Software Engineering and Infrastructure for Self-Driving Vehicle Development

Krainyak Andrey

ITRI Mechanical and Mechatronics Systems Research Labs Self-Driving Vehicle Dept. Deputy R&D Manager

摘要：開發龐大自駕車之軟體系統具有高度挑戰性，其需要穩定的軟體模組執行環境、強大的原始程式碼管理系統、原始程式品質驗證系統、分散式運算網路系統，以及建立工程師與任務管理者之間的高效溝通機制。本文將介紹工研院機械所自駕車團隊在軟體系統工程上的開發經驗。

Abstract：Self-driving vehicle software development is challenging. It requires a stable execution environment for software modules, a robust source code management system, a source code quality verification system, a distributed computation network making system, and establishment of an efficient communication between engineers as well as task management. This article is to shed some light on the aforementioned aspects of the software development processes.

關鍵詞：自駕車、原始碼管理、編碼式樣、盲打

Keywords：Self-Driving cars, Source code management, Branching, Code style, Touch typing

Introduction

Our team works on the development of the self-driving vehicle autonomy system. Most of the engineers are working on separate parts of the system. Every engineer has a different expertise, skills, as well as different understanding of the structure and capabilities of the system. Accordingly, extra work needs to be done to synchronize their work and integrate it into the system. The source code management system allows integration of the source code, as well as performing manual and automated source code quality checking. The task management system allows synchronization of working progress of the engineers.

While our fleet of vehicles is growing, every vehicle could have a custom sensor layout and hardware configuration. Thus, the system requires a scalable solution for being able to control each of the vehicles independently. The software system needs to be deployed on a broad range of platforms such as computers installed inside the vehicles and engineers' workstations. A unified interface must be developed for scalable integration of the system with different platforms. The computational power requirements of the system grow with every new submodule added to the system. This may require scaling of the computational power by installing extra computers into a vehicle. Running the system on multiple computers implies establishment of a networking between these computers, as well as providing the network transport layer between parts of the software modules distributed across the computers.

Having multiple engineers working on the software system requires declaring a code style that formalizes the ways of writing source code. This should simplify reading of the source code, which, in turn, simplifies maintenance and code review, as well as facilitates communication between engineers on the level of source code. Programming involves interaction with many tools. Engineers should learn how to use these tools efficiently. This will increase their productivity and let them focus on the implementation, releasing their minds from cognitive strain, induced by necessity to perform routine tasks. Engineers are humans. Human imperfection implies that there is a big part of the system that is neither standardized nor verbalized in the form of documentation. Integration of the system modules requires understanding of the functionality of the modules, hence, an efficient communication inside the team needs to be established. In this article, we shed some light on the aforementioned aspects of the software development processes and describe our approaches on software engineering for self-driving car development.

Source Code Management

Need for source code management is a well known subject [1]. Here are some advantages of having a version control system [2]:

- You won't lose your code by accident. Having a version control system, preferably with a remote service, will mean you are going to have another place where your code is stored. If several developers are working on the code simultaneously, then each one of them will have a copy of the entire code (or, in some cases, even the entire history).
- It allows you to keep historical versions of the code, for easy reverting, comparison and investigation. Let's say you introduced a bug. With a version control system you can easily revert to a previous version of the code where the bug was not present to verify that it did not exist there. Then you can diff the results, or even bisect the history to find the exact check-in that introduced this bug.
- It allows one to maintain several simultaneous lines of code (normally called "branches") and to easily compare between them and merge them.

There are a broad set of tools used for source code management. Our team uses Git for managing source code. The Git server is provided by our own instance of Atlassian Bitbucket Server [3].

Branching

In the repository of the self-driving car runtime system, we have three types of branches: master branch, release branches, feature branches. Master branch is used for storing long-term stable changes. Changes to this branch are accepted only with a pull request. The master branch can’t be rebased. Release branch is used for integration of code related to a particular milestone. Every engineer can commit code directly to this branch without a pull request. A release branch can’t be rebased. After the related milestone has been accomplished, the release branch is being rebased and merged into the master branch. Feature branch represents code developed usually by a single engineer and related to a single task. A feature branch is usually created on top of the master or release branch. It can be freely rebased, for example, to pick up changes from a release branch.

Build Automation

Working on a big project implies involvement of many developers and deployment of the system on multiple environments. When a developer works on a module, the developer may build only a part of the whole codebase. After the code is pushed to the source code management server, it may be picked up by other developers. If the developer, who pushed the

更完整的內容

詳見 ▌機械工業雜誌▌ • 445 期 • 109 年 4 月號

機械工業雜誌·每期 **220** 元·一年 12 期 **2200** 元

線上訂購網址：https://www.automan.tw/magazine/orderMag.aspx

▌付款方式▌

1. 郵局劃撥─戶名：財團法人工業技術研究院機械所　帳號：07188562
   請於劃撥單的通訊欄寫明：購買期數、金額等
2. 匯款資料─兆豐國際商業銀行新竹分行(代號 017)
   帳號：203-07-02288-0　戶名：財團法人工業技術研究院
3. 信 用 卡─請填寫信用卡訂購單

麻煩您將**繳款收執**或**信用卡刷卡單**傳真至(03)582-2011，我們會盡快處理您的訂單並開通權限，再次感謝您的支持與愛護。

訂書專線：03-591-9339　　　傳　　真：03-582-2011

機械工業雜誌·官方網站：www.automan.tw　機械工業雜誌·信箱：jmi@itri.org.tw

機械工業雜誌 優/惠/訂/購/單

訂閱一年 12 期

$ 2200 / 續訂戶 $ 2000

好禮二選一

A 史欽泰墨寶帆布袋

B 工研院機械所無人車USB (8G)

訂閱紙本+電子雜誌

$ 3000 ~~原價 $ 4400~~

一年12期

贈送

A 史欽泰墨寶帆布袋

訂閱二年 24 期

$ 4000 / 續訂戶 $ 3600

好禮四選二

A 史欽泰墨寶帆布袋

B 工研院機械所無人車USB (8G)

C 工具機叢書任一本

D 智慧機械人叢書任一本

限量專屬精品送給您

A

B

C

D