

온라인 번역 플랫폼과 CI/CD를 활용한 국제화 프로세스 개선 아키텍처

*권오민, 남도하, 노현호, 여윤동, 최영락
경북대학교 컴퓨터학부

e-mail : {dhalsdl12, waroad, shgusgh12, yeo1998, ian.choi}@knu.ac.kr

Improved Internationalization Process Architecture Using Online Translation Platform and CI/CD Tool

*Oh-Min Kwon, Do-Ha Nam, Hyun-Ho Noh, Yoon-Dong Yeo, Yeongrak Choi
School of Computer Science and Engineering
Kyungpook University, Daegu, Korea

Abstract

As developers from various countries could collaborate through open source project, the importance of an internationalization process that translates open sources documents in English into various languages have emerged. Translations are made through open source platforms such as GitHub in the form of text such as Markdown, but there is a hassle of manually managing both original and translated text when translating. In this paper, we propose an ‘automated open source document internationalization process’ architecture using online translation platform and CI/CD tool, and implement this architecture with real open source online translation platform and GitHub Actions in English & Korean open source repository to confirm its applicability.

I. 서론

소프트웨어가 점차 복잡해지고 인터넷으로 인해 글로벌 협업이 증가하면서, 오픈 소스 프로젝트를 통해 여러 국가 개발자들이 협업하여 진행하는 추세는 더욱 뚜렷해지고 있으며, 글로벌 소프트웨어 문서를 코드로 관리하는 Docs as Code 방식에 따라 관리되는 많은 오픈 소스 소프트웨어를 GitHub에서 확인할 수 있다. 이때, 나라마다 언어와 문화가 다르므로 문서를 다국어로 번역하고 문화에 맞게 내용을 고려하는 국제화가 중요하지만, 이를 고려할 때 소프트웨어 문서를 관리하는 복잡도가 더욱 증가하여 모든 오픈 소스에서 문서에 대해서 국제화를 고려하지 못하고 있다. 실제로 클라우드를 대표하는 쿠버네티스 오픈 소스[1]의 경우, 모든 소프트웨어 문서를 Markdown이라는 포맷으로 텍스트로 관리하고 있으나, 다국어로 된 문서 또한 별

도의 Markdown 텍스트 파일로 관리하면서, 해당 문서와 원문에 대한 상관관계를 추적하는 과정이 점차 복잡해지는 상황이다.

따라서 본 논문에서는 오픈 소스로 된 국제화 플랫폼 및 오픈 소스에서 점차 도입이 증가하고 있는 지속적인 통합(Continuous Integration, CI) 및 배포(Continuous Deployment, CD) 방법론을 활용한 아키텍처 제안을 통해 더욱 효율적인 현지화와 국제화를 위한 개선 방안을 제안하고, 이를 필요로 하는 오픈 소스에 직접 적용하여 해당 아키텍처의 실제 적용 가능성을 확인해 보았다.

II. 본론

2.1 오픈 소스를 활용한 국제화 프로세스 아키텍처

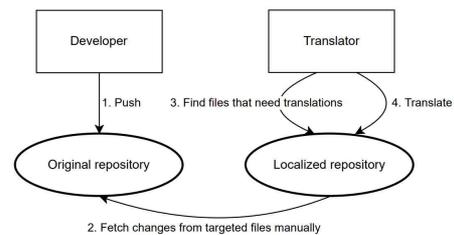


그림 1. 기존의 아키텍처 순서도

일반적으로 개별 언어에 대해 별도로 문서를 관리하는 오픈 소스의 경우, 대부분이 그림1과 같은 순서도를 따른다. (1) 개발자가 오픈 소스 원본 저장소에 새로운 커밋을 가하면, (2) 현지화된 저장소의 기여자가 해당 내용을 수동으로 가져온다. (3) 이후 번역가가 현지 저장소에서 번역이 필요한 부분을 직접 하나하나 찾아서, (4) 번역 후 저장소에 커밋을 하는 방식이다. 이때, 번역된 문서는 오픈 소스 저장소 관리 방식에 따라 같은 저장소 또는 다른 저장소가 될 수 있다.

위의 방식은 2가지 큰 단점이 있는데, 첫 번째로 원본 문서의 내용을 가져오는 것부터 해당 문서를 다국어로 번역하여 저장소에 적용하는 모든 과정이 수동으로 이루어지기에 지속적인 관리를 필요로 한다는 것이다. 두 번째로 번역하는 과정이 상당히 번거롭다. 번역가는 번역이 필요한 문서도 직접 찾아야 하며, 해당 저장소에 번역 내용을 저장하기 위해 convention도 찾아서 지켜야 한다. 위와 같은 단점들을 보완하기 위해 그림2의 새로운 아키텍처를 고안하고자 한다.

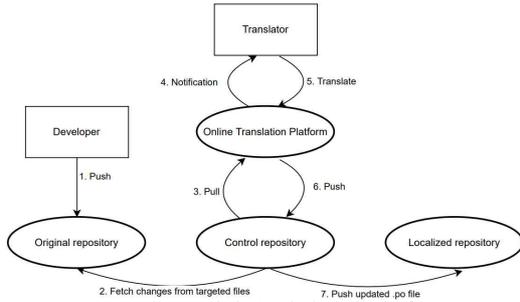


그림 2. 제안된 아키텍처 순서도

(1) 오픈 소스 원본 저장소에 새로운 내용이 추가되면, (2) Control repository에서 변경된 내용을 자동으로 감지하여 (3) 오픈 소스 번역 플랫폼에 업데이트 한다. (4) 해당 플랫폼을 통해 번역가에게 알림이 가능하며, (5) 번역가가 번역한 번역 문서를 (6) 해당 플랫폼에서 저장하여 동시 작업이 가능해지며, (7) 이 플랫폼 내에서 일정 퍼센트 이상 번역이 이루어지면 이를 오픈 소스 저장소에 convention에 맞게 Push 하여 최종 반영을 한다.

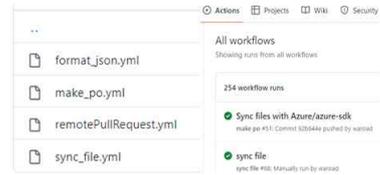
제안하는 아키텍처는 번역을 제외한 모든 것이 자동으로 이루어지기에 기여자는 번역에만 집중할 수 있는 장점을 준다. 그뿐만 아니라 오픈 소스 번역 플랫폼을 통해 번역이 이루어지는 모든 프로젝트는 사이트에서 확인 및 번역을 할 수 있기에, 더 많은 기여자의 참여를 유도할 수 있다는 장점이 있다[2].

2.2 제안된 아키텍처 구현

제안하는 아키텍처를 영어 문서 및 한글 번역 문서를 Markdown으로 따로 관리하는 저장소에 아래 오픈 소스 번역 플랫폼을 활용하여 구현하였다.

- 오픈 소스 번역 플랫폼 Weblate (<https://github.com/WeblateOrg/weblate>)
- 오픈 소스 문서 저장소: Azure 소프트웨어 개발자 키트 (SDK) 영문 및 한글 저장소
 - 영문 저장소: <https://github.com/Azure/azure-sdk>
 - 한글 저장소: <https://github.com/Azure/azure-sdk-korean>

Weblate는 다국어 문서를 PO라는 파일 단위로 관리하기 위해 Markdown과 PO 파일을 변환하는 mdpo라는 오픈 소스를 사용해 변환하였다. 또한, 지속적인 통합 및 배포를 위한 CI/CD는 GitHub에서 자체 지원하는 GitHub Actions를 사용하여 구현해보았다. 이때, 별도의 control repository를 사용하여 영문 및 한글이 저장된 오픈 소스 저장소의 의존성을 제거할 수 있었다.



```

1 {
2   "path": ["docs/policies/support.md", "docs/cpp/refs.md"],
3   "sync-repo": "Azure/azure-sdk",
4   "branch": "main"
5 }
  
```

그림 3. Github Action 및 CI/CD 결과, settings.json

그림3에 있는 4개의 Github Action으로 control repository 안에서 자동화를 구현하였다. 그림3의 settings.json에 지정된 원본 저장소에서 원문인 영어 문서를 가져와 싱크를 맞출 파일들을 지정해준다. 이를 format_json.yml에서 추후에 실행할 Github Action들의 입력값에 알맞게 정제 후 json을 새로 생성하고, sync_file.yml에서 주기적으로 해당 json을 참고하여 원본 저장소와 control repository를 싱크한다. 변경사항이 있으면 make_po.yml로 해당 Markdown 파일들의 PO 파일을 만들고, 이 PO 파일을 Weblate에서 자동으로 적용해 번역이 이루어진다. 마지막으로 remotePullRequest.yml이 Weblate에서 번역본을 가져와 localized repository에 pull request 한다.

III. 결론

기존 오픈 소스 문서를 수동으로 관리하는 국제화 방식 대신, 오픈 소스 번역 플랫폼 및 CI/CD 방법론을 도입한 새로운 자동화 아키텍처를 제안해 보았다. 기존의 언어별 저장소를 수동으로 관리하는 방법 대신, 제안한 새로운 아키텍처를 통해 번역을 제외한 모든 부분을 자동화하였으며, 이를 통해 기여자가 온전히 번역에 집중할 수 있는 환경을 만들어주고, 오픈 소스 번역 플랫폼을 통해 더 많은 기여자의 참여를 유도할 수 있다는 것을 제안한 아키텍처 구현을 통해 확인해 보았다.

제안하는 아키텍처는 GitHub Actions를 이용해 자동으로 번역이 필요한 내용 감지, 업데이트, 번역본 저장과 여기서 필요한 변수들은 다 settings.json으로 입력받기에, GitHub에서 관리되고 있는 대부분의 오픈 소스에 별도의 수정 없이 적용할 수 있다. 향후 한글 번역/문서화뿐만 아니라 국제화를 필요로 하는 보다 많은 언어에 활용되어 글로벌 개발자들이 오픈 소스 프로젝트에 협업할 때 유의미한 기여를 할 수 있을 것으로 기대된다.

참고문헌

1. Kubernetes 문서 저장소, <https://github.com/kubernetes/website>
2. Weblate all projects, <https://hosted.weblate.org/projects/>