Dynamical Systems on the Web: Classification and Challenges

Prof. Dr. Andreas Harth

Friedrich-Alexander-University Erlangen-Nuremberg and Fraunhofer IIS-SCS

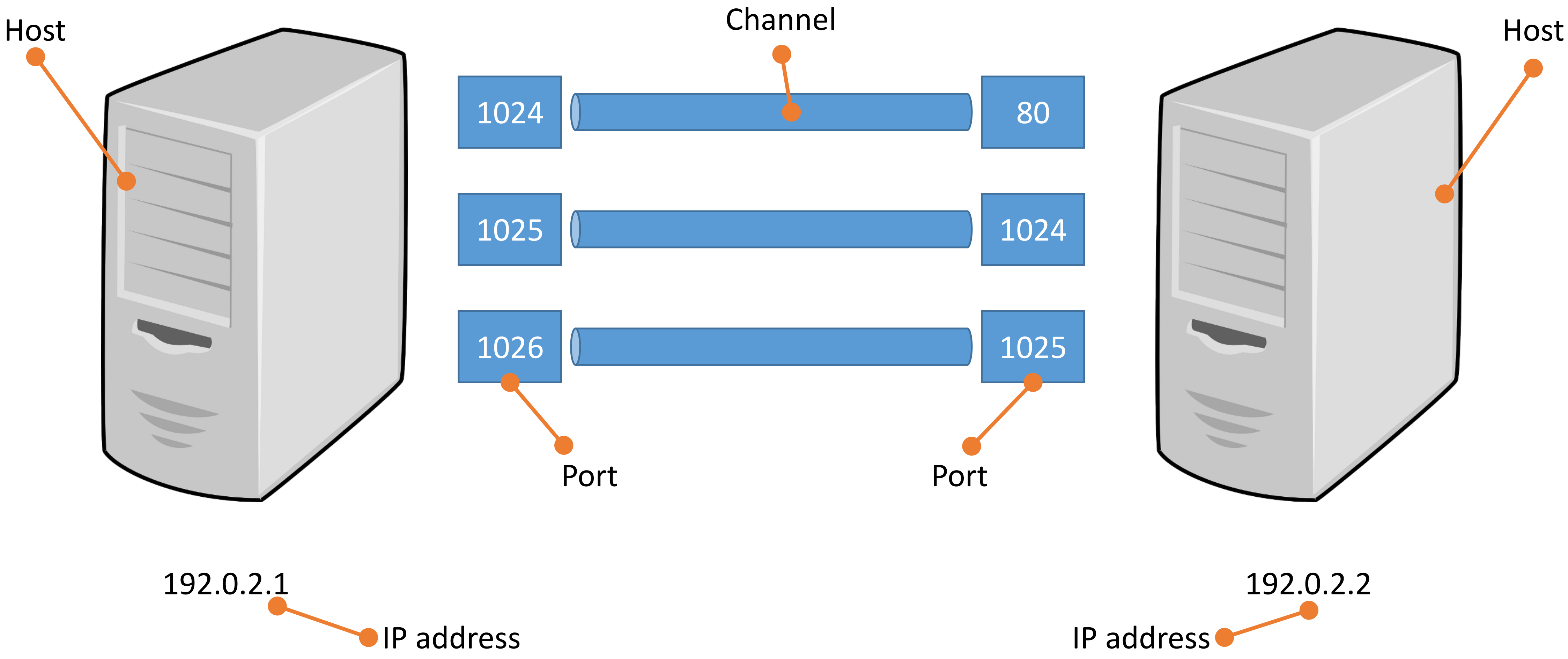
Joint work with colleagues from KIT: Tobias Kaefer, Sebastian Speiser and Steffen Stadtmueller

Fourth Stream Reasoning Workshop, Linköping University, April 2019

Agenda

1. **Internet Architecture vs. Web Architecture vs. Linked Data**
2. User Agents in Dynamical Web Environments
3. Scenarios and Evaluation
4. Conclusion

Internet Architecture: Sockets



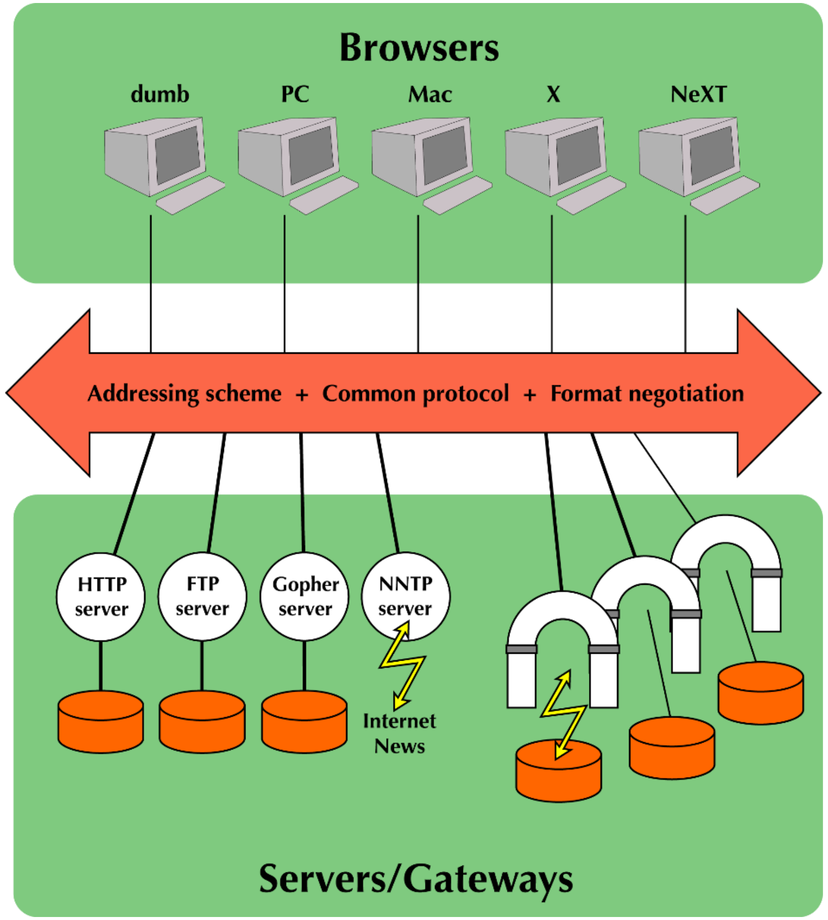
The Web vs. the Internet

https://www.w3.org/20/

Web Architecture

- URIs act as names for resources: RFC 1630 (1994), now RFC 3986
- HTTP to interact with resources/resource state: RFC 1945 (1996), now RFC 7230 - 7235

- Web architecture assumes a strict separation between user agents and servers
- User agents emit requests, receive response
- Servers answer to incoming requests with a response



https://www.w3.org/DesignIssues/diagrams/history/Architecture_crop.png

Semantics of HTTP Messages

HTTP Request Method	HTTP Request, or Response Code	HTTP Message Semantics: The HTTP Message Body contains…
GET	Request	Nothing
PUT	Request	State of the resource
POST	Request	State of the resource or arbitrary data
DELETE	Request	Nothing
Any	Non-2xx	State of the request
GET	2xx	State of the resource
PUT	2xx	State of the resource or empty
POST	2xx	State of the request (refering to new resource)
DELETE	2xx	State of the request or empty

Andreas Harth, Tobias Käfer. "Towards Specification and Execution of Linked Systems". 28. GI-Workshop Grundlagen von Datenbanken, May 24 - 27, 2016, Nörten-Hardenberg, Germany.

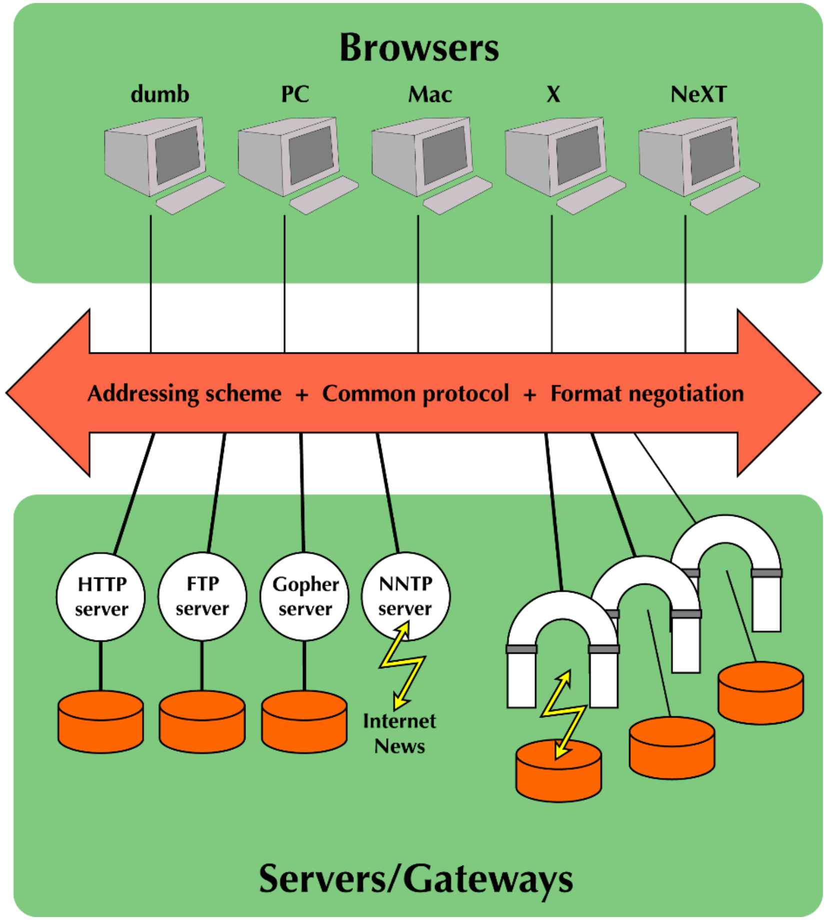
RDF Dataset

Definition (Named Graph, RDF Dataset). *Let $\mathcal{G}$ be the set of RDF graphs and $\mathcal{U}$ be the set of URIs. A pair $\langle g, u \rangle \in \mathcal{G} \times \mathcal{U}$ is called a named graph. An RDF dataset consists of a (possibly empty) set of named graphs (with distinct names) and a default graph $g \in \mathcal{G}$ without a name.*

Web Architecture/Linked Data

- User agent:
- RDF dataset S ⊂ Web

- Servers:
- RDF dataset Web (infinite)



https://www.w3.org/DesignIssues/diagrams/history/Architecture_crop.png

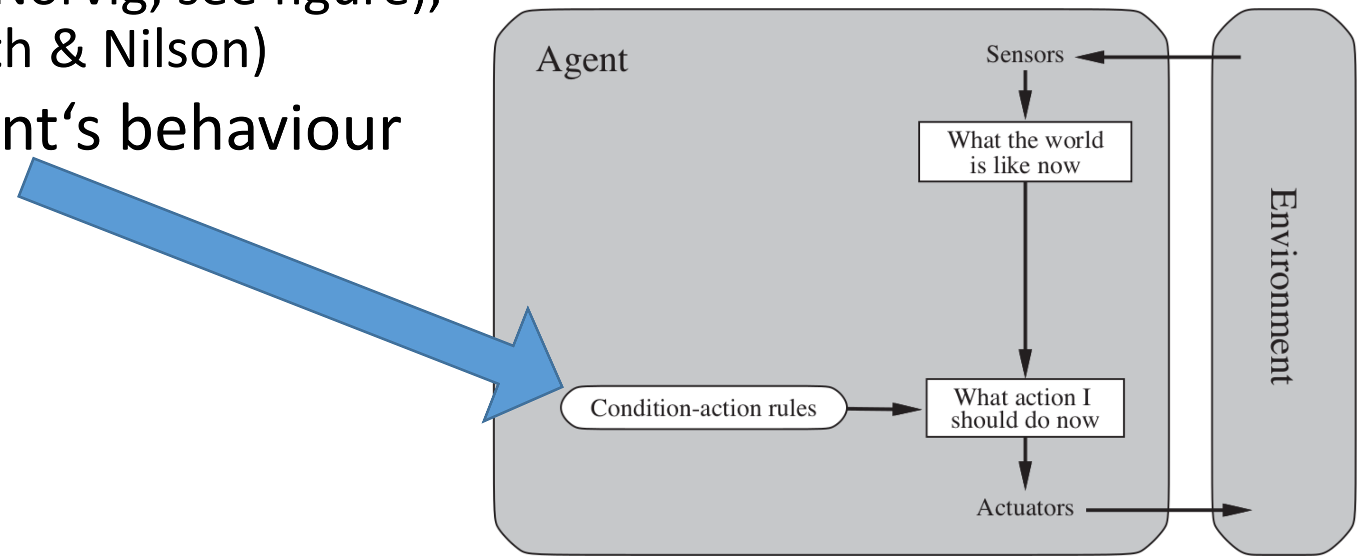
Agenda

1. Internet Architecture vs. Web Architecture vs. Linked Data
2. **User Agents in Dynamical Web Environments**
3. Scenarios and Evaluation
4. Conclusion

Cognitive Architectures

- SOAR (initially: State, Operator, Apply, Result),
- ACT-R (Adaptive Control of Though – Rational)
- Goal: to create „intelligent agents“
- For starters we only consider user agents that are
  - „simple reflex agents“ (Russel & Norvig, see figure),
  - aka „tropistic agents“ (Genesereth & Nilson)
- We use rules to control the agent‘s behaviour
- What the world is like now:
  - safe HTTP methods (GET)
- What action should I do now:
  - unsafe HTTP methods

Russel and Norvig, Artificial Intelligence – A Modern Approach, Third Edition, 2010



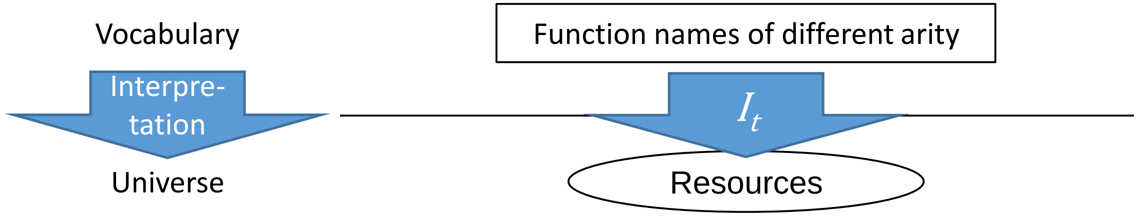
Some Models of Computation

Candidates

Model of Computation	Main Mismatch to Linked Data
Lambda Calculus	Based on Events / Functions
Pi Calculus	Based on Events / Channels
Petri Nets	Based on Events
Graph Rewriting	Unclear data access + FOL-handling
Turing Machine	Abstraction too low-level
Finite State Machines	Unclear state + condition representation
Abstract State Machines	?

Abstract State Machines [G]

- Provide a good fit to Linked Data
  - First-order logic-based (cf. RDF(S)/OWL)
  - State as first-class citizen (HTTP)
- About the evolution of first-order structures (aka. states)
  - Specifically, how the interpretation of function names changes over time



- Evolution (so-called transition function) in rules:
  - **If** *condition(s) hold* **then** *update the interpretation(s)*
- Execution in ASM Steps:
  - Collect all updates, execute updates in bulk

[G] Yuri Gurevich: “Evolving algebras: Lipari guide”. In: E. Börger (ed.): Specification and validation methods. Oxford, United Kingdom: Oxford University Press (1993)

Abstract State Machines for Linked Data

Basic Definitions / Simplifications:

- Ground graphs (ie. no blank nodes)
- U – the set of all URIs, L – the set of all Literals, interpreted to the same resources in all graphs
- $IP \subseteq IR$ (required in RDF and more constrained interpretations)
- No HTTP redirects

1. **Define RDF model theory for Linked Data using RDF datasets**
   - Different extension functions (IEXT) in [Z] for RDF datasets:
     a) For updates: *Named Graphs are in a particular relation with what the graph refers to*
        $IEXT_c := \ Extension\ function\ of\ the\ graph\ available\ at\ c$
     b) For conditions: *Default graph as union or as merge*
        $$IEXT^{UNION} \coloneqq \bigcup_c IEXT_c$$

2. **Define ASM functions for the model theoretic views on Linked Data / RDF datasets**
   - $quad(\cdot,\cdot,\cdot,\cdot): IR \times IR \times IR \times IR \rightarrow \{\text{TRUE}, \text{UNDEF}\}$ the ASM characteristic function for the set of all quads ~ IEXT in a)
   - $statement(\cdot,\cdot,\cdot): IR \times IR \times IR \rightarrow \{\text{TRUE}, \text{UNDEF}\}$ the ASM characteristic function for the set of all triples ~ IEXT in b)

[Z] Antoine Zimmermann (ed.): “RDF 1.1: On Semantics of RDF Datasets”, W3C RDF WG Note, 2014.

Abstract State Machines for Linked Data

3. **Define an ASM transition function $\boldsymbol{T}$ for the Linked Data ASM functions**
   - **If** conditions hold in $statement(\cdot,\cdot,\cdot)$ **then** update $quad(\cdot,\cdot,\cdot,\cdot)$
   - Conditions in $statement(\cdot,\cdot,\cdot)$ ~ SPARQL BGP Queries

4. **Define how the ASM evaluation of the ASM functions maps to the HTTP request semantics**
   - $statement(\cdot,\cdot,\cdot)$ in conditions ~ GET request to all sources
   - $quad(\cdot,\cdot,\cdot,\cdot)$ in updates ~ PUT request to given source(s)

5. **Define the ASM $(\boldsymbol{\Upsilon}, \boldsymbol{X}, \boldsymbol{I}, \boldsymbol{T})$ using the semantic conditions**
   - $\Upsilon \coloneqq U \cup L \cup \{true, undef\} \cup \{\wedge\} \cup \{quad, statement\}$
   - $X \coloneqq IR \cup IP \cup \{\text{TRUE}, \text{UNDEF}\} \cup \{\, f \mid X^n \to X\}$
   - $I_t(y) \coloneqq \begin{cases} IS(y) & if\ y \in U \\ IL(y) & if\ y \in L \\ \text{TRUE} & if\ y = true \\ \text{UNDEF} & if\ y = undef \\ \in \{f \mid f\colon X^n \to X\} & if\ y \in \{quad, statement, \wedge\} \end{cases}$

- Execute following **ASM steps**: First evaluate all conditions, then apply the collected updates in bulk

Result: Operational Semantics for the Linked Data-Fu Language [SSHS]

```
@prefix … # See prefix.cc
```

Assertions

```
<http://building3.example/lamps/0#l>
  rdf:type saref:LightingDevice ;
  ssn:hasProperty <http://building3.example/lamps/0/power#p> .
```

RDF / quad(), statement()

Deductions

```
{ ?thing ssn:hasProperty ?prop . }
=> { ?prop ssn:isPropertyOf ?thing . } .
```

Reasoning / Semantic Conditions

Conditional GET requests

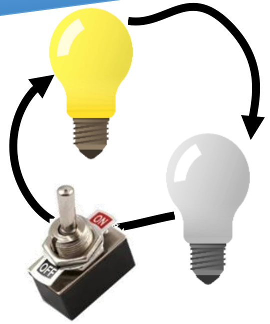
```
{ ?y ssn:isPropertyOf ?x . }
=> { [] http:mthd httpm:GET ; http:requestURI ?y . } .
```

Retrieve world state / quad(), statement()

Conditional PUT/POST/DELETE requests

```
{ ?lamp a saref:LightingDevice .
  ?property ssn:isPropertyOf ?lamp ; rdf:value "off" . }
=> { [] http:mthd httpm:PUT ; http:requestURI ?property ;
        http:body { ?property rdf:value "on" . } . } .
```

Define the logic and write state / T



[SSHS] Steffen Stadtmüller, Sebastian Speiser, Andreas Harth and Rudi Studer. “Data-Fu: A Language and an Interpreter for Interaction with Read/Write Linked Data”. In: Proceedings of the 22nd International Conference on World Wide Web (WWW), 2013.

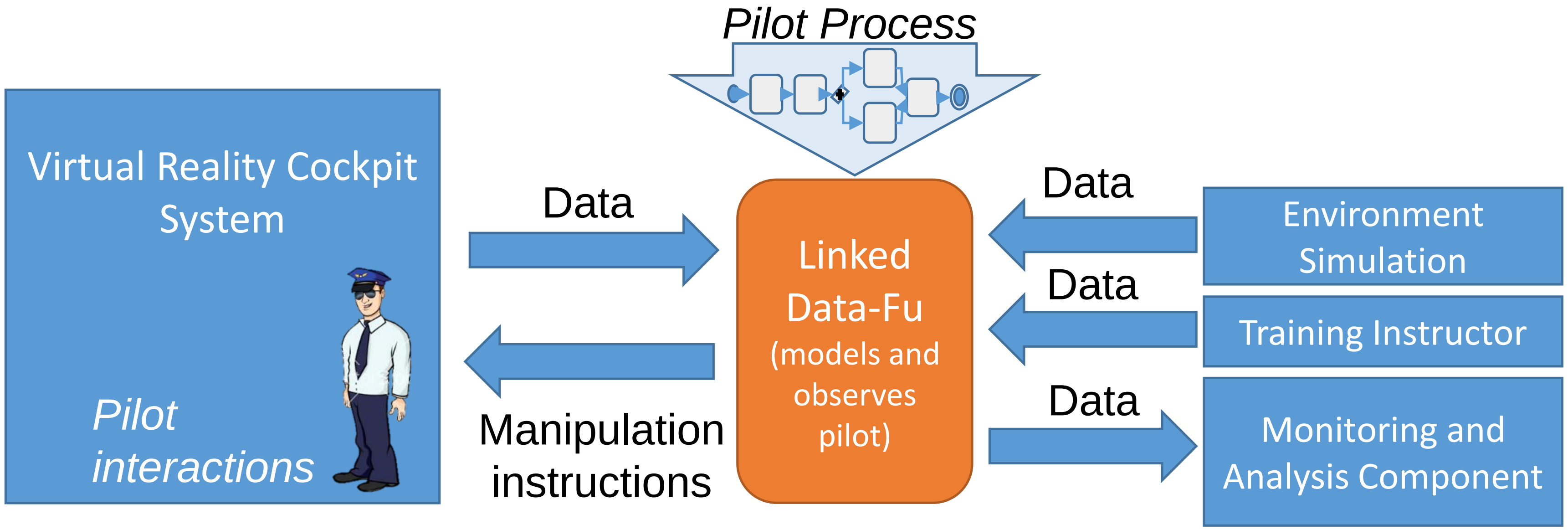
Agenda

1. Internet Architecture vs. Web Architecture vs. Linked Data
2. User Agents in Dynamical Web Environments
3. **Scenarios and Evaluation**
4. Conclusion

Scenario: Data Processing within Virtual and Augmented Reality Environments

Scenario “Virtual Airplane Pilot Training”

- **Workflows** of the pilot in a plane are derived by human factor methods
  - E.g., behavior during emergency landing
- **Decisions** depend on a multitude of environment conditions, including the behavior of the actual pilot
- **Linked Data-Fu**: specification and execution of dynamic workflows in a real training scenario



Scenario: Building Automation/Evaluation

1. Formal: Turing Completeness
2. Performance
   - Automating Building 3 of IBM Dublin, as described using the Brick ontology [1]
   - Interpreter: Linked Data-Fu
   - W1→W5 differently complex automation
   - D1: GET building data from *one source*
   - D2: GET building data from *many sources* following links

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

Median Time [ms] for One ASM Step in D1

Rooms	W1	W2	W3	W4	W5
1	484	572	510	554	561
5	480	582	501	574	582
10	498	584	529	605	618
20	537	631	562	719	687
First Floor	563	629	590	750	728
Wing 42	527	595	550	651	604
Building 3	605	734	613	794	788

Same order of magnitude

Median Time [ms] for One ASM Step in D2

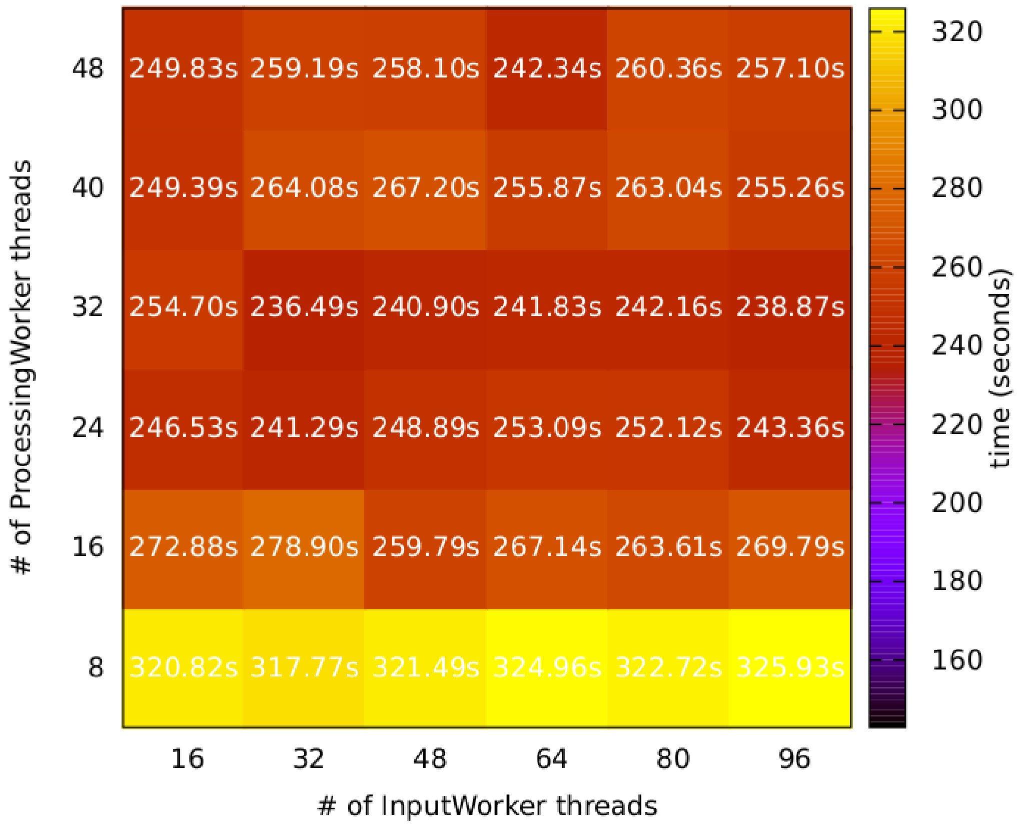
Rooms	W1	W2	W3	W4	W5
1	8	8	8	8	8
5	40	38	38	40	40
10	85	80	79	88	88
20	259	238	228	320	268
First Floor	938	1690	891	1063	1048
Wing 42	1435	1427	1371	1664	1408
Building 3	2442	2187	2192	2542	2497

Raising with number of Rooms

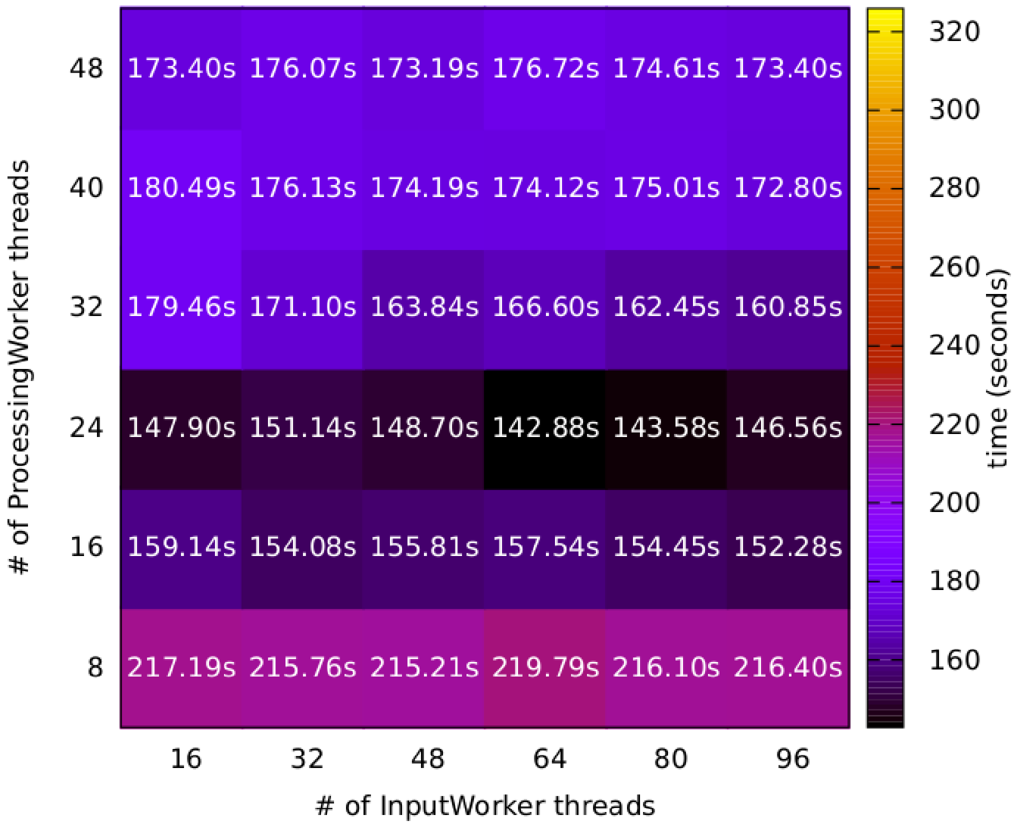
[1] Balaji et al.: “Brick: Towards a Unified Metadata Schema For Buildings”. BuildSys@SenSys 2016

Balancing IO and Reasoning on LUBM-LD(100)

Batch Processing: first IO, then processing, then IO,…



Stream Processing: IO and processing intertwined



InputWorker threads: I/O, ProcessingWorker threads: reasoning/materialisation

See: Andreas Harth, Link Traversal and Reasoning in Dynamic Linked Data Knowledge Bases, Habilitation Thesis, KIT 2015

Agenda

1. Internet Architecture vs. Web Architecture vs. Linked Data
2. User Agents in Dynamical Web Environments
3. Scenarios and Evaluation
4. **Conclusion**

Conclusion

- We have shown how to build systems in building automation and mixed reality systems
- Other scenarios could involve virtual assistants
- We adapted Abstract State Machines for Linked Data as a formalism for specifying user agent behaviour, and used ASM4LD to give an operational semantics to a workflow vocabulary
- We have a multithreaded implementation (Linked Data-Fu)

- Future work: How to use link-following for discovery and goal-directed user agent behaviour?
- Future work: How to leverage formalisms based on state machines to perform model checking (finite models) or simulation (infinite models)?

Further Reading

- Tobias Käfer, Andreas Harth. "Specifying, Monitoring, and Executing Workflows in Linked Data Environments". International Semantic Web Conference 2018 (ISWC 2018), October 8-12, 2018, Monterey, California, USA
- Tobias Käfer, Andreas Harth. "Rule-based Programming of User Agents for Linked Data". WWW2018 Workshop on Linked Data on the Web (LDOW2018), April 23, 2018. Lyon, France.
- Andreas Harth, Tobias Käfer, Felix Leif Keppmann, Dimitri Rubinstein, René Schubotz, Christian Vogelgesang. "Flexible industrielle VT-Anwendungen auf Basis von Webtechnologien". VDE Kongress 2016, Internet der Dinge, Nov 7-8, 2016, Mannheim, Germany.
- Tobias Käfer, Sebastian Bader, Lars Heling, Raphael Manke and Andreas Harth. "Exposing Internet of Things Devices on REST and Linked Data Interfaces". 2nd International Workshop on Interoperability & Open Source Solutions for the Internet of Things. Co-located with 6th International Conference on the Internet of Things (IoT 2016). Nov 7, 2016, Stuttgart, Germany.
- Felix Leif Keppmann, Maria Maleshkova, Andreas Harth. "Semantic Technologies for Realising Decentralised Applications for the Web of Things". 21st International Conference on Engineering of Complex Computer Systems, Nov 6-8, 2016, Dubai, UAE.
- Sarah Brauns, Tobias Käfer, Dirk Koriath, Andreas Harth. "Individualisiertes Gruppentraining mit Datenbrillen für die Produktion". GI-Jahrestagung 2016.
- Andreas Harth, Tobias Käfer. "Towards Specification and Execution of Linked Systems". 28. GI-Workshop Grundlagen von Datenbanken, May 24 - 27, 2016, Nörten-Hardenberg, Germany.

Time: Synchronised Clocks

- Synchronized clocks are difficult to achieve in distributed systems with many participants
- On the web, resource state is usually just “now”
- RFC 7089 “HTTP Framework for Time-Based Access to Resource States – Memento” allows for accessing previous resource state (https://www.rfc-editor.org/rfc/rfc7089.txt)