Meinung die Entwicklung zweier voneinander getrennten Anwendungen. Das CommandCenter dient der Bereitstellung und der Auswahl von Pads, indem es eine Visualisierung des Pad-Pools erstellt. Die Bereitstellung von Informationen über einfache Pads oder zusammengesetzte Anwendungen ist Aufgabe des ControlCenters.

Ein wichtiger Aspekt ist die Erstellung Metadaten. Sie bilden schließlich die Grundlage für die Auswahl und Komposition von Pads und werden die Wege beschreiben, die ein Benutzer der IntelligentPad Technologie zu ihrer erfolgreichen Benutzung beschreitet.

## 2 Das Cockpit Konzept

Zunächst gilt es die Bedürfnisse eines potentiellen Benutzers der intelligent Pad Technologie am Beispiel der *CHIP*-Implementierung<sup>2</sup> zu analysieren. Diese Implementierung stellt dem Benutzer Pads als Softwarebausteine mit elementarer Funktionalität zur Verfügung (z.B. Texteingabe Bausteine, oder Bilddarstellungsbausteine), die ähnlich dem LEGO-Prinzip zusammengesteckt werden können. Dabei lässt sich zunächst grundsätzlich feststellen, dass ein Benutzer an zwei Aspekten im Umgang mit Pads interessiert ist:

1. Kontrolle und Management von elementaren Pads sowie Pad-Applikationen.
2. Informationsgewinnung über elementare Pads sowie Pad-Applikationen.

Differenziert man die genannten Aspekte weiter aus, so ergibt sich für den Bereich der Kontroll- und Managementfunktionen:

- Starten/Beenden elementarer Pads.
- Starten/Beenden von Pad-Applikationen.
- Verwalten von Pad-Applikationen in strukturierter Form (z.B. Baumstruktur).
- Anpassen von Pad-Applikationen auf unternehmensspezifische Anforderungen (Corporate Design).

Für den Bereich der Informationsgewinnung ergibt sich:

- Abfrage aller relevanten Daten (Lage auf dem Bildschirm, enthaltene Informationen, Slotverbindungen zu anderen Pads, etc.).
- Abfrage aller Metadaten für ein elementares Pad oder eine Pad-Applikation.
- XML-Struktur der Pad-Applikation.

Nach Ansicht der Autoren können diese unterschiedlichen Anforderungen nicht in einer Applikation zusammengefasst werden, denn anstatt dem Benutzer zu ermöglichen schnell, effizient und intuitiv mit der IntelligentPad Technologie umgehen zu können, wäre eine Applikation entstanden, die ihrerseits höchste Anforderungen an den Benutzer stellen würde, weil sie einfach „funktionsüberladen“ und wenig intuitiv wirken würde.

Aus diesem Grund geht das Cockpit Konzept von einer Zweiteilung der Anforderungen aus und implementiert die beiden Bereiche in streng getrennten Applikationen, die jede für sich dem Benutzer exakt einen Funktionsbereich zur Verfügung stellt. Das CommandCenter ist hierbei für die Kontroll- und Managementaspekte verantwortlich, während das ControlCenter den Benutzer mit allen Informationen versorgt, die er über Pad Applikationen haben möchte.

---

<sup>2</sup>siehe <http://www.meme.hokudai.ac.jp/> für weitere Informationen zu dieser Implementierung.

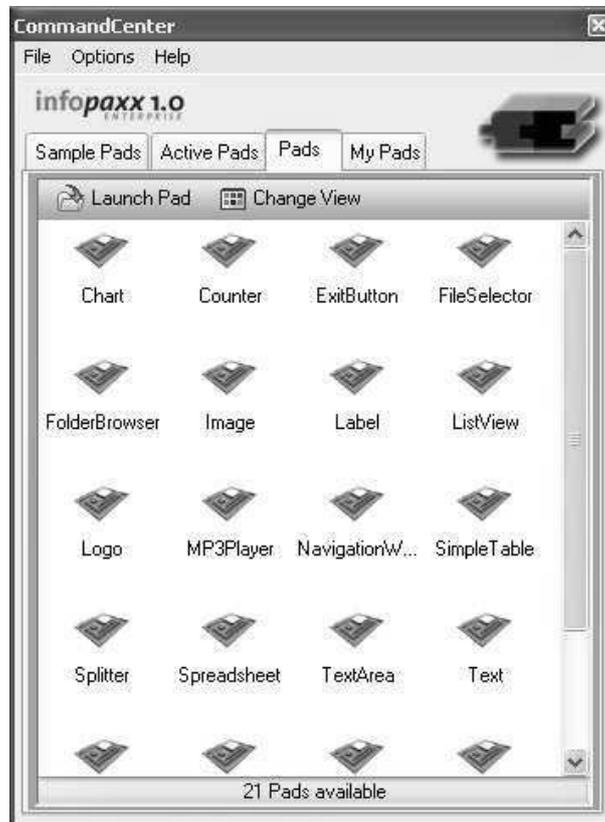


Abbildung 1: Das geöffnete CommandCenter zeigt alle verfügbaren elementaren Pads an.

### 3 CommandCenter

#### 3.1 Funktionalitäten

Das CommandCenter stellt dem Benutzer eine Reihe von Funktionalitäten und Konfigurationsmöglichkeiten im Umgang mit Pads zur Verfügung. Beginnend bei den elementaren Bausteinen bestehen zunächst die grundsätzlich notwendigen Funktionalitäten, wie z.B. die Anzeige aller verfügbaren Pads oder die Möglichkeit diese zu starten.

Fertiggestellte Pad-Applikationen können in einem eigens dafür geschaffenen Bereich („Meine Pads“) abgelegt und verwaltet werden. Dieser Bereich ermöglicht es dem Benutzer die Pad-Applikationen mit einem eigenen Namen zu versehen und die Organisation der Applikationen in einer Baumstruktur, welche mit Hilfe von Ordnern weiter untergliedert und kategorisiert werden können. Die Ablage der Applikationen in diesem Bereich des CommandCenters geschieht mittels den gängigen Drag&Drop-Mechanismen, welche

dem Benutzer aus seinem Umgang mit Microsoft Windows bereits intuitiv vertraut sind.

Ein weiterer Bereich des CommandCenters ist ein sog. Task-Manager („Aktive Pads“ im CommandCenter), wie er in ähnlicher Form ebenfalls bereits in das Windows-Betriebssystem integriert ist. Hierbei sieht der Benutzer auf einen Blick, welche Pads er bereits zum Arbeiten aktiviert hat. Zusätzliche Angaben, wie die Namen der Pads, sowie z.B. deren aktuelle Laufzeit werden ebenfalls mitgeteilt. Sollte der Fall eintreten, dass ein oder mehrere Pads nicht mehr reagieren, besteht die Möglichkeit diese Pads über den Task-Manager zu deaktivieren, also den Task zu beenden.

Der vierte und letzte große Bereich des CommandCenters („Anwendungen“) resultiert aus den Erfahrungen und Analysen der Bedürfnisse im Business Bereich. Unternehmen haben ein Interesse daran, dass nicht unbedingt alle Mitarbeiter, die mit der IntelligentPad Technologie arbeiten, auch bereits erstellte Pad-Applikationen manipulieren (umbauen) oder neue Applikationen kreieren. Weiterhin kann der Bedarf bestehen eine Pad-Applikation komfortabel an mehrere Mitarbeiter parallel auszuliefern ohne diese auf jedem Arbeitsplatz explizit neu erstellen zu müssen. Im Bereich „Anwendungen“ können also Pad-Applikationen quasi distribuiert werden um diese beispielsweise unmittelbar nach der Installation bereits auf einem Arbeitsplatz zur Verfügung stellen zu können. Der Mehrwert entsteht dadurch, dass ein Benutzer, sofern dies für seine unmittelbare Aufgabe nicht relevant ist, die Erstellung von Pad-Applikationen nicht beherrschen muss, sondern eine Applikation erhält mit der er sofort arbeiten kann.

### **3.2 Auto-PlugIn**

Über die Konfigurationsmöglichkeiten des CommandCenters hat der Benutzer die Option das neu entwickelte Auto-PlugIn zu aktivieren. Hierbei kann der Benutzer zwei beliebige Pads direkt miteinander verbinden, einfach durch das „übereinanderlegen“ eines Pads und eines darunterliegenden Slots. Diese Funktionalität wird dann auch visuell unterstützt (verbindungsreifer Slot leuchtet grün auf), was dem Benutzer die Arbeit weiter vereinfacht.

### **3.3 Konfiguration von Styles**

Der praktische Einsatz hat gezeigt, dass über die eigentliche Arbeit mit Pads hinaus auch der Anspruch besteht, Pads und Pad-Applikationen in der CI des Unternehmens oder zumindest in einer Gestaltung, welche der CI gleicht, darzustellen. Über das CommandCenter hat der Benutzer die Möglichkeit dies zu konfigurieren und seine Konfiguration entsprechend auf seinem System zu speichern. Diese Konfiguration kann bei Bedarf auch mit einer Pad-Applikation weitergegeben werden, so daß auch die CI des Unternehmens beispielsweise bei einem Einsatz bei Kunden nicht verloren geht (s. Abb. 2).



Abbildung 2: Konfigurationsfenster für PadStyles.

### 3.4 Multilingualität

Die IntelligentPad Implementierung stammt, wie bereits mehrfach erwähnt vom Meme Media Laboratoy unter der Leitung von Prof. Yuzuru Tanaka in Japan. Betrieben wird Sie inzwischen sowohl in Deutschland, als auch in den Vereinigten Staaten von Amerika und weiteren Ländern dieser Welt. Zwar hat sich die englische Sprache speziell in der wissenschaftlichen und geschäftlichen Welt zur „Standardsprache“ entwickelt, aber es ist nicht immer davon auszugehen, dass jeder Mensch dieser Erde neben seiner eigenen Sprache das englische Idiom beherrscht. Gleichwohl kann aber angenommen werden, dass nahezu jeder Mensch ein potentieller Anwender der IntelligentPad Software sein könnte. Aus diesem Grunde wurde in das CommandCenter eine Funktionalität implementiert, welche automatisch feststellt auf welchem Sprachsystem die Software betrieben wird. Arbeitet das zugrunde liegende Windows System in englischer Sprache, so wird auch die IntelligentPad Software automatisch in einer englischen Sprachvariante betrieben. Man muß jedoch erwähnen, dass alle zur Verfügung gestellten Sprachen vorher in der Software hinterlegt werden müssen. Existiert für das System also keine japanische Sprachdatei, erkennt es zwar ein zugrund liegendes japanisches Windows, kann die IntelligentPad Software jedoch nicht in die korrekte Sprachversion umschalten. In einem solchen Falle wird dann dennoch auf die englische Fassung ausgewichen.

## 4 ControlCenter

Die mit Hilfe des CommandCenters zusammengestellten Applikation aus einzelnen Pads, können in sogenannten „chp-Files“ abgespeichert werden. Diese „chp-Files“ sind nichts anderes als Baupläne der Applikation im XML-Format. Als Beispiel für einen solchen Bauplan soll nachfolgendes Codebeispiel dienen:

```
<MP3Player_View assembly="file://MP3Player_Pad.dll">
  <FormBorderStyle>
    <FormBorderStyle>SizableToolWindow</FormBorderStyle>
  </FormBorderStyle>
  <Location>
    <X>533</X>
    <Y>260</Y>
  </Location>
  <Size>
    <Width>321</Width>
    <Height>254</Height>
  </Size>
</Size>
<ParentSlot />
<PrimarySlot>MP3-File</PrimarySlot>
<Model>
  <MP3Player_Model>
    <Slots>
      <Slot name="MP3-File" />
    </Slots>
  </MP3Player_Model>
</Model>
```

Der vorliegende Bauplan zeigt, dass ein Pad durch eine Reihe von Informationen beschrieben wird, die zum Beispiel zunächst sein „Aussehen“ auf dem Bildschirm betreffen (s. z.B. <Location>- oder <Size>-Tags). Mit „ParentSlot“ wird beschrieben welcher „Eingang“ eines Pads aktuell in der Lage ist Informationen aufzunehmen und zu verarbeiten, für den Fall, dass eine Verknüpfung mit einem weiteren Pad hergestellt wird.

Solche XML-basierten Pläne sind für den nicht versierten Benutzer, nicht nachvollziehbar. Das bedeutet, wenn der Benutzer wissen möchte welche Pads in seiner selbst entwickelten Applikation eingebaut sind, wie sie miteinander verbunden sind und wie sie untereinander Informationen austauschen, wird er es schwer haben, dies an Hand des XML-Bauplans zu rekonstruieren. Deshalb bietet das ControlCenter als Hauptfunktionalität, eine graphische Visualisierung dieses XML-Bauplans. Diese Visualisierung erlaubt es weiterhin zusätzliche Informationen zu den Pads abzurufen und auch, allerdings eingeschränkt, in einer sehr benutzerfreundlichen Art und Weise solche abgerufenen Informationen zu manipulieren. Die Darstellung dieser Pläne in textueller Form wird zusätzlich angeboten, da sie dem wirklichen „Power-User“ wesentlich mehr Manipulationsmöglichkeiten für die Pad-

Applikation bietet. Im Folgenden soll jedoch nur die visuelle Darstellung näher erläutert werden.

#### 4.1 Graphische Visualisierung

Eine Pad-Applikation ist bezogen auf ihren Bauplan immer in der gleichen Struktur aufgebaut. Es existiert immer ein Pad (Root-Pad, Basis-Pad), an dem sogenannte „Child-Pads“ angebunden sind. An diesen Child-Pads können wiederum Child-Pads verbunden sein, so dass der Aufbau von Pad-Applikationen einer Baumstruktur gleicht. Es ist daher naheliegend, den visualisierten Bauplan ebenfalls in einer Baumstruktur (TreeView) wiederzugeben. Die einzelnen Knoten dieser Baumstruktur werden von den beteiligten Pads in der gerade visuell dargestellten Applikation gebildet. Diese Knoten (Pads) sind wiederum in unterschiedlichen Farben, und mit ihrem Namen gekennzeichnet. Es besteht darüberhinaus die Möglichkeit ein bestimmtes Pad (Knoten), in einem „Property-Window“ mit seinen Eigenschaften anzeigen zu lassen und einzelne Eigenschaften, innerhalb dieses „Property-Windows“ komfortabel zu editieren.

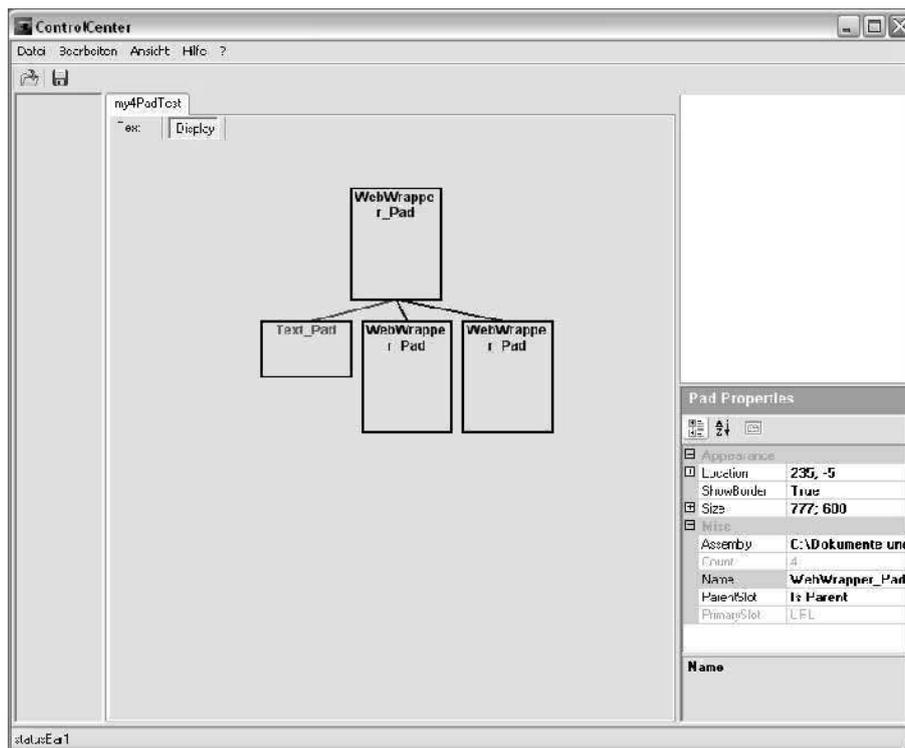


Abbildung 3: Prototyp des Control Centers

## 5 Der Einfluß von Metadaten

*Metadaten* heißt ein Schlagwort der Informationstechnologie, welches zur Zeit in aller Munde ist und auch immer mehr an Bedeutung gewinnt. Metadaten sind einfach gesagt „Daten über Daten“. Auch für die IntelligentPad-Technologie spielen Metadaten eine wichtige Rolle. Das Ausstatten elementarer Pads mit Metadaten ist unabdingbar, reflektiert man an dieser Stelle noch einmal die eingangs gemachten Bemerkungen über Biotop, Evolution und Austauschbarkeit von Pads. Doch Metadaten lösen nicht nur Probleme, sie schaffen auch neue.

Mit Hilfe von Metadaten werden Pads charakterisiert. Was kann ein Pad? Wofür kann ich es einsetzen? Diese Fragen müssen durch die Metadaten beantwortet werden und dies stellt einen nicht zu unterschätzenden Aufwand dar. Jedes Pad und jede Pad-Applikation muss annotiert werden. Doch auch hier taucht direkt die Anschlußfrage auf, wie diese Annotation aussieht, damit sie „richtig“ (einheitlich) ist. Schließlich auch noch die Frage nach einer Unterstützung der Annotation durch ein geeignetes Werkzeug. Wenn man in Biotopen navigieren will und wenn Evolution unter Pads stattfindet, wird schnell klar, dass mit zunehmendem Wachstum auch Verlust der Übersicht droht. Dies kann zum Problem werden, wenn man darüber nachdenkt dem Benutzer auch Hilfestellungen anbieten zu wollen, wie zum Beispiel Vorschläge, welche Pads mit seiner vorliegenden Konstruktion verknüpfbar wären und zu welchen Ergebnissen dies führen könnte. Auch wenn der Benutzer das sprichwörtliche Rad neu erfindet, sollte ein Assistent (z.B. das CommandCenter) ihn darauf aufmerksam machen, dass ein solches Konstrukt bereits erfunden wurde. Metadaten helfen dieses und andere Probleme die bei zunehmender Komplexität und Masse von Pads und Pad-Applikationen auftreten können zu lösen und wiegen damit deutlich schwerer, als die vermeintlichen Mehraufwände durch zusätzliche Annotation.

Ein Metadatenkonzept sowohl für elementare Pads, als auch Pad-Applikationen befindet sich derzeit in der Entwicklung. Dazu muss natürlich im Sinne einer vereinfachten Bedienung ein entsprechender Editor gebaut werden, der es dem Benutzer ermöglicht schnell und einfach an diese Daten zu gelangen und diese wiederum effizient einsetzen zu können.

## 6 Ausblick

Ein sehr aktuelles Thema, welches auch vor dem Hintergrund von wachsenden Pad-Biotopen und Evolution von Pads eine bedeutende Rolle spielen wird, sind Semantik und Ontologien. Die IntelligentPad-Technologie wird oft mit dem System von Lego verglichen. Hauptgrund dafür ist mit Sicherheit, dass bei der Pad-Technologie, genau wie bei Lego auch, jeder Baustein prinzipiell mit jedem anderen verbunden werden kann, ohne dass dem Benutzer irgendwelche Grenzen gesetzt werden. Die Frage ist, ob es überhaupt sinnvoll ist was ein Benutzer gerade „zusammenstecken“ will. An dem Vorgang des Zusammensteckens selbst wird er nicht gehindert, bzw. bekommt kein „Feedback“ ob die eben durchgeführte Aktion zu einem Ergebnis führt, welches für den Benutzer einen Mehrwert bringt. Die Lösung dieser Problematik liegt nach Ansicht der Autoren in Semantik und

Ontologien. An Hand von Metadaten, mit denen Pads ausgestattet werden sollen, könnten sich mögliche semantische Beziehungen bzw. Ontologien zwischen Pads entwickeln lassen. Man kann Beziehungen entwerfen, die darauf beruhen, dass man bestimmte Regeln festlegt. Solche Regeln sehen auf Grund bestimmter Charakteristiken (Metadaten) eines Pads so aus, dass eine sinnhafte Verknüpfung zwischen Pad X und Pad A, oder zwischen Pad C und Pad F entsteht, falls diese miteinander verknüpft werden. Im Gegenzug wird natürlich auch festgelegt, dass es keinen Sinn macht Pad X mit Pad C in Verbindung zu bringen. An dieser Stelle taucht die Gefahr auf, dass die Möglichkeit der Kombination von Pads für den Benutzer bereits soweit beschnitten wird, dass einer der Vorteile der Technologie verloren geht, nämlich jedes Pad mit jedem anderen verknüpfen zu können. Eine Möglichkeit dem entgegen zu wirken wäre zum Beispiel, dem Benutzer während seines Kompositionsvorgangs mit Vorschlägen zu unterstützen, ohne ihn bei seiner Auswahl einzuschränken. An Hand der vorher beschriebenen Regeln, könnte man ihm, ähnlich wie es eine Entwicklungsumgebung mit Methodenvorschlägen (Code Completion) macht, eine Auswahl von Pads anbieten, die besonders gut geeignet wären sich mit dem aktuellen Pad zu verbinden. Er kann aber auch eine ganz andere Komponente wählen und selbst erfahren, ob es Sinn macht oder nicht. Man kann soweit gehen und die Behauptung aufstellen, dass die Evolution schon „aussieben“ wird, was nicht geht.

Eine weitere Möglichkeit wäre auf der Basis von Metadaten und Ontologien nach Design Templates zu suchen. Baut nun ein Benutzer ein paar Pads zusammen, dann ist er nach den ersten wenigen Schritten noch nicht fertig, hat aber eine konkrete Baumstruktur. Diese wird in vielen Fällen völlig neu sein. Aber man kann von den konkreten Pads abstrahieren und nur bestimmte Metadaten betrachten, die nun auch baumstrukturiert sind. Dann gilt es zu untersuchen, ob diese Struktur auch schon an anderer Stelle vorgekommen ist, vielleicht sogar so oft, daß es als Muster gespeichert wurde. Während des Kompositionsvorgangs des Benutzers könnte die aktuelle entstandene und weiter entstehende Baumstruktur immer wieder mit bereits bestehenden Mustern verglichen werden. Gibt es Ähnlichkeiten oder sind sie womöglich sogar identisch, wird der Benutzer wieder mit Pad-Vorschlägen versorgt die auf ein bestehendes Muster passen. Dieser Vergleich von aktueller Komposition und bestehenden Mustern könnte immer nach dem Anschluss eines weiteren Pads erfolgen. Diese Vorschläge können nun vom Benutzer angenommen werden, was ihn jedoch in eine bestimmte Richtung lenken wird. Er kann aber auch hier die Vorschläge bewußt ignorieren und ganz nach seinem freien Willen seinen Pad-Verbund weiterentwickeln und seine Erfahrungen sammeln.

## Literatur

- [Bla99] Susan Blackmore. *The Meme Machine*. Oxford University Press, 1999.
- [Jan05] Klaus P. Jantke. The Biotope Issue in Meme Media Implementations. In *Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets*, Volume 3359 of *LNAI*, Seiten 99–107. Springer, 2005.
- [Tan03] Yuzuru Tanaka. *Meme Media and Meme Market Architectures*. IEEE Computer Society Press, 2003.