Service Oriented Architecture (SOA) and the Integration of Online Learning-based Applications

Simon Speth, Johannes Landgraf

Technical University of Munich
Department of Informatics
Chair for Robotics, Artificial Intelligence and Real - time Systems

Garching, 01.February.2019
Outline

1 Motivation
2 Service Oriented Architecture
3 Design of an Automotive Architecture
4 Guarantee of Safety and Performance
5 Implementation
6 Conclusion and Outlook
Modern car: 100+
Facebook: 62
Windows Vista: 50
Large Hadron Collider: 50
Linux Kernel 3.6: 16
Android: 15
Boeing 787: 7
Outline

1 Introduction and Motivation
2 Service Oriented Architecture
3 Design of an Automotive Architecture
4 Guarantee of Safety and Performance
5 Implementation
6 Conclusion and Outlook
Evolution of SOA

<table>
<thead>
<tr>
<th>1990s and earlier</th>
<th>2000s</th>
<th>2010s</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Coupling</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pre-SOA (monolithic)</td>
<td>Traditional SOA</td>
<td>Microservices</td>
</tr>
<tr>
<td>Tight coupling</td>
<td>Looser coupling</td>
<td>Decoupled</td>
</tr>
</tbody>
</table>

[Diagram showing the evolution of SOA from pre-SOA (monolithic) with tight coupling, to traditional SOA with looser coupling, to microservices with decoupled coupling.]
What is SOA?

- Architectural pattern in computer software design
- Enables application functionality to be provided as a set of services
- Allows creation of applications that make use of software services
- Couples services loosely by using standards-based interfaces
SOA Key Principles

- Loose Coupling
- Composability
- Abstraction
- Discoverability
- Reusability
- Statelessness
- Autonomy

Key Principles
Service Contract

- Service Contract
- Documents expressing meta information
- Technical interface of service
- Service Level Agreement (SLA) specifies quality-of-service features
Service-oriented analysis and design phase

Technology-agnostic

- Identify service candidates
- Identify service composition candidates

Technology-specific

- Specification of service API
- Specification of service level agreements (SLA)

Service-oriented analysis → Service logic design → Service implementation design
Service properties

- **Granularity** (basic – composite)
- **Weight** (heavy – light)
- **Runtime environment** (onboard – offboard)
- **Communication pattern** (synch – asyn)
- **Runtime service binding** (dynamic – static)
- **Scope** (powertrain etc.)
Outline

1. Introduction and Motivation
2. Service Oriented Architecture
3. Design of an Automotive Architecture
4. Guarantee of Safety and Performance
5. Implementation
6. Conclusion and Outlook
ECU Based Architecture

Drawbacks:
- Complex architecture
- Communication over different Bus Systems (CAN, LIN)
- Mass ~ 50kg
- No central data gateway
Zone Architecture
Combining Zone Architecture with SOA

Simpler architecture
- Clear interconnection of services
- High level of abstraction
- Logical composability
- Central data fusion place
- High bandwidth for computer vision and machine learning tasks
- Upgradability of hardware and software (gpu sockets / over the air update)
SOA enables Hardware Independence

- Different types of Sensors:
  - Ultrasound
  - Lidar (Laser Scanner)
  - Radar
  - Camera

- Upgradability of sensors:
  - Camera cheapest
  - Later expanded with lidar
Advantages of SOA for AI in cars

- More structured Online learning implementation → cheaper
- Faster development
- Developing new features/algorithms for customers after car sale

Examples for autonomous car algorithms:
- Computer Vision
- Sensor fusion
- Trajectory planning

- Parallelizable
- Computational Complex
Computer Vision Computational Complexity

Nvidia estimation:

- Feature Detection / Tracking: 30 GFLOPS (30 Hz)
- Object Recognition: 180 GFLOPS (30 Hz)
- 3D Scene Interpretation: 280 GFLOPS (30 Hz)
Online learning in the car or in cloud?

On-car training not yet feasible for safety critical functions:

- On-car training:
  - Danger of malicious learning
  - Need to install expensive hardware → car cost

- Off-car training:
  - Store produced data
  - 5G connection for sending training data to server
Outline

1  Introduction and Motivation
2  Service Oriented Architecture
3  Design of an Automotive Architecture
4  Guarantee of Safety and Performance
5  Implementation
6  Conclusion and Outlook
Functional safety overview

- Safety defined as minimization of risk and uncertainty
- Functional safety of automotive software addressed by ISO 26262
- Automotive Safety Integrity Level (ASIL)

![Safety Integrity Level (ASIL) Table]

<table>
<thead>
<tr>
<th>Severity class</th>
<th>Probability class</th>
<th>Controllability class</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E2</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E3</td>
<td>QM</td>
<td>A</td>
</tr>
<tr>
<td>E4</td>
<td>QM</td>
<td>B</td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>QM</td>
<td>QM</td>
</tr>
<tr>
<td>E2</td>
<td>QM</td>
<td>A</td>
</tr>
<tr>
<td>E3</td>
<td>QM</td>
<td>B</td>
</tr>
<tr>
<td>E4</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>QM</td>
<td>A</td>
</tr>
<tr>
<td>E2</td>
<td>QM</td>
<td>B</td>
</tr>
<tr>
<td>E3</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>E4</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>
Middleware crucial for ASIL rating

- Data Distribution Service (DDS)
  - Scalable, distributed, interoperable data sharing
  - Pub / Sub System

- Time-Sensitive Networking (TSN)
  - Real-time communication over Ethernet
  - Guarantees of delivery
Checking time constraints is hard

- Services may be deployed on different platform
- Bring your own device
- Assume the worst case
- Enforce static service binding

Static service binding

Dynamic service binding
Monitor / Actuator services

- Actuator designed on low ASIL
- Vehicles have short failover missions
- Global Scope:
  - High-ASIL monitor service
  - Other services run low-ASIL
Black Box service

- Accident reconstruction data
- Important for legal issues
- Safety assurance argument
- Valuable training data
- Fits in zone architecture
- Log DDS
Outline

1 Introduction and Motivation
2 Service Oriented Architecture
3 Design of an Automotive Architecture
4 Guarantee of Safety and Performance
5 Implementation
6 Conclusion and Outlook
Autonomous driving platform
Failover test

- Failover test from Kugele et al.
  - Failover time $t_{t0}$
  - Response interval $t_i$
  - Service deployed in cloud $s_A^{\text{Cloud}}$
  - Service deployed locally $s_A^{\text{Local}}$
Implementation Cloud Service

1. Car sends request to cloud service over the internet
2. Google Server computes response
3. Car receives response over internet
Implementation Local Service

1. Car sends request to local service over Wi-Fi network
2. Local Tomcat Server computes response
3. Car receives response over local network
Measurements of failover time

<table>
<thead>
<tr>
<th></th>
<th>$t_{i,\text{cloud}}$</th>
<th>$t_{i,\text{local}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.740</td>
<td>0.066</td>
</tr>
<tr>
<td>median</td>
<td>0.557</td>
<td>0.033</td>
</tr>
<tr>
<td>max</td>
<td>5.446</td>
<td>1.830</td>
</tr>
<tr>
<td>min</td>
<td>0.312</td>
<td>0.022</td>
</tr>
<tr>
<td>std</td>
<td>0.572</td>
<td>0.167</td>
</tr>
</tbody>
</table>
Implementation Result

- Median Cloud takeover time 859ms
- Median Local takeover time 335ms
- Both not real time capable
- Worst case analysis would clearly lead to crash
- Cloud service one order of magnitude slower than local service
- Cloud service not suited for safety critical failover
Outline

1. Introduction and Motivation
2. Service Oriented Architecture
3. Design of an Automotive Architecture
4. Guarantee of Safety and Performance
5. Implementation
6. Conclusion and Outlook
Conclusion

- SOA for hardware independence
- SOA improves deployment and development
- Cloud services no real time guarantee
- High ASIL monitor service for failover initiation
- Middleware has to support TSN and DDS
- Dynamic binding in safety critical functions not possible
Thank You for Your attention!

Any Questions?