

# DeepaMehta — Technisches Strategietreffen 1.10.2007

Annette Leeb, Jürgen Neumann, Matthias Staps, Malte Reißig, Jörg Richter

Paradigmen	1
“Kognitiv Kompatibel”	1
Privatsphäre	1
Gemeinsame Handlungsräume	1
Stand der Entwicklung	2
Personalisierung von Topic Maps	2
Rollen- und Rechte-System	2
Notwendige Arbeiten	4
Erweiterung der Kollaborationsfunktionen	4
Neukonzeption des Rollen- und Rechte-Systems	4
Neukonzeption der GUI	5

## Paradigmen

### “Kognitiv Kompatibel”

Alles was der Benutzer auf dem Bildschirm sieht, ist seine persönliche Sicht. Der Benutzer (nicht das Programm) hat die volle Kontrolle über den Bildschirm. Er bestimmt zu 100% selbst, was er sehen möchte, wie die Dinge auf dem Bildschirm angeordnet sind, und was er *nicht* sehen möchte. Dadurch erhält der Benutzer die Möglichkeit, den Bildschirm gemäß seiner persönlichen inneren kognitiven Karte zu gestalten. Der Bildschirm wird zu einem externen visuellem Zwischenspeicher, der (im Idealfall) stets den inneren geistigen Zustand des Benutzers widerspiegelt. Die Benutzerschnittstelle verschwindet, die Maschine wird zum veritablen Denkwerkzeug.

### Privatsphäre

Alles, was der User tut, tut er in seiner Privatsphäre. Er soll sich niemals beobachtet fühlen und seiner Kreativität und Experimentierfreude freien Lauf lassen.

Derzeitige Konzeption: Die Privatsphäre wird durch den “Personal Workspace” repräsentiert. Jeder DeepaMehta-Benutzer hat genau einen Personal Workspace. Der Personal Workspace dient zur Ablage der persönlichen Topic Maps und ist selbst eine Topic Map. Neue Topic Maps werden grundsätzlich im Personal Workspace angelegt.

### Gemeinsame Handlungsräume

Zum gemeinsamen themenbezogenen Arbeiten gibt es gemeinsame Handlungsräume (“Shared Workspaces”). Shared Workspaces haben verschiedene Zwecke:

- Ablageort für gemeinsam benutzte Sichten (“Topic Maps”)

- Festlegen der Begrifflichkeit und der Werkzeuge (“Topic Types”, “Association Types”)
- Festlegen der Mitglieder (“User”, “Membership”)
- Festlegen von Rollen und Rechten (“Type Access”, “Membership”-Properties)
- Anbieten von Standard-Kommunikationswerkzeugen (“Forum”, “Chat”)

Erst nachdem eine Topic Map in einen Shared Workspace publiziert wird (“Publish”-Kommando) sind die Inhalte potentiell für andere Benutzer sichtbar.

Wenn der Benutzer eine Topic Map aus einem Shared Workspace öffnet, erzeugt DeepaMehta eine Kopie der Topic Map und legt diese im Personal Workspace des Benutzers ab. Geöffnete Topic Maps sind also stets personalisiert.

## Stand der Entwicklung

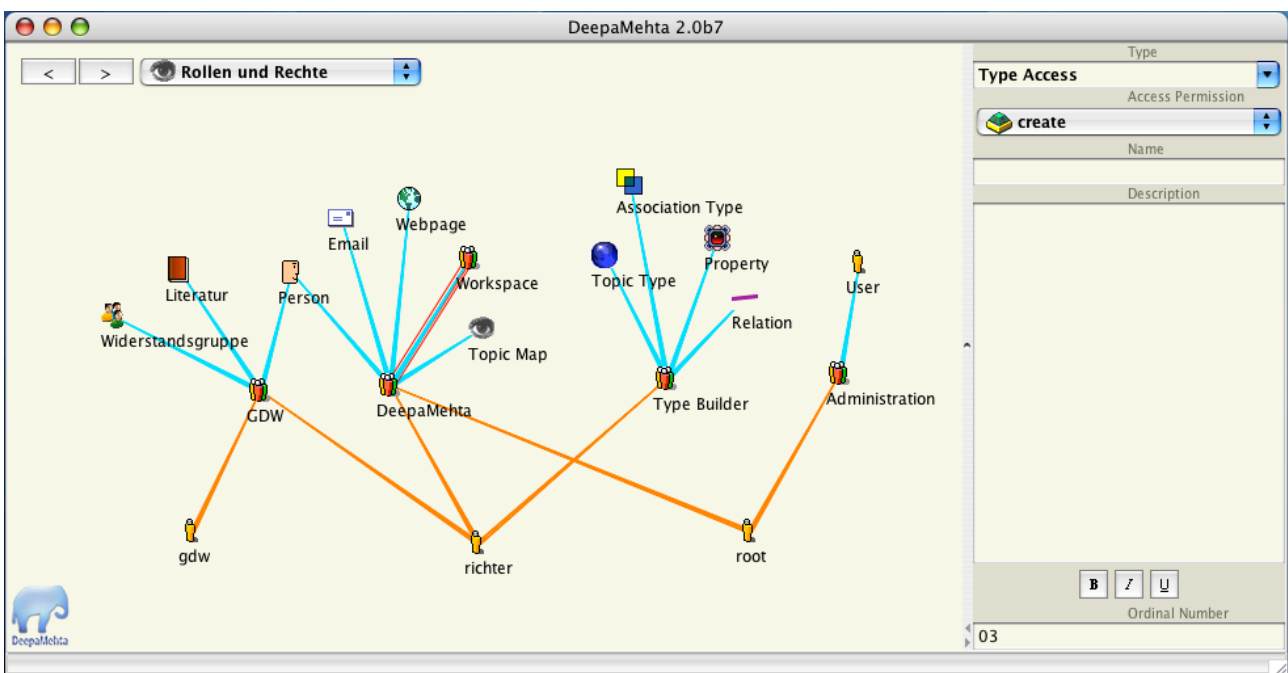
### Personalisierung von Topic Maps

Beim Kopieren einer Topic Map (Personalisierung) werden tatsächlich nicht die Inhalte kopiert (die Werte der Properties), sondern nur die Sicht, d.h. Geometrie- und Sichtbarkeitsinformationen der Topics und Assoziationen. Geänderte (Property-)Inhalte ...

- werden sofort in das Corporate Memory zurückgeschrieben und überschreiben vorhandene Inhalte: die vorherige Fassung geht verloren.
- sind unabhängig vom Freigabestatus (Publishing) potentiell für andere Benutzer sichtbar: Ignorierung der Privatsphäre.

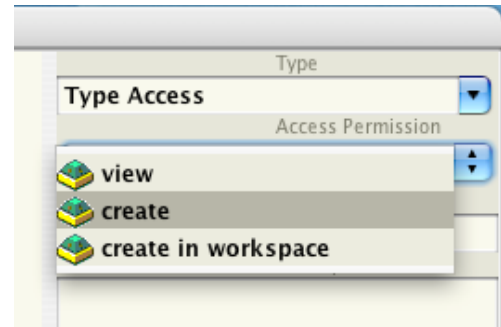
### Rollen- und Rechte-System

Auf welche Topic-Typen und damit auf welche Topic-Instanzen ein Benutzer Zugriff hat, hängt im Wesentlichen von seinen Workspace-Mitgliedschaften ab: ein Benutzer hat Zugriff auf alle Topic-Instanzen der Typen, die den Workspaces zugeordnet sind (mittels “Type Access”-Assoziationen), in denen er Mitglied ist. Die Zugriffsrechte sind also



grundlegend auf Typ-Ebene definiert, nicht auf Instanz-Ebene. Tatsächlich sind einzelne Topic-Instanzen nicht explizit einem oder mehreren Workspaces zugeordnet, sondern nur implizit über ihre Typen: Wenn ein Benutzer z.B. Zugriff auf den Topic Typ "Person" hat, kann er (Workspace-übergreifend) auf *alle* Personen zugreifen, die im Corporate Memory gespeichert sind. Derzeit ist eine Hilfsmethode, (Pseudo-)Topic-Subtypen zu definieren, was semantisch gesehen aber nicht Sinn der Sache ist.

Die Art des Zugriffsrechts — "View" oder "Create" — hängt von der Art der Typ-Workspace-Zuordnungen ab ("Access Permission"-Property der "Type Access"-Assoziationen). Die View- und Create-Rechte sind also grundlegend auf Workspace-Ebene definiert, und nicht auf Benutzer-Ebene: Wenn z.B. in einem bestimmten Workspace ein Create-Recht für den Topic Typ "Person" definiert ist, können *alle* Mitglieder dieses Workspaces Personen-Topics anlegen. Oder, wenn z.B. ein Benutzer das View-Recht für Topic Maps und für Emails hat, kann er auch die privaten Topic Maps und Emails der anderen Benutzer einsehen. Derzeit ist eine Hilfsmethode, (Pseudo-)Workspaces anzulegen, was semantisch gesehen aber nicht Sinn der Sache ist.



Das View-Recht ist gegenwärtig nicht vollständig realisiert: Wenn ein Benutzer das View-Recht für einen bestimmten Typ hat, heißt das gegenwärtig nur, daß der Typ im "Search"-Untermenü auftaucht, der Benutzer also nach Topic-Instanzen dieses Typs im Corporate Memory suchen kann. Über "What's related?" hingegen, kann der Benutzer Topic-Instanzen von Typen aufdecken, zu denen der Benutzer gar kein View-Recht hat.

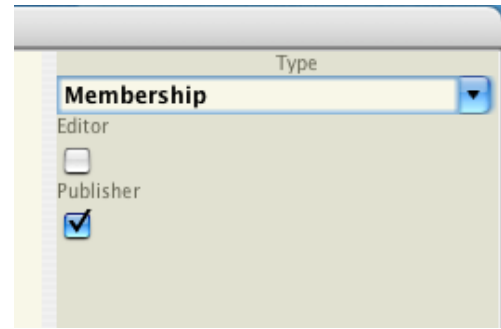
Für die weiteren Zugriffsrechte — "Edit", "Delete" und "Retype" — gibt es das Konzept des Besitzers (*Owner*). Edit-, Delete- und Retype-Operationen auf einem Topic/einer Assoziation sind exklusiv nur dem Topic/Assoziation-Besitzer erlaubt. Der Topic/Assoziation-Besitzer ist derjenige Benutzer, der den Topic/die Assoziation angelegt hat. Es ist also tatsächlich zunächst nicht möglich, gemeinsam an den Inhalten eines Topics Änderungen vorzunehmen. Um diesen Mangel an Kollaborationsfunktionalität zu relativieren, wurden in DeepaMehta 2.0b8 (noch nicht released) behelfsmäßig zwei Rollen eingeführt.

Bevor die zwei neuen Rollen beschrieben werden, muß ein weiteres Zugriffsrecht — "Publish" — erklärt werden. Durch Publizieren werden persönliche Topic Maps in Shared Workspaces bewegt. Die bisherige Publishing-Politik ist wie folgt:

- jedes Workspace-Mitglied kann Topic Maps in diesen Workspace publizieren.
- eine neu angelegte Topic Map kann vom Benutzer in jeden Workspace publiziert werden, in dem er Mitglied ist.
- eine bereits publizierte Topic Map kann nur in den Workspace publiziert (zurückgelegt) werden, in den sie ursprünglich publiziert wurde.

Die beiden seit DeepaMehta 2.0b8 eingeführten Rollen — "Editor" und "Publisher" — sind an die User-Workspace-Zuordnungen geknüpft (Ankreuzfelder bei "Membership"-Assoziationen), können also auf Basis einzelner Benutzer vergeben werden, und haben folgende Bedeutung: ein Benutzer mit gesetzter Editor-Rolle für einen bestimmten Workspace darf *alle* Topic-Instanzen *aller* Typen ändern, die dem Workspace zugeordnet

sind. Ein Benutzer mit gesetzter Publisher-Rolle für einen bestimmten Workspace darf Topic Maps in diesen Workspace publizieren. Während die Editor-Rolle zur Ausweitung des standardmäßigen Edit-Rechts dient, kann mit der Publisher-Rolle das standardmäßige Publish-Recht eingeschränkt werden. Diese beiden Rollen werten DeepaMehtas Kollaborationsfunktionalität wesentlich auf, daß aber dies nur eine ad-hoc Lösung ist, die keine feingranularen Rechtedefinitionen ermöglicht, liegt auf der Hand.



DeepaMehta-Administratoren haben implizit alle Rechte. Administratoren sind Benutzer, die Mitglied des "Administration"-Workspaces sind.

## Notwendige Arbeiten

### Erweiterung der Kollaborationsfunktionen

(aus: "DeepaMehta — Another Computer is Possible")

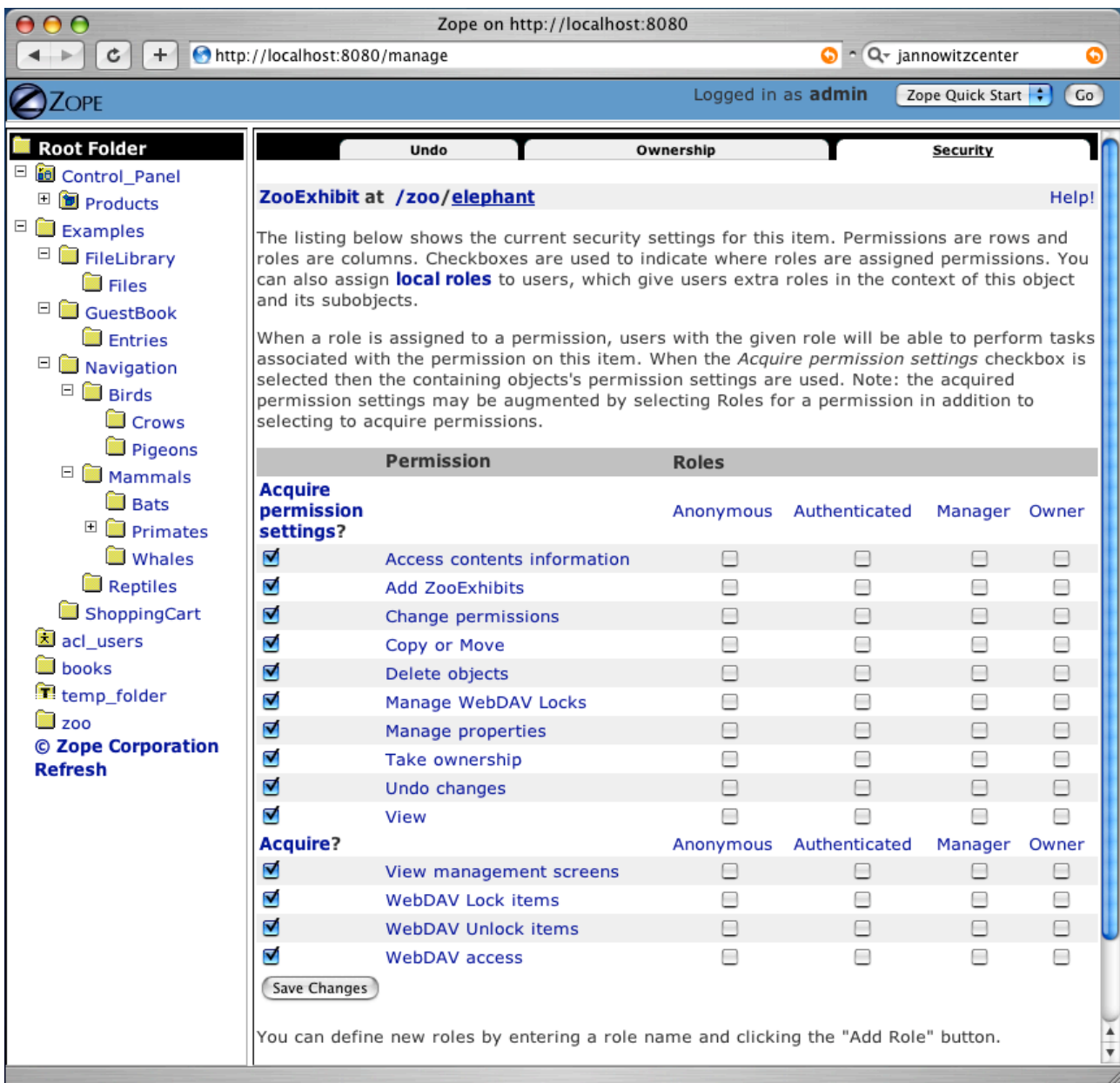
Essential features of every collaboration environment are a) versioned data storage, b) a change history, c) notification, and d) access control.

- Versioning is a good method to cope with competing change requests without establishing a locking mechanism. Currently DeepaMehta contents are not versioned. Future versions of DeepaMehta will support versioning at 4 levels: topic contents, association contents, topic map geometry/visibility, and file contents.
- A change history helps to keep track of content changes performed by users: what was changed by whom and when? Changes can be presented visually like in Microsoft Word's Change-View or in Wikis. Certain changes can be reverted. Currently DeepaMehta provides no change history.
- A notification mechanism informs users of relevant actions performed by other users: what has been done by my colleagues since my last login? A crucial concern is granularity: who should be notified and in what detail? Currently DeepaMehta's notification mechanism is quite limited: members of shared workspaces are informed via messaging and email once a topic map is updated but not about what has been changed.
- An access control mechanism is crucial to protect confidential information and privacy and to enforce user hierarchies. Currently DeepaMehta provides an access control mechanism but it is not flexible enough. Basically it works on type-level and has the concept of a topic-owner. Future versions of DeepaMehta will provide access control at instance-level and even property-level.

### Neukonzeption des Rollen- und Rechte-Systems

Vorbild für das DeepaMehta Rollen- und Rechte-System könnte z.B. die Zope-Plattform sein: zu jedem Objekt (Topic, Assoziation) gibt es eine Anzahl von möglichen Operationen (*Permissions*) sowie eine Zuordnung von Benutzer-Rollen, deren Inhaber die jeweiligen Operationen ausführen können. Standard-Operationen wären z.B. "View", "Create", "Edit", "Delete". Aufgrund des (Topic-)Typs könnten weitere Operationen zur Verfügung stehen,

so würde z.B. eine Topic Map eine "Publish" Operation mitbringen, und ein Workspace "Join" und "Leave" Operationen. Standard-Rollen könnten z.B. "Anonymous", "Authenticated", "Manager" und "Owner" sein. Spezifische Anwendungen könnten weitere Rollen mitbringen, so könnte z.B. die Anwendung "Kompetenzstern" eine "Template Builder"-Rolle definieren.



## Neukonzeption der GUI

Hinsichtlich der Workspaces ist ein Problem der derzeitigen GUI-Konzeption, daß die Shared Workspaces und der Personal Workspace zu stark getrennt sind. Daß mit jedem Öffnen einer Topic Map aus einem Shared Workspace eine Topic Map-Kopie im Personal Workspace abgelegt wird, ist nicht intuitiv. Zwei wesentliche Probleme sind die Folge:

- Der Personal Workspace wird mit immer neuen Topic Map-Kopien überladen, ohne daß der Benutzer das beabsichtigt hat.
- Im Personal Workspace sammeln sich Kopien aus *verschiedenen* Workspaces, was sehr unübersichtlich werden kann.