Karlsruhe Institute of Technology

The Architecture of Future Automotive Applications based on Web Technologies

Luka Bradesko, Marko Grobelnik, Andreas Harth, Achim Rettinger and **Sebastian Speiser** (JSI, AIFB, KSRI)

W3C Workshop on Web and Automotive Rome, Italy November 13th 2012

Karlsruhe Service Research Institute (KSRI), Institute of Applied Informatics and Formal Description Methods (AIFB)

Agenda

- Motivation
- Overview
- Memory: Virtual Knowledge Base
- Interoperation Layer
- Conclusions

MOTIVATION

Motivation

- Besides reaching B from A, a driver has other goals, e.g.,
  - Adjusting delivery routes according to traffic
  - Finding cheapest petrol station along the route
  - Handling the radio and phone
- Result:

Source: http://www.adslogistics.com/blog/bid/53021/Avoid-the-Three-Types-of-Distracted-Driving

Motivation

A rare example of a driver solely concentrated on driving

…. but …

Source: http://www.f1-site.com/wallpapers/2009/f1/abu_dhabi/drivers/photo-f1-team-driver-wallpaper-2009-3.jpg

Motivation

A lot of assistants help out in the background with a convenient speech interface

Source: http://www.f1-site.com/wallpapers/2009/f1/abu_dhabi/drivers/photo-f1-team-driver-wallpaper-2009-3.jpg

Motivation

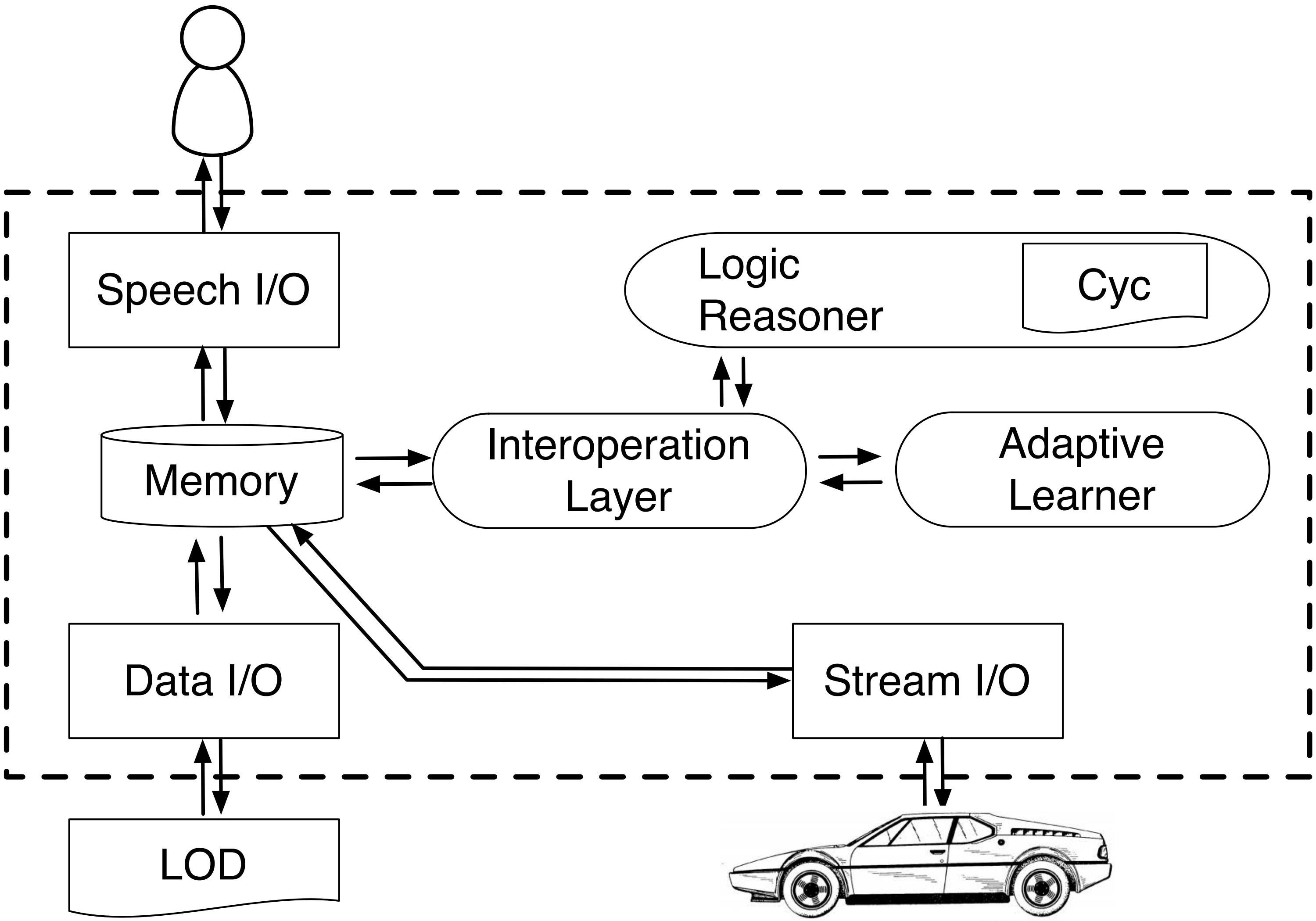
- Formula 1 solution is impractical for widespread use: costs of about 1,000 USD / kilometer

  source: http://www.faz.net/aktuell/sport/formel-1/was-kostet-die-formel-1-ein-teures-rennvergnuegen-1257804.html

- Our proposed solution:
  Driver assistant systems based on
  - Speech recognition
  - Virtual knowledge base integrating data from the web
  - Logical inferences
  - Statistical learning

OVERVIEW

High-level Architecture

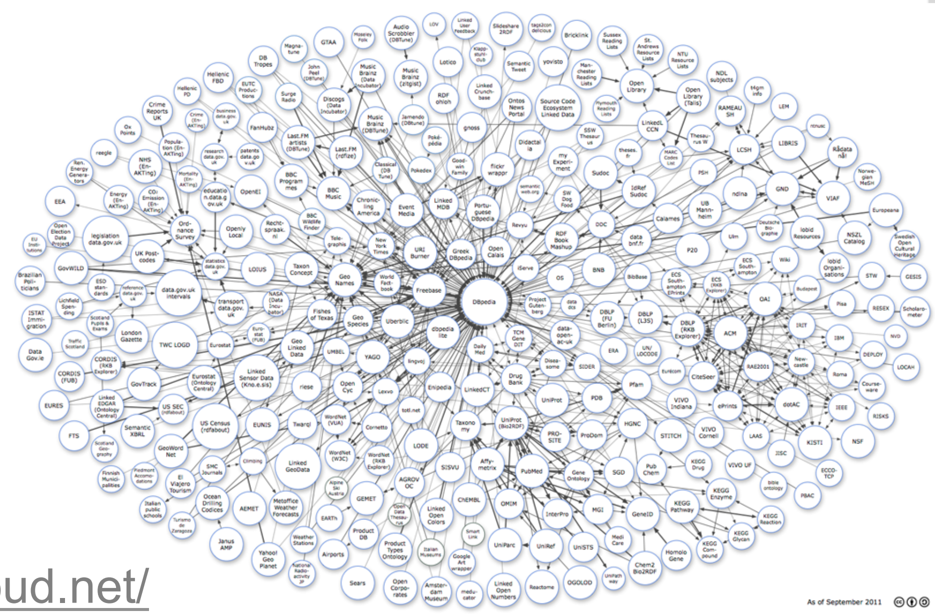


Overview of Architecture

- Speech I/O: Conversational interface
  - use patterns only to help us converting text to logic
  - reasoning engine to control the conversation
  - replies: use Cyc's reasoner logic-to-language translation
  - reasoning lets us get rid of the need for lots of prepared patterns
- Stream I/O
  - constant stream of data and events
  - source: car sensors, web streams, user sensors
  - representing information about car, environment, user

Data I/O: Linking Open Data Cloud

- Linked Data (RDF data accessible via HTTP lookups), 2006
- Yearly growth rates of ~200%
- Many datasets, covering descriptions of millions of entities
- Large number of interlinked distributed disparate small data sources rather than single-source single-organisation knowledge bases



source: http://lod-cloud.net/

Combining and Using I/O Channels

- All the data is put in a virtual knowledge base (more in a minute)
- Interoperation layer relates data from different channels (more in two minutes)

- Based on the data infer new knowledge
  - Logic reasoner (exact methods)
  - Learning component (statistical and heuristic methods)

MEMORY: VIRTUAL KNOWLEDGE BASE INTEROPERATION LAYER

Memory: Virtual Knowledge Base

- Based on Linked Data and REST principles

- Basic abstraction: a resource
  - different representations, e.g., a POI as 3D XML, RDF, or JPEG
  - resources have references (links) to other related resources, e.g., leading to the next step in a series of operations
  - support standard operations: CRUD

- A resource has an identifier (HTTP URI)
- The identifier specifies a way to access information about the resource (performing a HTTP lookup)
- The information is in standardised format (an RDF graph describing the resource using its identifier)

Interlinking in Linked Data

- Establishing equivalencies across sources, e.g.,
  `car:currentDriver owl:sameAs facebook:JohnDoe`
  `(car: stands e.g. for http://localhost/...)`
- All statements about JohnDoe also apply to current driver and vice versa

- Using URIs across services, e.g.,
  ```
  car:car :location car:point
  car:point foaf:based_near wikipedia:Rome
  ```

Streams, Services and Compositions

- Streams, either (depending on frequency of updates)
  - just do not close HTTP connection and continue to list
  - pull “stream source” regularly

- Services / dynamic data / data with limited access patterns
  - integrate with Linked Data

- How to build
  - applications,
  - compositions,
  - workflows
- based on REST resources?

Linked Services

- Linked Services:
  - service approach based on web architecture (REST)
  - describing web resources with RDF/Notation3
    aspects: input, output, relation between input and output)
  - linking between: services, descriptions, data in input and output

- CRUD operations on resources

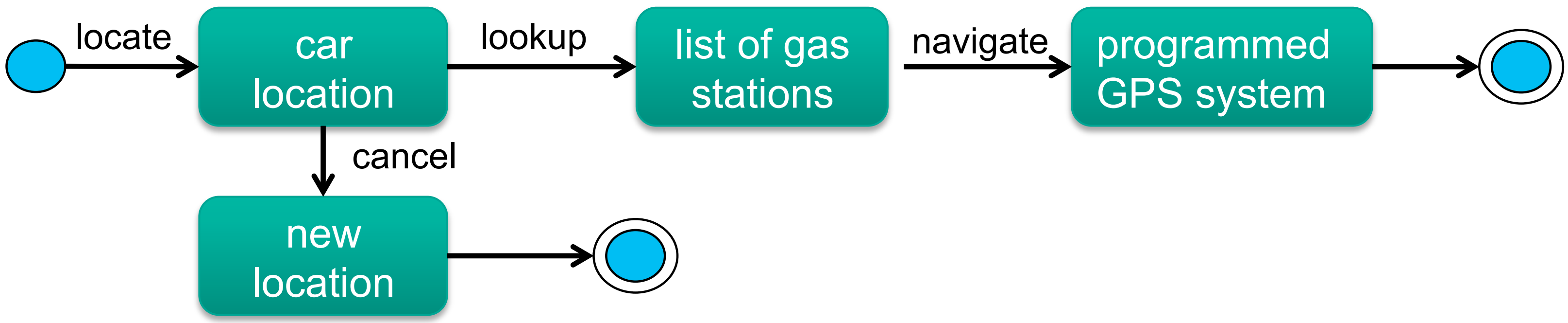
Barry Norton and Reto Krummenacher. Consuming Dynamic Linked Data. In Proceedings of the 1st International Workshop on Consuming Linked Data (COLD'10), 2010.

Sebastian Speiser and Andreas Harth. Integrating Linked Data and Services with Linked Data Services. In Proceedings of the 8th Extended Semantic Web Conference (ESWC'11), 2011.

Production Rules for Linked Services

- Decentralised linking between resources
- Use production rules to specify composition of resources
- Depending on current state of knowledge base (KB):
  - invoke new services (add their output to KB)
  - find links to new services in the KB

Example: navigate to gas station



Steffen Stadtmüller, Andreas Harth. „Towards Data-driven Programming for RESTful Linked Data“. Workshop PSW, ISWC 2012.

CONCLUSIONS

Conclusions

- Using web technologies to provide assistant system to driver’s at reasonable cost

- Speech interface based on logic reasoning

- Web technologies enable interoperability
  - new sources can be added on demand
  - pay-as-you go for integration
  - we can tap the long tail of sensors and data sources

Thank you for your attention.

www.ksri.kit.edu

www.aifb.kit.edu

Supported by the iZEUS project
(German Federal Ministry of Economics and Technology)

The Architecture of Future Automotive Applications based on Web Technologies

Luka Bradesko, Marko Grobelnik, Andreas Harth, Achim Rettinger and **Sebastian Speiser** (JSI, AIFB, KSRI)

W3C Workshop on Web and Automotive Rome, Italy November 13th 2012

Karlsruhe Service Research Institute (KSRI), Institute of Applied Informatics and Formal Description Methods (AIFB)