

Fr
19:30

Zielsetzungen Ubitrack

- Ideen für noch bestehende Fragen
- Gute Diplomarbeiten
- Langfristig: eine zentrale Frage in AR
- Allererste Lösung mancher Teilprobleme
- Letzte Implementierungen vor Dis-Abgabe
- Theorie darauf aufbauend weiterdenken
- Neue Probleme identifizieren
- Funktionierendes System/Demo bauen
- Einzelösungen, Teilalgorithmen, z.B. Autokalibration
- Implementierung, die wiederverwendet wird
- Ubitrack als Anwendung v. DWARF-Konzepten
- Verbesserung der Middleware
- Basis, die weiterentwickelt wird
- Erfolgserlebnis

Koordination / Planung

Fr ①

- Zeitplan
- Regeln (Protokoll etc.)
- Allg. Auf-den-neuesten-Stand-Bringen
 - Dieters neue Stelle in Graz
 - Wien: Joe koordiniert Antrag (EU)
 - Informelles EU-Feedback zu FET-Vorantrag
(nicht embryonisch genug, * wirtschaftl. Erfolgsaussichten)
 - Nächster Schritt: 5-Seiten-Vorantrag (Martin W + Joe)
 - Leitung: Joe, TU Graz/Wien
 - Industriepartner gesucht...
 - München: 2 neue externe Doktoranden bei BMW
(Verena Brany, Stephan Huber)
 - BMW-Montageplanungs-Szenario
 - Antrittsvortrags-Nachfragen
zeitverschiebungen, Gleichbewegungserkennung
 - Paper-Hinweise (Robotik 90er)

Koordination (2)

- Pivy Bawml Viewer Simulator Zeug

Martin W: Installationsanleitung \Rightarrow Martin W \Rightarrow RPMs

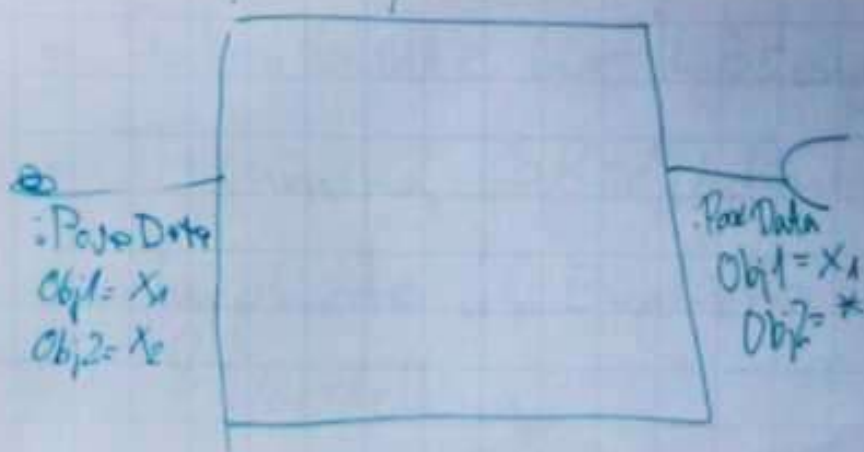
Asa: Otmar sagen, dass CVS-Soft auf alle Rechner soll
(von #at8, et 14)

- Supportmöglichkeiten aus Wien
 - Tamer: Pivy, Coin, BSD/Linux, etc.
 - Silk-Lente: Karin Kosina, Marius... : Coin
- Simulatoreinbau in DWARF: Virtuelle Szenen
 - Allg. Mein-Stand-ist - Runder

DA-Präsentationen - Diskussion

2a 3-11

- Mapping SR-Graph \leftrightarrow Kommunikationsstruktur
- Physische Netzwerk-Charakteristika
- Caching: Supernodes, Locales
- (Hardware) IDs für Identifikation
- Innerer Zusammenhang der Themen
- Verteilte Suche: Gerichtete Suche / Breitensuche / ??
→ Mapping auf DWARD?



- Auswertung v. Attributen, v.a. Stückweise
- Unterschied Kalman / Least Squares
- Allg. Anwendbarkeit v. Kalman Filtern
- Lernen / manuelles Tunen v. Parametern
- Welche Objekte autom. kalibrieren?

Diskussion Vorträge(2)

Sa 11h-12h

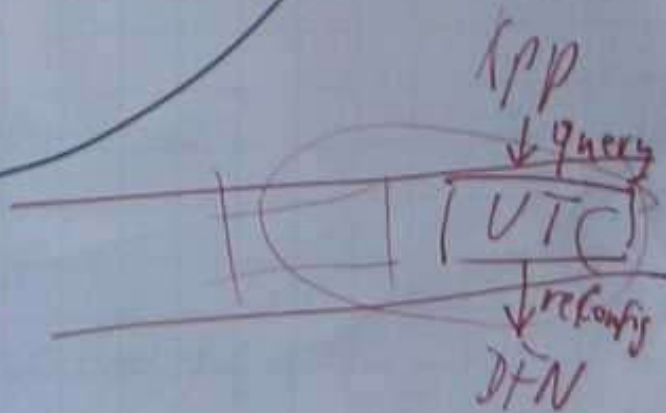
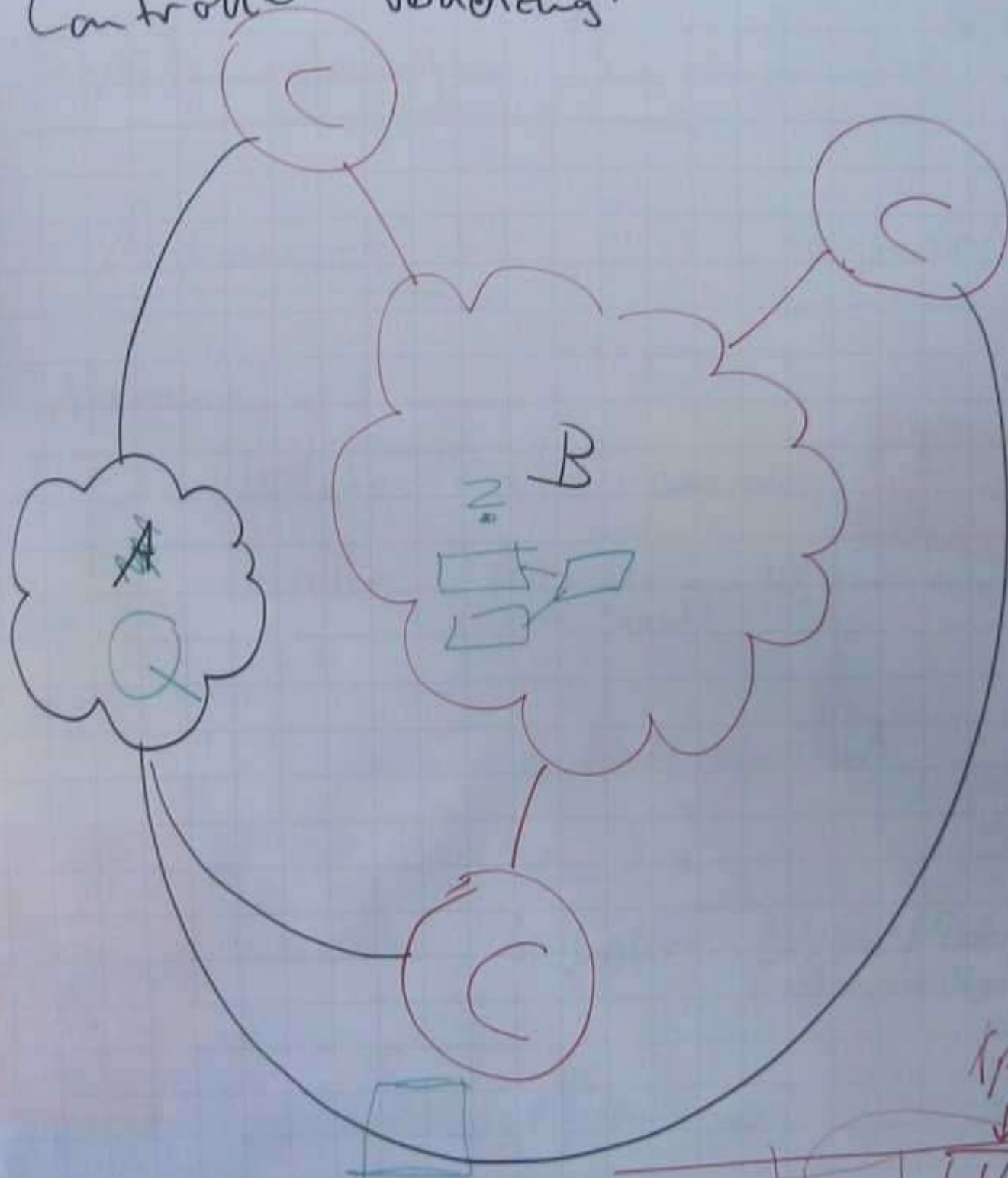
- Begriffe Calibration/Registration/Pose Estimation
- DWARF: Verteilte Datenhaltung
- Effizienz in DWARF-Middleware
- ARToolkit "t" vs. DWARF Services?
- Wie funktioniert Suchalgorithmus
- Integration München/Wien
- Rolle des Ubitrack Controllers
- Schnittstelle für "Lösungen": Pfade?
- Trennung SR-/Datenflußgraph?
- Implizite vs. Explizite Algorithmen & Darstellung
- Suchoptimierung/-abbruch mit Supernodes
- Einmalabfragen/Streams
- Rolle von OpenTracker

System design / Diskussion

- "Wir kennen die Welt"
vs.
"Wir kennen die Welt nicht"
- "Top-down vs. bottom-up"
- Phasen: Suche ~~und~~ Verschaltung
gleichzeitig oder nicht?
- Äquivalenz Ubitrack Controller
OWARF-Service Manager?

Controller-Vernetzung?

Sa 14h



Gemeinsame APIs

- Lösungspfade: eher schwer
- Anfragen: sollte möglich sein
"Query" → Konsens
- Eigentlicher Trackerdatenfluß:
Adapter einfach
"Sensor" API
- Datenfluß-Rekonfigurations-
Schnittstelle?
→ Für Service Manager & OpenTracker
- OpenTracker als gemeins. Datenfluß-Schicht

Systemdesign II

Anfrage-API (wg. "Erfolgsresultat")

Parameter:

Update Rate

Update-Rate

Quelle

Quelle } Mehr als nur ID, evtl. Wildcards,
Ziel } evtl "nicht"

Ziel

Δt

Δt (Vergangenheit-Zukunft, ~~Time~~ Offset)

Error Func.

Error Function !

Synchron/Async

Synchron / Asynchron } "Modus" / "Protocol"
Push / Pull } \rightarrow versch. Signature?

Pull

(Zeit Absolut) \leftrightarrow Δt ?

QoS-Anforderung

Resource Discovery?

Query "preparation"

Spatial Granularity (z.B. Squares)

(APIS...) Anfrage...

• Fehlerfunktion

- Code übertrage?

- In XML spezifizieren?

- Versch. vorgegebene Strategien?

- Soll alles von Anwendung vorgegeben werden?

- Zweck: Lösungskandidaten gegeneinander vergleichen

- Fehlerfunktion: $e: (\text{KantenAttribute})^* \rightarrow \mathbb{R}^+$

(Einschr.) - Grundfunktionalität kombinieren
↳ interpretierbare Beschreibung

- Einschränkung auf Handhabbare Form

- Offen gegenüber neuen Attributen

(Einschr.) - Einschränkung $(\text{Kantenattribute})^* \rightarrow \mathbb{R}^+$

- Abbruchkriterien ↳ Diskussion

Fehlerfunktion, nochmal

So 9h

- Beschreibung als Kombination von Grundfunktionen (z.B. in XML, z.B. als Formel)
- Basierend auf Attributen
- Zweck: Vergleich v. Lösungskandidaten
- Frage: $e: A \rightarrow \mathbb{R}^+$
oder $e: A^* \rightarrow \mathbb{R}^+$?

Fakten[✓] / Positiv / Negativ / Ideen / Gefühl

Relevanz Frage A oder A^* ?

- || - Genau das Gleiche, wg Mechanismus $A^* \rightarrow A$
- Detail im Gesamtzusammenhang \rightarrow weitere Bearbeitung
- || - Eigentl. egal
- Beeinflusst Design u. Beschreibungssprache
- Ziemlich wichtig, aber Sensoren wichtiger
- Konsistentes Denkmodell
- Wichtig; evtl. Probleme Mapping $A^* \rightarrow A$?
- Auswertung der Attribute selbst wichtiger.
- Beeinflusst Trennung Anwendung / Implementierung

	A	A^*
⊕	"KISS" - Einfach Mathematisch rigide Einfache Erweiterbarkeit	Detaillierte Kontrolle Keine Vorschriften für kat. nötig (Mapping $A^* \rightarrow A$) Man muß nicht alle k. betrachten
⊖	Eingeschränkt Verschenkung v. Optimierung Einzelparameter schwieriger	<u>Geringfügig</u> konkretere Spezifika Deutlich " " Nicht mächtiger Sicht auf Gesamtsystem schwieriger Anwendungsseite $A^* \rightarrow A$ wissen Detaillierte Kontrolle unnötig

Ideen: A oder A*

- Trennung in API:
 - Fehlerfunktion $A \rightarrow \mathbb{R}$
 - Mapping $A^* \rightarrow A$, mit Default-Verhalten
- Layered
Doppel-API: A oder A* angeben
 - A-Funktionen, vordefinierte A*-Makros
- "Bausteine" v. Komponenten in DFG
 - Anwendungsunabhängige Beschreibung der Auswirkung dieser Komponenten auf die Attribute
 - Auch Mapping $A^* \rightarrow A$ in Komponente kapseln
- Welche Fehlerfunktionen wollen wir überhaupt?
 - vorher klären, dann API festlegen
- Mächtige API-Möglichkeiten zunächst nicht implementieren
- Zuerst mit einfacherer API anfangen
- Zuerst: "Welche Attribute wollen wir überhaupt?"
- Extreme Vereinfachung: $enum\{schnell, gut\}$

"Gefühl" darüber

- Interessante Frage, aber später klären
- "Fuzzy" API gut: "bisschen schnell"
- Akademische Frage... eher Detail
- Konflikt entschärft - zunächst einfach, dann mächtig
- Aufpassen, dass erweiterbar
- Gut, klein anzufangen
- Aufpassen, dass Framework-Funktionalität nicht in Anwendung gerät.
- Erst prinzipielle Frage "A gleich mächtig" als wahr annehmen, klein anfangen

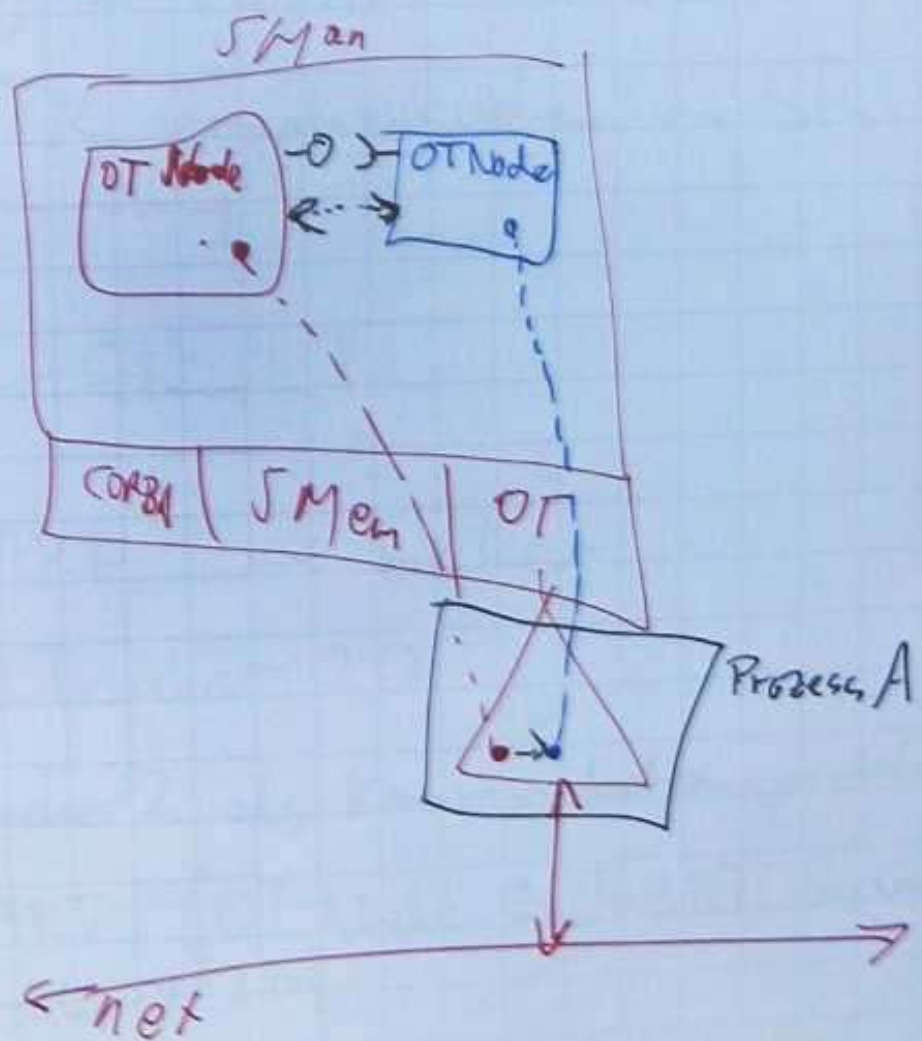
⇒ Klausur: String: "gut" / "besser"
später XML usw.

Martin W: Spezifiziert erstmal Sensor- und Query-API

Koordination Wien-Münde

So 11h

- Welche SW-Module sollten implementiert werden, und wer macht das?
- Welche Schnittstellen gibt es zwischen den Modulen, und wer spezifiziert sie?
- EU-Antrag (kurz)



Koordination - Ergebnisse

- Schichtung Anwendung / Ubitrack / Sensoren

- Ubitrack-Schicht:

TUM - S/R-Graphensuche ^{+DFG Aufba} (implizit, über Attribute) in DWARF

Thomas P - Oder: S/R Graphensuche im SR ver ("Thomas-only")

- Sensoren-Schicht:

TUM - In DWARF, wie bisher

TU Wien In OpenTracker 2.0

TUM - OpenTracker 2 als Kommunikationsprotokoll
+TU Wien in DWARF (OT-Node $\hat{=}$ DWARF Service)

Thomas P OpenTracker 2 API

Martin W • Query API: Need-Typen + Attribute
(unabh. von Protokollen)

Martin W • Sensor API: Abstr. Daten, die von Sensor kommen
(Attribute kommen später)

Was fehlt noch?

- Einfache, jedermann Verständliche Motivation, warum man das ganze braucht
 - Applikation, die sonst nicht geht?
 - Ideen rumschicken! Vorstellen: perfektes Tracking Krankenhaus, Notfall, ...
- Modellierung unserer Annahmen, um gewisse Lösungsmöglichkeiten zu erlauben
- Demoanwendungen, Demo-Hardware-Setups
- Verschiedene Fehlerfunktionen
- "Welche Objekte sind in der Nähe?"
 - ↳ Spatial Index?
- Test Cases
 - z.B. Kreisrunde Kanovas
 - Ideen sammeln ins Web!