



A Propose and Design of Advanced Software Traceability Readability Opportunity Model

Gi-Hun Park, Ki-Won Song*

Department of Medical Engineering, Konyang University

ABSTRACT

This paper proposes ASTRO models that complement the problems of the existing S/W Stack Architecture. For the software stack architectures that are currently commonly used, it is not possible to provide much information about the increasing S/W complexity and progress in project development, which may lead to confusion in project progress. Therefore, this paper proposes ASTRO models that complement these problems and include all the information needed to develop S/W. For an intuitive understanding of the S/W project, the ASTRO Model visualizes and expresses IP information in more detail than the layers, subsystems, and existing Stack Architecture that are included in that S/W and uses different markers to easily determine whether the IP is an internal asset, an external asset, or an undecided one. In addition, this paper also implements systems for creating ASTRO models and managing projects. The system helps create ASTRO models and supports reuse of layers, subsystems, and IPs. It is also implemented to check detailed information such as resources allocated to each IP, core functions, defects and development period. Based on the current proposed model, future research plans to provide a reference model for each domain, reflecting the characteristics of various domains, and to develop Web-based ASTRO II considering the additional features, convenience and accessibility required for systems for integrated project management.

© 2019 KKITS All rights reserved

KEYWORDS : S/W architecture, Reuse, Readability, Traceability, Opportunity, Android architecture, Internal property, S/W project management, 3rd party, S/W development scope

ARTICLE INFO: Received 10 December 2018, Revised 21 January 2019, Accepted 8 February 2019.

*Corresponding author is with the Department of Medical Engineering, Konyang University, 158 Gwanjeodong-ro

Seo-gu Daejeon, 35365, KOREA.

E-mail address: kiwonsong@konyang.ac.kr

1. 서론

최근 4차 혁명 시대를 맞아 기존의 H/W중심의 시스템에서 S/W(Software)의 중심으로 IoT, Mobile, Big-data, AI등의 다양한 도메인에서 개발시스템의 S/W의 역할이 기하급수적으로 증가되고 있다.[1][2]

하지만, S/W의 특성상 가지고 있는 비가시성(invisibility)과 유연성(flexibility)으로 인하여 프로젝트에서의 개발 범위(Scope)과 관련된 서브시스템들의 단위(IP: Internal Property)를 관리함에 있어 S/W의 크기와 복잡도가 높아지면서 개발 역할과 책임의 모호성(Ambiguity)이 커지고 있다.[3][4]

현재 일반적으로 그려지는 S/W 스택 아키텍처의 경우는 앞서 서술한 내용대로 레이어와 서브시스템만을 표현하기 때문에 서브시스템의 상세한 IP를 확인하기 힘들다.[5] 또한, S/W의 복잡도 및 규모가 커지면서 그만큼 재사용성이 중요시되고 있고 그만큼 이전에 그려진 S/W 스택 아키텍처의 재사용도 필요해지고 있다.[6][7]

따라서, 본 논문에서는 S/W 스택 아키텍처의 단점을 보완한 ASTRO(Advanced S/W Traceability Readability Opportunity) Model을 제안한다.

2. 기 반 연 구

2.1 S/W Architecture

S/W Architecture는 거대한 시스템 개발을 진행할 때 필요한 모델로 개발파트의 확실하게 분리하여 각 파트의 협력 및 문제과약과 문제해결에 큰 도움이 된다. 또한, 현재 개발 중인 시스템을 이해하는데 도움이 되므로 신규인력 또는 시스템 설명에 있어도 적합하다.[8][9]

2.2 OSI Model

OSI(Open System Interconnection)모델은 상호 이질적인 네트워크 간의 연결 호환성 결여를 막기 위해 국제표준화기구(ISO)에서 제시한 표준모델이다. 해당 표준을 통해 여러 정보 통신 업체 장비들이 자신의 업체 장비 사이에서의 연결만 지원하고 다른 업체 장비와의 호환성을 지원하지 않는 문제를 해결하여 모든 시스템에 상호 연결이 가능하도록 만든다.[10][11]

OSI 모델은 <표 1>과 같이 7계층으로 되어 있으며, 1계층부터 Physical, Data Link, Network, Session, Presentation, Application 순서로 구성되어 있다.[12]

표 1. OSI 7 Layer
Table 1. OSI 7 Layer

7 Layer	Application	응용프로그램
6 Layer	Presentation	데이터 포맷 및 암호화
5 Layer	Session	연결유지
4 Layer	Transport	패킷 생성 및 오류 관리
3 Layer	Network	논리적 주소(IP) 관리
2 Layer	Data Link	물리적 주소(MAC) 관리
1 Layer	Physical	네트워크의 물리적 주소

2.3 TCP/IP Model

TCP/IP(Transmission Control Protocol / Internet Protocol)는 패킷 통신 방식의 인터넷 프로토콜 IP와 전송 조절 프로토콜인 TCP로 이루어져있는 네트워크통신 모델이다. 수행하는 역할로는 OSI와 같이 상호 이질적인 네트워크 간의 연결 호환성 결여를 막기 위해 사용되어 왔으며, OSI표준 이전부터 사용되어 왔다.

TCP/IP 모델은 4계층으로 이루어져 있으며, <표

2>와 같이 구성되어 있다.[13][14]

표 2. TCP/IP Model
Table 2. TCP/IP Model

4 Layer	Application	연결유지 및 데이터 포맷, 암호화, 응용프로그램
3 Layer	Transport	패킷 생성 및 오류 관리
2 Layer	Internet	논리적 주소(IP) 관리
1 Layer	Network Access	네트워크의 물리적 연결 및 주소(MAC)관리

TCP/IP모델과 OSI모델을 비교하면 <그림 1>과 같이 TCP/IP 모델에서 4계층은 OSI 모델의 5~7계층의 역할을 하며 3계층은 4계층, 2계층은 3계층, 1계층은 1,2계층의 역할을 수행한다.

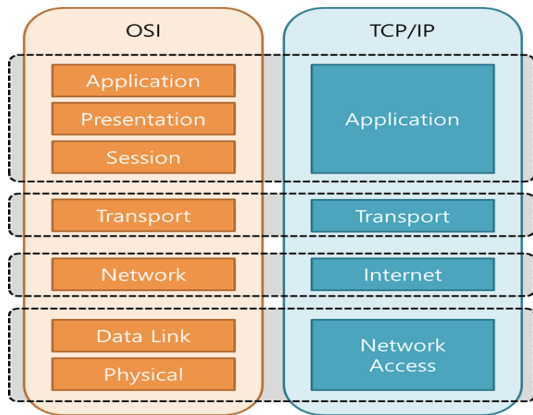


그림 1. OSI 7 Layer와 TCP/IP Model 비교
Figure 1. Comparing OSI 7 Layer and TCP/IP Model

이와 같이 두 가지의 네트워크 통신 모델이 Layer의 차이가 나기 때문에 프로젝트 진행시 혼동이 생겨 개발에 차질이 발생할 수 있다. 따라서 프로젝트의 가시화를 통해 개발 영역을 확실히 구분해줄 필요가 있다.

2.4 기존 Stack Architecture

기존의 구글 Android Stack Architecture의 경우 <그림 2>와 같이 UML(Unified Modeling Language)도 아니며 커다란 레이어 / 서비스 단위로만 제공하고 있다.[15]

설계시 제한된 UML을 이용하거나 S/W 스택 아키텍처(S/W Stack Architecture)는 개발하는 S/W에 포함되는 레이어(Layer)와 해당 레이어에 포함되는 하위 서브시스템(Sub-System)을 가시화시켜 표현해주는 구조도이다.

예를 들어 <그림 2>의 안드로이드 스택 아키텍처는 Libraries 레이어에서 멀티미디어에 관련된 내용이 Media Framework라고 서브시스템으로만 표현되어 있고 실질적으로 각 vendor들이 개발해야 할 세부 IP에 대한 설명이 부족하다.

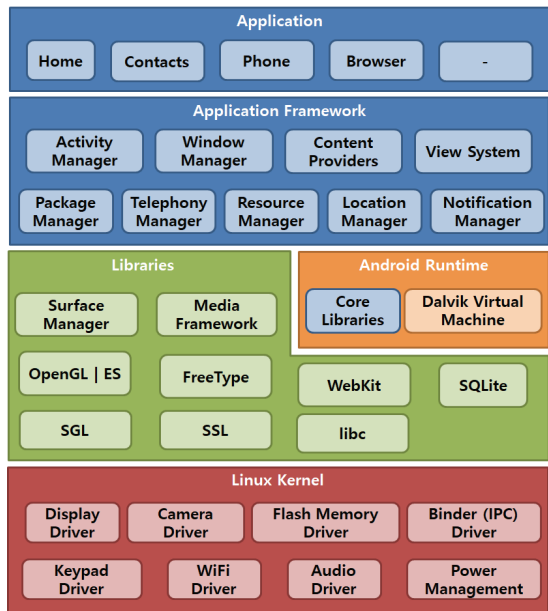


그림 2. Android Architecture
Figure 2. Android Architecture

프로젝트에 개발 및 관리에 있어서 개발범위를 가시화(Readability)하고 관련된 서비스간의 계층적 서비스의 추적성(Traceability) 개발현황(인력, 결함, 개발담당자, 3rd Party) 및 중요한 부분 중 하나는 설계과정이라고 할 수 있다.

이전의 Mobile AP(Application Processor) Feature phone 개발 시 개발하는 Chip vendor의 S/W 개발범위를 물어보면 대부분의 개발자들이 Device Driver 만 개발하기 때문에 BSP(Board Support Package) + α 를 개발한다고 개발범위에 대한 모호함을 보여 왔다.

그러나 현재 Smart Phone 개발의 경우 개발범위가 단순 BSP가 아닌 안드로이드 플랫폼을 기반으로 한 개발범위가 PSP(Platform Support Package)까지 늘어나면서 S/W 개발시 복잡도와 크기가 커지면서, 프로젝트관리(범위, 일정, 진척사항, 리소스, 결함현황 등)의 중요도가 높아지고 있다.

3. 소프트웨어의 가독성 및 추적성 제공 기법 고급(ASTRO)모델 제안

본 논문에서는 기존의 S/W Stack Architecture의 문제점을 보완한 ASTRO Model을 제안한다.

본 모델은 S/W 스택 아키텍처에서 각각의 서비스 시스템에 포함되어있는 IP까지 가시화하여 표현해 주며, 각각의 IP마다 key feature, defect, 개발기간 및 리소스 등을 확인할 수 있으며, 해당 IP가 다른 레이어의 어느 서비스 시스템에 포함되어있는지 추적하여 연관성을 확인 할 수 있다.

그리고 해당 Model를 자동으로 그려주고 각각의 IP가 다른 Layer나 Sub-System의 어느 파트와 연관되어 있는지 추적하는 기능 등을 지원해주는 Tool 또한 구현한다.

3.1 ASTRO Model 구조

본 논문에서 고급 소프트웨어 추적성 및 가독성 기법 모델인 ASTRO Model을 제안한다. ASTRO Model은 레이어와 각각의 레이어에 포함되어 있는 서비스 시스템, 그리고 서비스 시스템의 하위의 IP(재사용 단위)를 시각적으로 표현해주는 모델이며, 해당 IP의 분류(Internal IP, 3rd Party IP 등) 또한 시각적으로 표현해준다. 아래 <그림 3>은 Android 스택 아키텍처를 ASTRO Model로 표현한 것이다.

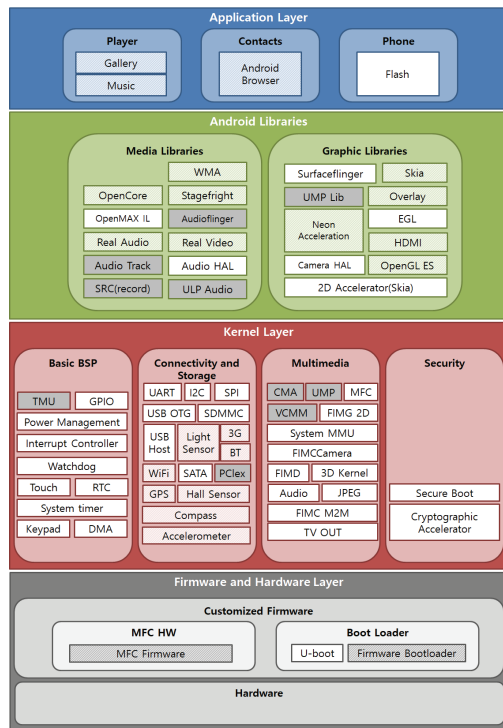


그림 3. ASTRO Model로 표현한 Android Architecture
Figure 3. Android Architecture in ASTRO Model

본 논문에서 제안한 ASTRO Model에서는 Android Libiraries 레이어에서 멀티미디어의 하위 IP 구성까지 가시화하여 표현해 주고 있다. 또한 각각의 IP가 Internal IP인지 3rd Party IP인지를 간단히 구분할 수 있도록 그려준다.

또한, 기존의 안드로이드 스택 아키텍처의 경우

각각의 서브시스템이 다른 레이어의 서브시스템과 연관되어 있는지 알아보기 힘들다. 따라서 ASTRO Model은 아래 <그림 4>와 같이 특정 IP가 다른 레이어나 서브시스템에 포함되어 있는지 표현해준다.

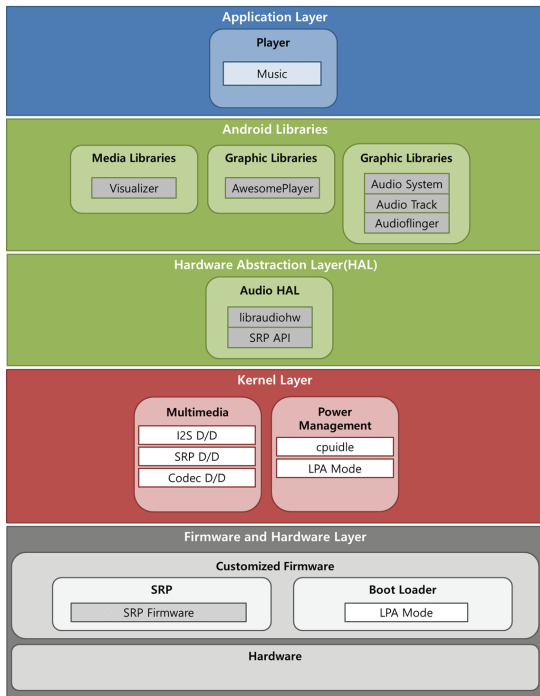


그림 4. ASTRO Model로 표현한 Android Low Power Audio IP
Figure 4. Android Low Power Audio IP in ASTRO Model

<그림 4>에서는 Android에서 Low Power Audio와 관련된 내용만을 ASTRO Model로 표현한 것이다.

3.2 ASTRO Model System

본 논문에서는 ASTRO Model을 구현하고 활용하기 위한 시스템을 구성한다. 해당 시스템의 구성도는 다음과 같다. 클라이언트(S/W Architecture Tool)에서 특정 프로젝트의 아키텍처를 생성하면 해당 아키텍처의 구성과 포함되어있는 모듈 정보를 서버로 업로드하여 저장한다. 그리고 다른 클라이언

트에서 해당 내용을 다운로드하여 내용을 추가 및 수정, 삭제를 한다. 서버에서는 수정내용에 대한 로그를 저장하여 다른 클라이언트에서 수정내용에 대한 확인이 가능하다.

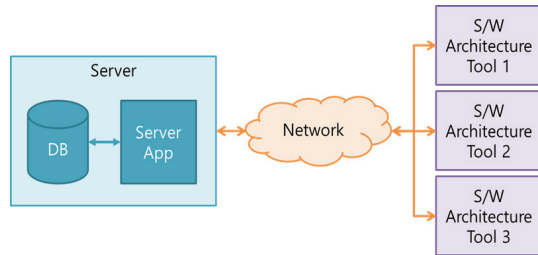


그림 5. ASTRO Model을 이용한 프로젝트 관리 시스템 구성도
Figure 5. Project Management System Diagram Using ASTRO Model

<그림 5>와 같이 시스템을 구성하여 S/W 개발 시 ASTRO Model 바탕으로 프로젝트의 전체 진행도를 확인할 수 있다. 또한, 이렇게 구현된 ASTRO Model이 서버에 저장되고 각각의 개발 파트에서 현재 진행사항이나 결함, 기한 등을 업데이트하여 프로젝트를 관리가 좀 더 수월해질 수 있다.

3.3 ASTRO Model 구현

본 논문에서는 제안한 ASTRO Model은 앞에서 언급한 안드로이드만이 아닌 다양한 단위와 범위를 수용하며 쉽게 작성할 수 있도록 구현하였다.

공통적이고 Hierarchical 한 일반적인 구조로 Layer, Subsystem, IP, 각 IP에 할당된 인력 및 Resource, 3rd Party와 내부 IP의 구별, 결함 현황, 개발 현황 등을 기본으로 <그림 6>과 같이 구성되어 있다.

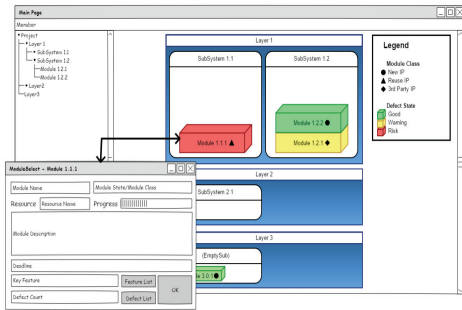


그림 6. ASTRO Model 작성 툴 (IP 정보 화면)
Figure 6. ASTRO Model Creation Tool (IP Information view)

ASTRO Model 작성툴에서는 ASTRO Model 가시화를 위해 아키텍처에 필요한 Layer, Sub-System, IP Module에 대한 템플릿을 제공해주며 진행 중인 프로젝트에 맞게 Layer와 Sub System, IP Module을 추가하여 자동적으로 배치한다.

각각의 IP Module은 개발 내용과 개발 진척도, 개발 기간, 할당된 인원(자원), Key Feature, Defect 등을 확인할 수 있는 기능을 <그림 7>과 같이 제공한다.

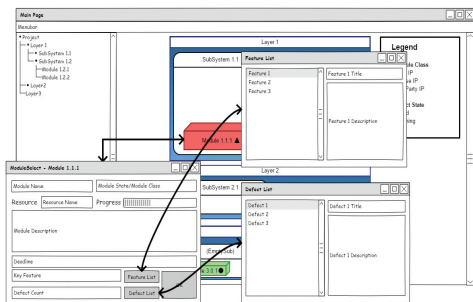


그림 7. ASTRO Model 작성 툴 (핵심기능 및 결함내용 화면)
Figure 7. ASTRO Model Creation Tool (Key Feature and Defect view)

<그림 7>에서처럼 하나의 IP Module에 대한 정보로 IP(Module)명, 현재 결함상태, 상위 서브시스템, 할당된 리소스, 진척도, IP에 대한 상세설명, 개

발 기간, 핵심기능(Key Feature), 결함내용을 제공한다.

4. 결론

기존의 스택 아키텍처의 경우에는 소프트웨어 개발 프로젝트 진행을 위한 충분한 내용이 포함되어 있지 않아 서로 개발영역에 있어 혼란이 올 수 있다. 따라서 본 논문에서는 IP기반 소프트웨어 자산 아키텍처 모델을 제안하여 해당 소프트웨어 프로젝트에 대해 직관적으로 이해 할 수 있도록 하였다. 또한 해당 모델을 작성, 관리 및 재사용을 위해 시스템을 구성하였고, 서버를 통해 현재 프로젝트 진행내용에 따라 IP에 할당된 자원과 개발기간, 결함관리 등이 가능하도록 구성하였다.

향후 연구에서는 현재 제안한 모델을 기반으로 다양한 도메인(모바일,의료,etc.)에 대해 도메인별 특성을 반영하여 프로젝트 착수 시 사용할 수 있는 아키텍처 레퍼런스 모델을 제공할 계획이며, 통합 프로젝트 관리를 위한 시스템(일정,품질,비용 등)에 필요한 프로젝트 재사용 및 수정 기능을 추가하고, 사용자의 편의성 및 접근성을 고려한 Web 기반 ASTRO II를 개발할 계획이다.

References

- [1] T. Hall, I. Biosciences, and C. Carlsbad. *BioEdit: an important software for molecular biology*, GERF Bull Biosci Vol.2 No.1, 2011.
- [2] A. I. Wasserman, *Software engineering issues for mobile application development*, Proceedings of the FSE/SDP workshop on Future of software engineering research. ACM, 2010.

[3] D. L. Parnas, P. C. Clements, and D. M. Weiss, *The modular structure of complex systems*, IEEE Transactions on software Engineering Vol.3, 1985.

[4] H. Zuse, *Software complexity*, NY, USA: Walter de Cruyter, 1991.

[5] J. Bosch, *Software architecture: The next step*, European Workshop on Software Architecture. Springer, Berlin, Heidelberg, 2004.

[6] M. L. Griss, *Software reuse: architecture, process and organization for business success*, tools. IEEE, 1998.

[7] R. Prieto-Diaz, *Implementing faceted classification for software reuse*, Communications of the ACM Vol.34 No.5, 1991.

[8] P. Clements, D. Garlan, L. Bass, J. Stafford, R. Nord, and R. Lvers, *Documenting software architectures: views and beyond*, Pearson Education, 2002.

[9] W. Royce, *Software project management*, Pearson Education, 1998.

[10] H. Zimmermann, *OSI reference model-The ISO model of architecture for open system interconnection*, IEEE Transactions on communications Vol. 28 No.4 pp. 425-432, 1980.

[11] B. Mann, and D. Duffy, *Local area network for digital data processing system*, U.S. Patent No.4, 1989.

[12] N. Briscoem, *understanding the OSI 7-layer model*, PC Network Advisor Vol.120 No.2, 2000.

[13] W. R. Stevens, *TCP/IP illustrated, volume 1: The protocols*, addison-Wesley, 2011.

[14] B. A. Forouzan, and S. C. Fegan, *TCP/IP protocol suit. McGraw-Hill Higher*

Education, 2002.

[15] Android Architecture, <https://developer.android.com/guide/platform/>, Oct. 2018.

고급 소프트웨어 가독성 및 추적성 기회 제공모델의 제안

박기훈¹, 송기원²

¹건양대학교 의료공학과 박사과정

²건양대학교 의료IT공학과 교수

요 약

본 논문에서 기존의 S/W Stack Architecture의 문제점을 보완한 ASTRO Model를 제안한다. 현재 흔히들 사용되고 있는 S/W 스택 아키텍처의 경우 점점 높아지는 S/W 복잡도와 프로젝트 개발 진행사항에 대해 많은 정보를 제공해 줄 수 없으며, 이에 따라 프로젝트 진행에 있어 혼동이 올 가능성이 있다. 따라서 이런 문제점을 보완하여 S/W 개발에 필요한 정보를 모두 포함한 ASTRO Model를 본 논문에서 제안하고 있다. ASTRO Model은 S/W 프로젝트를 직관적으로 이해할 수 있도록 해당 S/W에 포함되어 있는 레이어, 서브시스템 그리고 기존의 Stack Architecture보다 상세한 IP정보를 가시화하여 표현하며, 해당 IP가 내부 자산인지 외부자산인지 아니면 미정상상태인지를 쉽게 확인 할 수 있도록 각각 다른 표기를 사용한다. 또한, 본 논문에서는 ASTRO Model 작성하고 프로젝트를 관리하기 위한 시스템 또한 구현하고 있다. 이 시스템에서는 ASTRO Model 작성을 도와주며, 작성된 레이어나 서브시스템, IP의 재사용을 지원한다. 그리고, 각각의 IP에 할당된 리소스 및 핵심기능, 발생한 결함과 개발기간 등 자세한 정보를 확인 할 수 있도록 구현된다. 향후 연구에서는 현재 제안한 모델을 바탕으로 다양한 도메인의 특성을 반영하여 각 도메인별 아키텍처 레퍼런스 모델을 제공할 계획이며, 통합 프로젝트 관리를 위한 시스템에 필요한 추가적인 기능과 편의성 및 접근성을 고려한 Web 기반 ASTRO II를 개발할 계획이다.



Gi Hun Park received the bachelor's degree in the Department of Computer Engineering from the Konyang University in 2015. He received the M.S. degree in the Department of Medical Engineering from Konyang University in 2017, respectively. He has a Ph.D. course in the Department of Medical Information Technology at Konyang University since 2017.

E-mail address: ghpark86@gmail.com



Ki-Won Song received the bachelor's degree in computer science from Konyang University in 2002. He received the M.S degree and Ph.D degree in software engineering from Chung-Ang University in 2004 and 2007, respectively. He worked in Samsung Electronics from 2007 to 2014. He has been a professor of medical information technology engineering at Konyang University since 2015.

E-mail address: kiwonsong@konyang.ac.kr