


CommunityMirror

Die Aufgabe von CommunityMirror-Anwendung ist die Visualisierung von Informationsobjekten (welche mit dem CommunityMashup gesammelt / bereitgestellt werden) sowie die Interaktion mit diesen Visualisierungen ist.

Siehe hierzu auch [CommunityMirror - Grundarchitektur und Wording](#)

 Suche in diesem Bereich

Basis der Implementierung von CommunityMirror-Anwendungen ist das CommunityMirrorFramework.

Das CommunityMirrorFramework

- stellt Basisbibliotheken/Bausteine zur Realisierung solcher Views zur Verfügung - sowohl für Interaktionselemente (z.B. Suche), Visualisierungselemente (z.B. Detailsichten) als auch Datenzugriff (High Level Mashup API)
- stellt einen Rahmen zur Verfügung, in den Views eingesetzt werden können und der das Navigieren zwischen Views umsetzt ("View Manager" / "Window Manager")
- Grundkonzept ist Multi-User, Multi-Touch, Multi-Screen und ggf. später Multi-Device

Fokussierung auf

- Daten vom CommunityMashup
- Nicht-Maus UIs - also Touch, Gesten, ...
- Multiuser-Anwendungen / Multiuser-UIs - d.h. größer Wandbildschirme, Tabletops, evtl. Kiosk-Systeme
- halböffentliche Anwendungsszenarien
- Ein-Rechner-Anwendungen

CommunityMirrorFramework besteht aus / muss beinhalten

- View Manager - Immer ein View angezeigt oder Views in Arbeitsbereichen? Wechsel durch Bedienelemente oder Gesten?
- Bibliotheken mit Visualisierungs-/Interaktionskomponenten
- Bibliotheken mit (High-Level-)Zugriffsfunktionen
- Bibliotheken / Elemente zur Kommunikation mit anderen Devices
- High-Level-Event-Queue (Touch, Gesten, ...)

Für eine Implementierung des CMF muss festgelegt werden:

- Programmiersprache
- Basis Animations / Grafik-Technologie
- Basis oder ergänzende Event-Technologie

Es ist möglich, mehrere Umsetzungen mit unterschiedlichen Optionen zu realisieren.

Erste geplante Umsetzung des CommunityMirrorFrameworks

- Betriebssystem: Windows 7+
- Programmiersprache: Java(FX)
- Grafik/Animation: JavaFX
- Elemente soweit möglich als Web-Views, evtl. auch ganze Views als Web-Views
- Eventhandling: TUIO-basierend, optionales Plugin von Gestureworks

Nächste Schritte:

- Architekturworkshop mit Realisierung Rahmenanwendung und Test auf unterschiedlichen Devices
- Interative Weiterentwicklung

Beispielanwendung (zum Abklären der Anforderungen)

- MeetingMirror

Hier noch ein paar Textbausteine aus einer Vorversion zu "Band 1" - Können vielleicht noch gebraucht werden ...

Grundkonzept

Die Lösung geht davon aus, dass Daten beschafft und kombiniert worden sind (mit dem CommunityMashup).

In einem zweiten Schritt gilt es nun, eine geeignete Repräsentationsform der Daten zu finden, um sinnvoll strukturierte Informationseinheiten bereitzustellen zu können, welche zum einen für den Menschen natürlich verständliche Einheiten darstellen, also die realen Informationsobjekte widerspiegeln (und zum anderen intern eine sinnvolle Integration und Traversalion der Daten bei Suchanfragen ermöglichen.)

Unter einem CommunityMirror verstehen wir eine Lösung, die genau das macht – einer Community Daten vorstellen / spiegeln, welche den Mitgliedern der Community von Nutzen sein können.

Zur Realisierung von CommunityMirrors sollten unterschiedliche Geräte eingesetzt werden können – große Wandbildschirme, Tablets, Kiosk-Systeme und auch Smartphones.

Zur einfachen Realisierung von kontextspezifischen CommunityMirror-Lösungen haben wir das sog. „CommunityMirror Framework (CMF)“ implementiert. Hierbei handelt es sich um einen modularen Baukasten, der es schnell und einfach ermöglicht, verschiedene konkrete CommunityMirror-Anwendungen für verschiedene Einsatzszenarien kontextspezifisch bereitzustellen. Abgeleitet aus den oben beschriebenen Potenzialen und bereits existierenden Ansätzen konnten wir verschiedene Einsatzszenarien innerhalb kooperativer Wissensprozesse identifizieren und in realen Settings evaluieren:

- Visualisierung von Sozialen Netzwerken und virtuellen Teams („MeetingMirror“)
- Ubiquitäre Präsentation von Projekt- und Forschungsergebnissen („ExhibitMirror“)
- Crowdsourcing der Informationsbewertung, z.B. bei Innovationsmanagementsystemen („IdeaMirror“)
- Vermittlung von Awareness über kooperative virtuelle Zusammenarbeit, z.B. in Wikis („CollabMirror“)

Technischer Aufbau

Im Gegensatz zu existierenden Ansätzen zur Awareness-Unterstützung, die meist als eigenständige Anwendungen konzipiert sind und für das Sammeln, Speichern, Aufbereiten sowie Anzeigen von Information verantwortlich sind, basieren CommunityMirrors auf einem leichtgewichtigen Konzept, auf Basis dessen sie ohne eigene Datenhaltung einfach und schnell an existierende Anwendungssysteme im Unternehmenskontext angedockt werden können. Zur Andockung wird auf das CommunityMashup zurückgegriffen. Hierdurch können Informationsobjekte aus verschiedenen Datenquellen integriert und in für die Anzeige geeignete Strukturen konvertiert werden, ohne redundant in eigenen Datenbanken vorgehalten werden zu müssen. Grundsätzlich besteht CommunityMirror als Informationsstrahler damit jenseits der Datenquellen aus zwei Hauptkomponenten:

- Den generischen Framework-Bestandteilen, die zur Strukturierung der abstrakten Informationsobjekte aus den verschiedenen Datenquellen dienen (Objekte, Strukturen) sowie Visualisierungs- und Interaktionsmöglichkeiten für diese bereitstellen (Views, Touchscreen-Komponenten).
- Den konkreten Anwendungsszenarien, die den sozialen Kontext der Anwendung sowie den gewünschten Einsatzzweck determinieren

Basierend auf einem konkreten sozialen Kontext und ausgesuchten Datenquellen erfolgt der Datentransformationsprozess in einer CommunityMirror-Anwendung wie in folgender Abbildung dargestellt. Die Daten der verschiedenen im Unternehmen vorhandenen Quellen werden importiert und in Form greifbarer Informationsobjekte, wie beispielsweise „Inhalte“, „Personen“, „Projekte“ sowie ihrer Beziehungen untereinander aufbereitet. Anschließend werden die Informationsobjekte mit kontextspezifischen und intuitiv verständlichen optischen Informationsrepräsentationsformen ausgestattet, z.B. einer digitalen Visitenkarte mit Name und Foto einer Person, sowie mit Interaktionsmöglichkeiten versehen, z. B. das Ermöglichen einer Detailansicht beim Klick auf eine derartige Visitenkarte.

Das View-Konzept

Für unterschiedliche Einsatzszenarien sind grundsätzlich unterschiedliche optische Aufbereitungen für die darzustellenden Informationsobjekte erforderlich. Die Visualisierung von Sozialen Netzwerken erfordert beispielsweise gänzlich andere Darstellungen als die Anzeige von Ideen aus einem Innovationsmanagementsystem. Weiterhin lassen sich unterschiedliche Nutzer aufgrund persönlicher Faktoren, wie beispielsweise ihrer Technikaffinität oder der Stimmung zum Zeitpunkt der Interaktion, durch verschiedene Informationsrepräsentationsformen ansprechen. Auch die soziotechnischen Rahmenbedingungen, wie z.B. der Aufstellungsort, müssen berücksichtigt werden. In einem Gemeinschaftsbereich, wo Personen direkt vor dem Bildschirm stehen, sind auch kleinere Inhalte problemlos zu erkennen, wohingegen in einer Cafeteria aufgrund der größeren Entfernung und des anderen Aufmerksamkeitsfokus von potenziellen Nutzern, andere optische Reize und Darstellungsformen erforderlich sind. Im Gegenzug können jedoch Synergien entstehen, wenn verschiedene Anzeige-Elemente in einem sinnvollen Einsatzkontext kombiniert werden (z. B. zur Anzeige, wie Ideengeber in einem Innovationsmanagementsystem untereinander vernetzt sind). Aus diesem Grund werden Visualisierungen im CMF generisch in Form sog. „Views“ zur Verfügung gestellt.

Die grafischen Grundlagen dieser abstrakten Views wurden in Zusammenarbeit mit Kommunikationsdesignern entwickelt, um ausreichende Attraktivität zur Interaktionsmotivation zu gewährleisten. Sie sind ausschließlich für die Visualisierung sowie für das Bereitstellen von Interaktionsmöglichkeiten für die dargestellten Informationsobjekte zuständig und so generisch gestaltet, dass sie jeweils für verschiedene Klassen von Informationsobjekten (z. B. Personen, Inhalte, etc.) universell eingesetzt werden können. Das View-Konzept bildet damit die Grundlage für die Flexibilität des CMF, da Views und Datenquellen unabhängig voneinander für kontextspezifische Einsatzszenarien erweitert und kombiniert werden können. Durch das CMF wird sichergestellt, dass jeder View für alle Informationsrepräsentationsformen jedes internen Informationsobjekts zur Darstellung verwendet werden kann. In einer Rotationsansicht können deshalb neben den ursprünglich vorgesehenen Inhalts-Karteikarten auch Visitenkarten von Personen angezeigt werden. Genauso erlaubt die Social Networking Visualisierung auch die Anzeigen von Inhaltsbeziehungen (z. B. auf Basis identischer Tags).