Open Source and Standardization

Hyundai MOBIS (현대모비스)

Jeff Kim / 김정회 (jeff.kim@mobis.com)

About me

Started software development in 1990; over 30 years as a passionate developer:

- ➔ **Job**

  Hyundai Mobis(현대모비스) / Next Generation Platform Team (차세대플랫폼팀) / Principal Research Engineer(책임연구원)

- ➔ **Interests**

  Rust Program Language, SDV, LLM

- ➔ **Personal Project**

  Github - rsBinder / Reimplementing Android Binder IPC in Rust

The Cathedral and The Bazaar (성당과 시장)

Author: Eric S. Raymond (1997)

Inspired by **Linux development** and influenced **Netscape's decision to open-source** its browser, leading to the Mozilla project.

	Cathedral(성당) Model	Bazaar(시장) Model
Control	Centralized	Decentralized
Releases	Infrequent, Major	Frequent, Incremental
Development	Closed, Limited Developers	Open, Community-Driven
Feedback	Limited Pre-Release	Continuous and Integrated

The Conclusion of The Browser Wars

The Complete Victory of Open Source

Original Browser	Final/Current Browser	New Engine
Internet Explorer	**Microsoft Edge**	**Blink (Chromium)**
Netscape Navigator	**Mozilla Firefox**	**Gecko**
Opera	**Opera**	**Blink (Chromium)**

1998: KHTML (KDE's Engine) → 2001: WebKit (Developed by Apple) → 2005: WebKit Open Sourced → 2013: Blink (Forked by Google from WebKit) → Ongoing: Chromium

The Economic and Social Value of Open-Source

1. **Importance of OSS**: OSS is integral to modern technology and supports most of the software and services we use today.
2. **Supply-side value**: The cost to recreate widely used OSS is estimated at $4.15 billion(5조 5천억).
3. **Demand-side value**: If OSS didn't exist, companies would face costs of about $8.8 trillion. (1경 1700조)
4. The cost to redevelop the Linux kernel grew from **$612 million in 2004** to approximately **$14.7 billion (19.5조) by 2018**.

Standardization

Process

Requirements gathering → Formation of working group → Draft development → Public review and feedback → Revisions and finalization → Approval and publication → **Implementation** and compliance → Continuous maintenance and updates

Standardization Body	Standard or Domain
OMG (Object Management Group)	UML, DDS, CORBA
W3C (World Wide Web Consortium)	HTML, CSS, JavaScript, Web APIs
AUTOSAR (AUTomotive Open System ARchitecture)	AUTOSAR Standard (Classic and Adaptive)

Living Standard

Standardization Seems Perfect, But Reality Is...

Long Development Time

- Needs careful planning and thorough checks
- Slow adaptation to market changes
- Delay in standard establishment

High Cost

- Extensive testing and compliance efforts
- Significant investment in documentation and integration

Compatibility Issues

- Differences in interpretation and implementation of standards
- Partial adoption of standards and addition of custom features
- Challenges with standard updates and version management

Case Study: Autosar Classic

1. **Standard Overload**: The standard has too many features due to various member needs, causing setup challenges and higher CPU usage.
2. **RTE Complexity**: AUTOSAR's Runtime Environment (RTE) is **as complex as a mini-OS**, leading to high resource consumption and management challenges.
3. **Integration Challenges**: Inconsistencies in standard interpretation among various solutions complicate integration.
4. **Tool Complexity**: Different, often GUI-based tools increase development complexity and error rates.

Case Study: DDS & SOME/IP

DDS

1. **First Standardized**: DDS was first standardized by the Object Management Group (OMG) in 2004
2. **Variety of Implementations**: A vibrant ecosystem with a mix of open source and commercial solutions
3. **Challenges**: High network traffic due to QoS complexity and performance inconsistencies in Wi-Fi environments
4. **Traditional Approach**: DDS uses a standard process via OMG that may respond slowly to fast-evolving tech and network changes

SOME/IP

1. **First Standardized**: SOME/IP was first introduced by BMW in 2011 and is now managed by the AUTOSAR standard
2. **Variety of Implementations**: One open-source implementation(vSOMEIP) and there are numerous commercial alternatives
3. **Performance Issues**: vSOMEIP is over 10 times slower compared to standard protocols like HTTP
4. **Standardization Barriers**: The standardization body is blocking new open-source implementations due to conflicts with member company interests.

Case Study: Autosar Adaptive

1. **First Standardized**: The first official release of Adaptive AUTOSAR was in 2017, followed by the development of various solutions by multiple companies.
2. **Standard Variability and Interpretation**: Differences in versions and interpretations of the standard among solutions lead to poor interoperability.
3. **Cost Barriers**: High costs of solutions deter adoption outside of China, where there is reluctance due to financial concerns.
4. **China's Government-Led Adoption**: The Adaptive AUTOSAR standard is adopted and promoted under the guidance of the Chinese government.

What Are The Solutions?

Standardization Through Open-Source

1. **Accurate Implementation and Documentation Efficiency**: Implemented using open source to prevent misinterpretations and reduce documentation efforts.
2. **Collaborative Development and Cost Savings**: Developed collaboratively by various stakeholders to enhance interoperability and share costs, reducing development and licensing expenses.
3. **Living Standard Benefits**: Continuously updated and refined by the community to keep the standard relevant and up-to-date.
4. **Functional Safety**: ISO 26262 now includes guidelines for using open-source software in functional safety through **ISO/PAS 8926:2024**.

Propose

Propose leveraging SOAFEE to create an open-source-based alternative solution to Adaptive AUTOSAR.

- ➔ **Cost Efficiency**
  Lower development and operational costs through open-source software.
- ➔ **Speed of Implementation**
  Faster standardization and deployment using open-source frameworks.
- ➔ **Living Standard**
  Implement ongoing enhancements

References

1. Eric S. Raymond (1997), The Cathedral and The Bazaar
2. Johannes Foufas (2023), The reality of AUTOSAR and the way forward
3. Manuel Hoffmann, Frank Nagle, Yanuo Zhou (2024), The Value of Open Source Software
4. https://en.wikipedia.org/wiki/Linux_kernel
5. ChatGPT 4 by OpenAI

Q&A