

Spezifikation, Ausführung und Monitoring von Workflows in verteilten Wissensgraphen (Abstract)

Tobias Käfer und Andreas Harth, International Semantic Web Conference 2018

Moderne Komponenten liegen verteilt mit Wissensgraph-basierten Schnittstellen vor

- Wie integrierte Anwendungen erstellen? Mittels Workflows [1]?

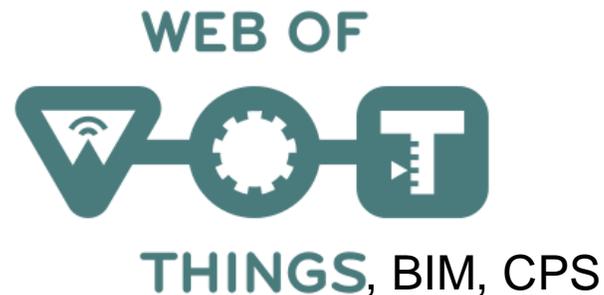
→ Suche nach der Kombination von:

- Workflows
- Semantische Datenverarbeitung/-integration
- REST-basierte Interaktion

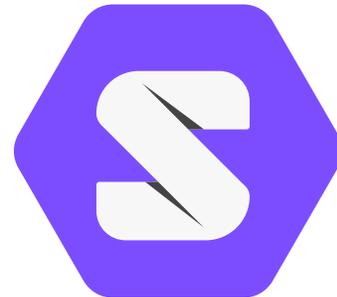


Umgebung mit Uniformer Schnittstelle *Read-Write Linked Data* (verteilte Wissensgraphen)

- REST als uniformes Interaktionsparadigma zwischen Systemen
- RDF als uniformes Datenmodell, Basis für semantische Integration



- Schnittstelle zu Sensoren und Aktuatoren
- Basiert auf Linked Data



- Schnittstelle zu persönlichen Datenspeichern
- Basiert auf Linked Data



- Schnittstelle zu Geschäftsfunktionen
- Basiert auf REST (Daten auf RDF hebbar)

Sam Newman:
Building
Microservices.
Sebastopol (CA),
USA: O'Reilly, 2015.

[1] Jablonski und Bussler: Workflow Management. London: International Thompson (1996)

“[...] confusion remains on how to properly use the web to build loosely coupled integrated systems.”

– Pautasso and Zimmermann: “The Web as a Software Connector”, IEEE Software 35:1 (2018)

...und wie war das jetzt mit dem Flugzeug?

Flight Simulator

- Projekt [1] zu ergonomischen Studien von Flugzeugcockpits in VR
- Read-Write-Linked-Data-Schnittstellen zu heterogenen Datenquellen:

- Flugsimulator
- VR-Engine
- Physischen Schaltern und Reglern
- Finger- und Head-Tracking
- ...

- Workflows
- Semantische Datenverarbeitung/-integration
- REST-basierte Interaktion

- Integration Produktbeschreibung – VR-Elemente
- Standardarbeitsanweisungen (SOPs) umgewandelt in Workflows
- Beispielanfrage für ergonomische Studien:

- *“SELECT the push-buttons in the Virtual Reality that are involved in the upcoming steps of the currently running take-off workflow”*

Release brakes

Accelerate

Rotate

→ Genug Beschreibungen, um das Flugzeug zu steuern?

Excerpt



[1] Käfer, Harth, Mamessier. The i-VISION case. Cpsdata@CPSweek 2016; <http://www.ivision-project.eu/>

Verwandte Arbeiten: Workflow-Sprachen + deren System-Umgebungen



	Ansatz	Bedeutung von Kästen	Bedeutung von Pfeilen	
BPMS	BPEL u.a. (Web Services)	Funktionsaufrufe (liefern Ergebnisdaten)	Abhängigkeiten und Parameter	Semantic-web-affin
	BPEL u.a. (Human Tasks)	Aufgaben (liefern Ergebnisdaten)	Abhängigkeiten und Parameter	
Workstation	Pegasus u.a. (Kommandozeilenprogramme)	Programme (terminieren)	Abhängigkeiten	
Cluster	Taverna u.a. (Datenstromverarbeitung)	Stromoperatoren (terminieren nicht)	Datenströme	
	WiLD (Read-Write Linked Data)	?	?	

Cf. Elmroth et al.: Three fundamental dimensions of scientific workflow interoperability – Model of computation, language, and execution environment. *Future Generation Computer Systems* 26 (2010)

ASM4LD – ein Rechnermodell für Read-Write Linked Data [1]

- Abstract State Machines
- Modelltheoretische Semantik von RDF-Graphen
- Semantik von RDF-Datasets
- Semantik von HTTP

ASM4LD [1]

- Unterstützt Read-Write Linked Data:
 - URIs als Identifikator für Dinge
 - HTTP als Interaktionsprotokoll
 - RDF zur Beschreibung von Daten
 - (mit Schreibmöglichkeit)
- Reasoning und Hypermedia eingebaut
- Regelbasiert (Notation3) →
- Turing-vollständig
- Engine: <http://linked-data-fu.github.io/>

1. Assertions

```
ldpc:wfi1 a wild:WorkflowInstance .
```

2. Deductions

```
{ ?x a wild:WorkflowInstance }  
=>  
{ ?x a wild:Instance } .
```

3. HTTP Interactions: reading

```
{ ?x a wild:WorkflowInstance }  
=>  
{ [] a http:Request ;  
  http:mthd httpM:GET ;  
  http:requestURI ?x . } .
```

4. HTTP Interactions: writing

```
{ ?x a wild:WorkflowInstance ;  
  wild:instanceOf ?y ;  
  wild:hasState wild:uninitialised }  
=>  
{ [] a http:Request ;  
  http:mthd httpM:PUT ;  
  http:requestURI ?x . ;  
  http:body  
  { ?x a wild:WorkflowInstance ;  
    wild:instanceOf ?y ;  
    wild:hasState wild:initialised } } .
```

[1] Käfer and Harth: Rule-based programming of user agents for Linked Data. Proc. 11th LDOW@TheWebConf (2018)

[2] Gurevich: Evolving algebras 1993: Lipari guide. Specification and validation methods 1993: 9-36

Workflow-Management in der Umgebung Read-Write Linked Data

- Ziel: Workflow-Sprache + operationale Semantik
 - Unterstützung der Basic Workflow Patterns [1]
 - Herausforderungen der (ungewöhnlichen) Umgebung:
 - Closed vs. Open World Assumption (Reasoning in RDF-basierten Sprachen)
 - Ereignis vs. Zustandsdaten (REST-basierte Interaktion)
- Bei der Entwicklung von Sprache und Semantik zu berücksichtigen



Excerpt

[1] Van der Aalst: Workflow Patterns. Distributed and Parallel Databases 14(1) (2003)

Verwandte Arbeiten: Workflow-Sprachen + deren System-Umgebungen

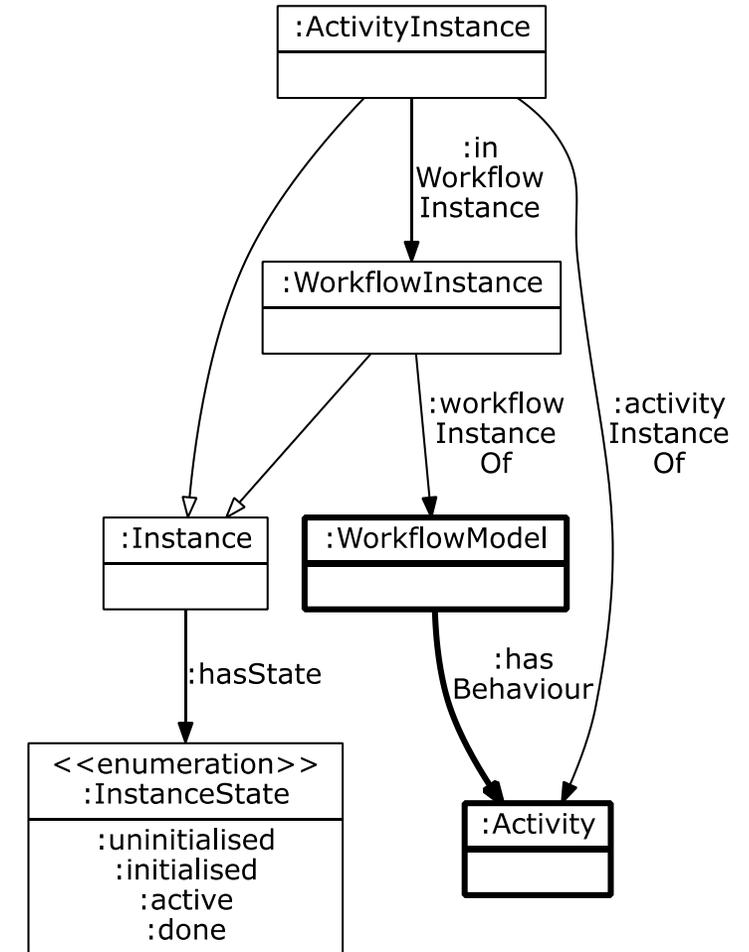
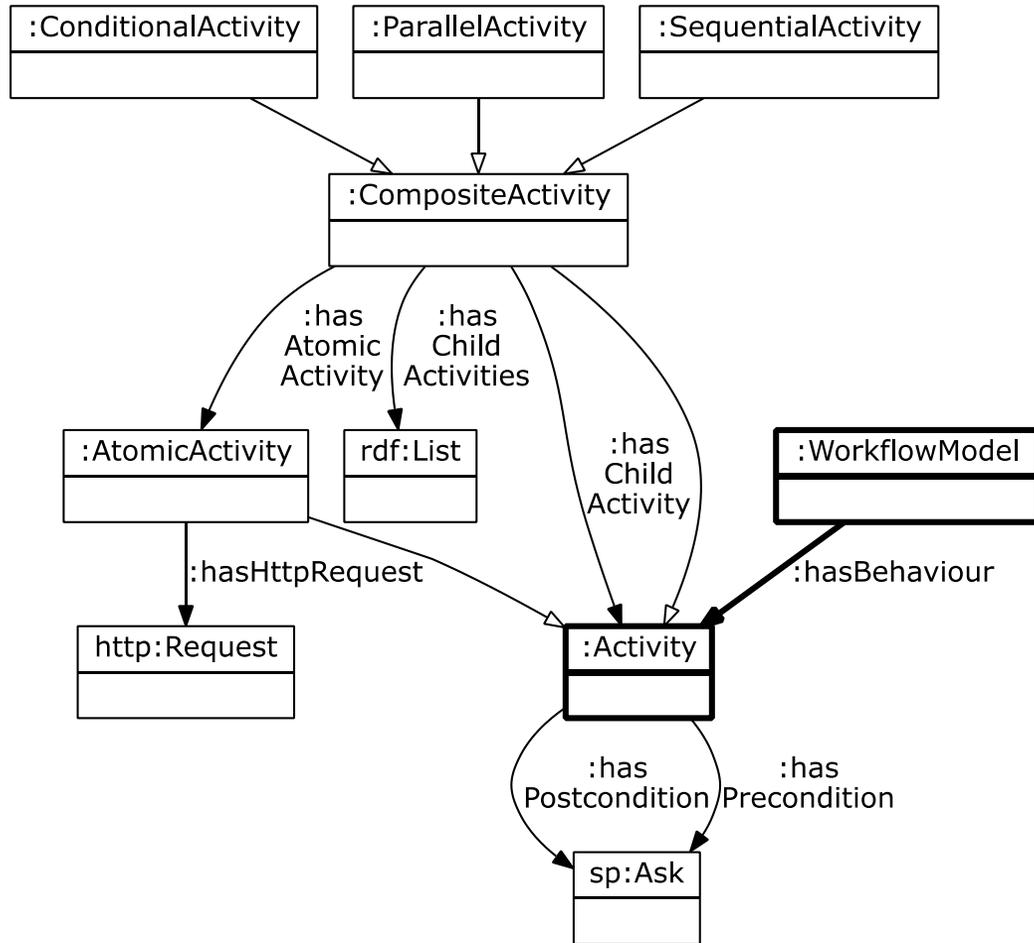


CWA + Ereignisdaten

Ansatz	Bedeutung von Kästen	Bedeutung von Pfeilen
BPEL u.a. (Web Services)	Funktionsaufrufe (liefern Ergebnisdaten)	Abhängigkeiten und Parameter
BPEL u.a. (Human Tasks)	Aufgaben (liefern Ergebnisdaten)	Abhängigkeiten und Parameter
Pegasus u.a. (Kommandozeilenprogramme)	Programme (terminieren)	Abhängigkeiten
Taverna u.a. (Datenstromverarbeitung)	Stromoperatoren (terminieren nicht)	Datenströme
WiLD (Read-Write Linked Data)	?	?

Cf. Elmroth et al.: Three fundamental dimensions of scientific workflow interoperability – Model of computation, language, and execution environment. *Future Generation Computer Systems* 26 (2010)

WiLD-Ontologie – Workflow-Sprache zur Beschreibung baum-strukturierter [1] Workflow-Modelle + -Instanzen



@prefix : <<http://purl.org/wild/vocab#>> .

Excerpt

[1] Jussi Vanhatalo, Hagen Völzer, Jana Koehler: The Refined Process Structure Tree. BPM 2008

Operationale Semantik – Beispielregel: Aktiviere die erste Aktivität einer Sequenz



Excerpt

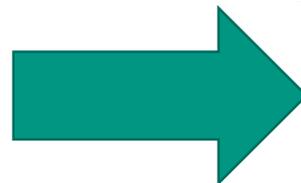
$wild: WorkflowInstance(? wfi) \wedge wild: hasState(? wfi, wild: active)$
 $\wedge wild: workflowInstanceOf(? wfi, ? wfm)$

$\wedge wild: WorkflowModel(? wfm) \wedge wild: containsActivity(? wfm, ? parentact)$
 $\wedge wild: SequentialActivity(? parentact) \wedge wild: hasChildActivities(? parentact, ? act1, ? act2, \dots)$

$\wedge wild: instanceof(? parentacti, ? parentact) \wedge wild: hasState(? parentacti, wild: active)$
 $\wedge wild: instanceof(? act1i, ? act1) \wedge wild: hasState(? act1i, wild: initialised)$

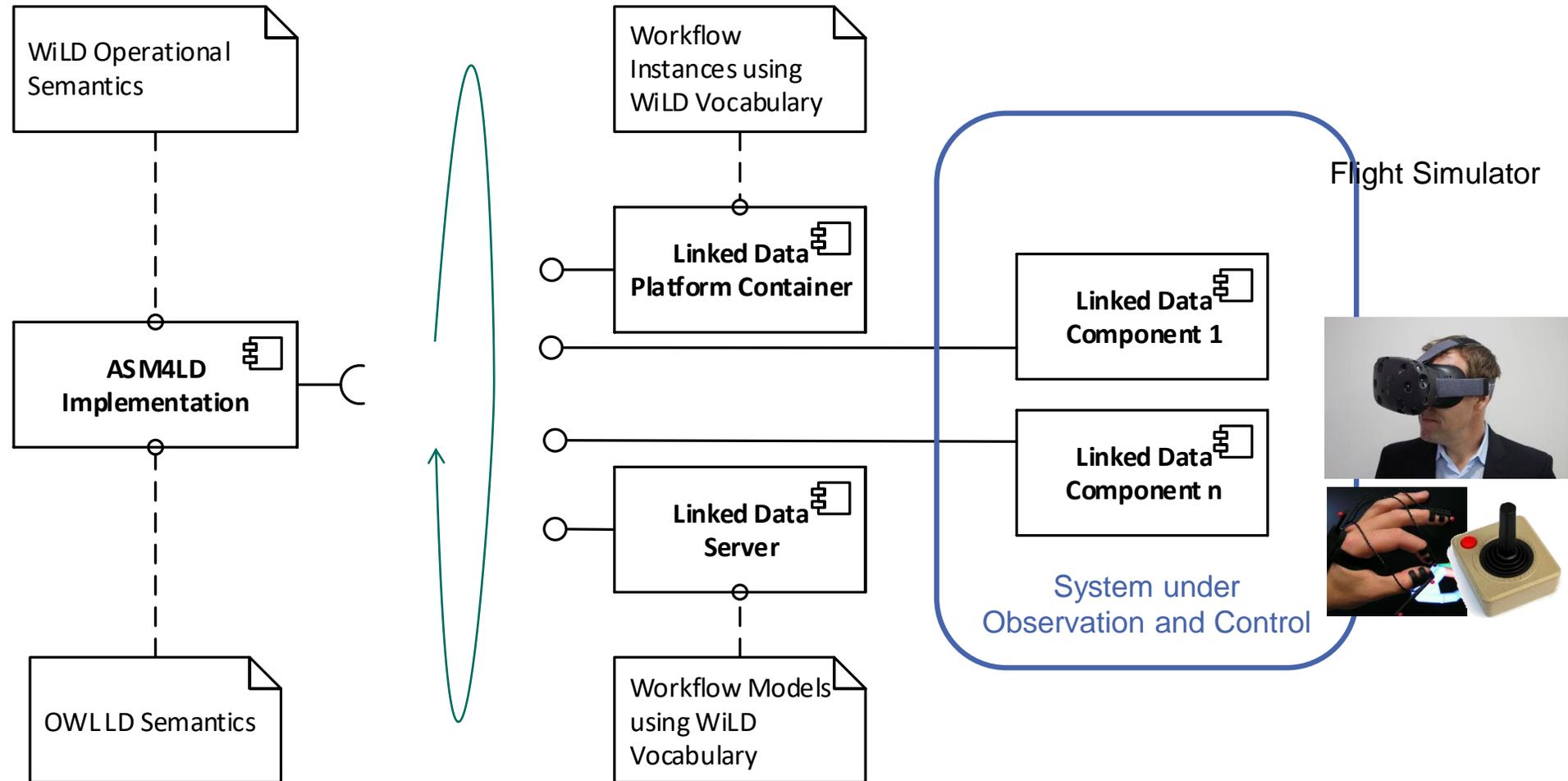
$\wedge wild: inWorkflowInstance(? parentacti, ? wfi) \wedge wild: inWorkflowInstance(? act1i, ? wfi)$
 $\wedge wild: hasHttpRequest(? act1, ? req)$

cf. <http://purl.org/wild/semantics>



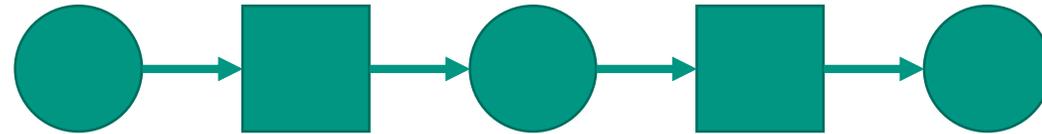
- Setze **?act1i** auf **aktiv** mittels HTTP PUT
- Führe HTTP request **?req** aus

Systemarchitektur mit Workflowmanagement-System, Systemzustand



Evaluation: Formale Analyse mit den Workflow Patterns [1]

- Semantik von Workflow Patterns definiert in Petri-Netzen, z.B.



Petri-Netze für Sequenz:

Abschluss einer Transition setzt eine Marke in die folgende Stelle

- In der operationalen Semantik schalten wir zwischen zwei sequentiellen Aktivitäten nur dann, wenn die erste terminiert hat
- Analog für die anderen Basic Workflow Patterns

[1] Van der Aalst: Workflow Patterns. Distributed and Parallel Databases 14(1) (2003)

Evaluation: Performance-Experimente in Gebäudeautomatisierung



- Datensatz: Read-Write-Linked-Data-Simulation von Gebäude 3 von IBM Dublin, modelliert mittels der Brick-Ontologie [1]
- Last: 5 Workflows W_n ermittelt von Ferme u.a. [2] als repräsentativ für Workflows im Allgemeinen

Average runtime [s] for workflows W_n in different numbers of buildings

	W1	W2	W3	W4	W5
1 Building	2	2	6	12	18
10 Buildings	8	9	26	61	75
20 Buildings	12	13	38	80	109
50 Buildings	19	21	61	156	218

Building 3 and Benchmark Statistics

Rooms	281
Floors	2
Wings	3
Lights w/ occupancy sensors	156
Lights w/ luminance sensors	126
Triples in IBM_B3.ttl, ~2.4MB	24947
Resources in the LDP container	3281
Dynamic resources (sensors)	551

[1] Balaji et al.: "Brick: Towards a Unified Metadata Schema For Buildings". In: Proc. BuildSys@SenSys, 2016.

[2] Ferme et al.: Performance Comparison Between BPMN 2.0 Workflow Management Systems Versions. In: Proc. BPMDS, 2017.

WiLD und das Große Ganze: Szenarien und Schichtenarchitekturen

- Anwendungsszenarien:
 - Verteilt vorliegende Komponenten mit RWLD-Schnittstelle (Wissensgraphen)
 - Datenintegration nötig
 - Komplexe Automatisierungslogik
- Schichtenarchitekturen:

Schichtenarchitektur unseres Ansatzes		Semantic Web
<i>Workflow Meta-Modell</i>	WiLD	RDFS/OWL LD
<i>Rechnermodell</i>	ASM4LD	
<i>Datenmodell</i>	Read-Write Linked Data	RDF
<i>Systeminteraktion</i>	HTTP	URI

[1] Russell, Norvig. Artificial Intelligence – A modern approach. Prentice Hall (1996)

Zusammenfassung: WiLD – Workflows in Linked Data

- Kombination von:
 - REST
 - Semantische KI-Technologien
 - Workflows
- Beiträge zum Stand der Forschung:
 - Ontologie in OWL LD
 - Operationale Semantik in ASM4LD
- Herausforderungen und Lödungen:
 - OWA: RDF-Listen
 - Keine Ereignisdaten: Postconditions für Aktivitäten
- Formal und empirisch evaluiert
- Charakteristika möglicher Anwendungsszenarien:
 - Komponenten weisen Read-Write Linked Data / REST-Schnittstellen
 - Datenintegration vonnöten
 - Komplexe Automatisierungslogik
- Ein Beitrag für intelligente KI-Agenten für das Web
- Vielen Dank für Ihre Aufmerksamkeit!
- tobias.kaefer@kit.edu
- <http://purl.org/wild>

Diese Forschung wurde unterstützt von der EU in Projekt i-VISION and durch das BMBF in Projekt AFAP.

BACKUP

Images

- https://www.w3.org/WoT/IG/wiki/Web_of_Things_Logos
- <https://www.inrupt.com/>
- <https://solid.mit.edu/>
- <http://shop.oreilly.com/product/0636920033158.do>
- https://upload.wikimedia.org/wikipedia/commons/e/e1/Atari_XE_joystick.jpg
- https://upload.wikimedia.org/wikipedia/en/7/71/Msfs1.00_000.png
- [https://upload.wikimedia.org/wikipedia/commons/thumb/3/3b/HTC_Vive_\(16\).jpg/1920px-HTC_Vive_\(16\).jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/3/3b/HTC_Vive_(16).jpg/1920px-HTC_Vive_(16).jpg)
- Hillebrand, Gerrit, et al. "Inverse kinematic infrared optical finger tracking." Proceedings of the 9th International Conference on Humans and Computers (HC 2006), Aizu, Japan. 2006.
- Google maps screenshot