Exploring Problems and Solutions About Launch Files in ROS from Q&A Community

Guangjun Jiang[2], Xinjun Mao[1,2*]

[1] College of Computer, National University of Defense Technology, Changsha, China
[2] Key Laboratory of Software Engineering for Complex Systems, National University of Defense Technology, Changsha, China

Abstract

Robot Operating System (ROS) is a widely used software architecture for robotic systems, which provides hardware abstractions and common functions in robotics. Launch files, as one of the configuration files in the actual program development of ROS, provide a method to start multiple nodes and set global parameters on the local or multiple machines at the same time. For launch files with complex and complex content, we need to get the questions we encounter in writing and using launch files from the Q&A community. We crawled 854 Q&A posts about Launch files from SO and ROS Answers respectively, and conducted in-depth empirical research. Based on the content of the Q&A posts, we roughly divided them into five topics, including syntax specification, function implementation, running error, dependency related and environment configuration. We used topic modelling LDA to categorize Q&A posts by topics, and segmented them by combining the five topics already obtained. Through the results, we found that parameter configuration and syntax errors were the main problems, and then we made a rough analysis of the answer content of the post, and found that the quality of the ROS-related Q&A posts was not high. The above findings inspired us to develop a template tool to simplify the writing of Launch flies and use the node information in the Launch files to recommend ROS nodes to lay a good foundation.

Keywords

Launch files、ROS、Empirical Study、Q&A Community、Topic Modelling

1. Introduction

With the rapid development of the field of robotics, robot development with ROS as the core attracts more and more attention. ROS is an open source operating system widely used in robot development[1]. It can provide developers with many functions such as hardware abstraction, device drivers, library functions, visualization, message passing, and package management. It provides the services expected of an operating system, including hardware abstraction, low-level device control, execution of common functions, message passing between processes, and package management. The official website of ROS also provides various supporting documents, and related resources form a powerful ecosystem that makes learning and using ROS very convenient[2]. Because robotic software is domain-specific and highly platform-dependent, robotic software development faces a series of challenges[3]: robotic software development requires developers to have both robotics domain knowledge and software development capabilities, and to be able to transform existing programs for different platforms.

ROS projects usually include many files, such as source code, configuration files, message files, etc. Launch files, as a configuration file, are used to facilitate launching multiple nodes at the same time[4]. Launch files are a startup configuration file in xml syntax with the extension .launch. If you want to start multiple nodes for a robot, this means opening corresponding multiple terminals. Starting each node one by one with the *rosrun* command in the terminal is cumbersome and error-prone. The above

ICBASE2022@3rd International Conference on Big Data & Artificial Intelligence & Software Engineering, October 21-23, 2022, Guangzhou, China
EMAIL: jgj16@nudt.edu.cn (Guangjun Jiang); xjmao@nudt.edu.cn (Xinjun Mao)

problems can be effectively solved by Launch files. We write all nodes in a Launch file based on XML format, use the *roslaunch* command to parse this file, and then easily start all nodes. The *roslaunch* command will automatically start ROS nodes Manage nodes and parameter servers, so we don't need to start *roscore* separately[5]. If Launch files are used, then all operations will be completed in one command, improving the development efficiency of ROS developers. It includes many types of tags, and different tags correspond to different functions. Table 1 clearly shows the relationship between them.

TABLE 1: Labels and functions of Launch files.

Label	Function
<launch>	root tag
<node>	start node
<include>	nested
<remap>	rename
<param>	global parameters
<rosparam>	load parameters
<group>	node grouping
<arg>	local parameters
<machine>	remote tag
<env>	environment variable
<test>	test node

General steps to write a Launch file: (1) Set default values for command line parameters. (2) Through the label, set the inclusion relationship of the Launch file. (3) Set node information, including name, namespace, parameter. (4) Set the remap as needed, then set the remap relationship.

In the actual ROS development process, various problems will be encountered when writing and using Launch files, which requires the study of common problems and solutions[6]. As different versions of ROS robots are iterated, upgraded, bringing new concepts, techniques and tools from software engineering and distributed systems, this is reflected in the rapid increase in discussions on this topic on Q&A sites such as Stack Overflow, and at the same time, we combined with ROS Answers, it is also a Q&A website. Its format is the same as Stack Overflow, and its content is related to ROS development. With the development of ROS, its community activity is also developing rapidly. We can see this trend from Figure 1. As a relatively small field, the use and writing of Launch files is very critical to improving the efficiency of ROS development. There are relatively few studies on Launch files. Estefo et al. [7] analyzed the code repetition in Launch files and found that 25% of them have more. The packages of two startup files have duplicated code, and the clones are very similar, quantifying the code quality of the ROS ecosystem for the first time.

We use LDA for topic modeling, statistical analysis of topic representations, and an in-depth exploration of our findings. We aim to analyze issues encountered by Launch file developers and corresponding solutions from SO and ROS Answers. Also, analyze and summarize solutions to different problems.

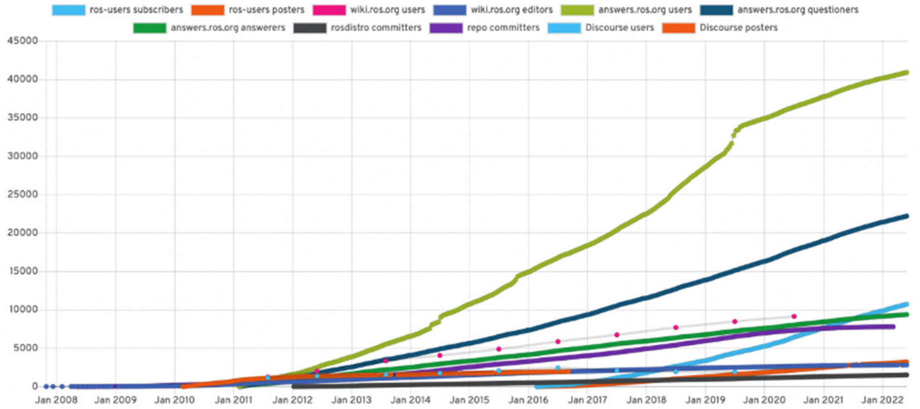

Figure 1: The number of users in the ROS community over time.

Therefore, this paper presents an analysis that aims at answering three research questions: **RQ1**: What are the problems related to Launch files in the SO and ROS Answers communities? **RQ2**: What are the corresponding solutions for different types of problems? **RQ3**: How to combine Launch files to improve the development efficiency of ROS developers? The contributions of this paper are twofold: (1) problems discovery and classification of Launch files in the Q&A community; (2) solutions for different problems.

2. Background and Definitions

Although Launch files are only a small part of the ROS files, but in the analysis of the ROS-related Q&A posts in SO, there are as many as 2954 Q&A posts related to ROS, of which there are 562 Q&A posts about the Launch files, accounting for 19.03 %[8], we can see the importance of Launch files for ROS development. The Q&A posts about Launch files in SO are usually about configuration environment, running errors, etc. But in our actual development process, about the writing specifications and syntax of Launch files, there are requirements for implementing a certain function with Launch files. Also many, in ROS Answers we can find posts on related topics.

In SO and ROS Answers, a post can have many tags to define, only if it is related to Launch files, we can consider as **Launch files-related**, but many **Launch files-tagged** posts and their content has nothing to do with Launch files[9]. It requires we manually remove such posts. At the same time, based on the development experience of ROS, we have summarized five topics of problems about Launch files.

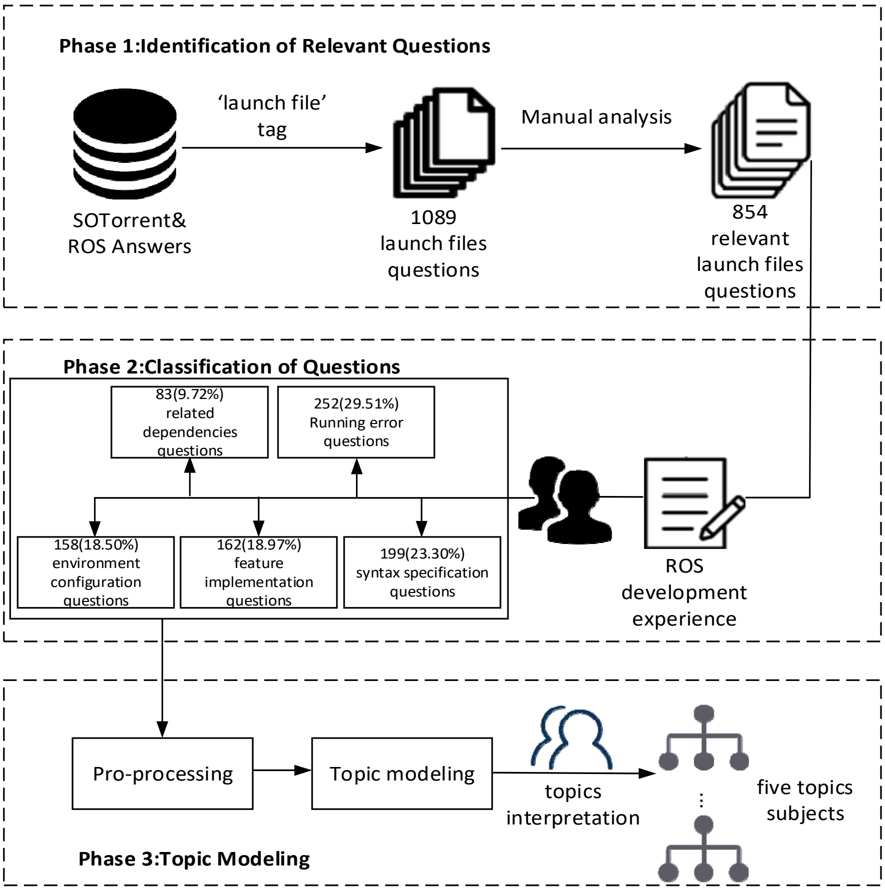
Phase 1:Identification of Relevant Questions
'launch file' tag
Manual analysis
SOTorrent& ROS Answers
1089 launch files questions
854 relevant launch files questions
Phase 2:Classification of Questions
83(9.72%) related dependencies questions
252(29.51%) Running error questions
158(18.50%) environment configuration questions
162(18.97%) feature implementation questions
199(23.30%) syntax specification questions
ROS development experience
Pro-processing
Topic modeling
topics interpretation
five topics subjects
Phase 3:Topic Modeling

Figure 2: Framework for extracting Launch files questions from the Q&A community. Phase 1 crawls the questions, phase 2 manually divides the questions into five categories, phase 3 employs topic modelling to group the questions into subjects.

3. Empirical Study Design

Our goal is to study questions related to Launch files on the Q&A community. From the obtained results, we can analyze the solutions for different problems in the answer.Therefore, we ask three research questions:

RQ1*: What are the problems related to Launch files in the SO and ROS Answers communities?*

RQ2*: What are the corresponding solutions for different types of problems?*

RQ3*: How to combine Launch files to improve the development efficiency of ROS developers?*

3.1. Data Collection

We downloaded all publicly accessible data (as of July 2022) in SOTorrent[10], and selected all SO questions and their answers, and utilized the scrapy framework to scrape relevant Q&A posts from the ROS Answer web page.

The framework we used to identify discussions related to Launch files from Q&A posts is shown in Figure 2. The framework consists of two phases. In the first stage, we retrieved 1089 related questions from the Q&A posts with launch file as the tag. At the same time, we also retrieved with tagged 'xml' and 'launch'. Because many Launch files-tagged posts are not related to the Launch files, we manually screened them one by one, and finally got 854 Launch files-related posts.

3.2. Classification of Launch files-related Questions

The second part of our experiment is mainly to initially classify 854 posts, which requires certain ROS development experience to guide, we need to have a deep and clear understanding of the writing and use of Launch files. We finally determined five categories of problems in Launch files, including syntax specification, function implementation, operation Errors, dependencies, environment configuration, which have a good reference for the later classification with LDA. The five categories nicely contain all the posts.

3.3. Topic modeling.

Data pre-processing. Firstly, We remove HTML tags, code snippets, URLs and stop words[11]. Secondly, to minimize noise, we use the initial set of *NLTK* stop words [12]to build a list of stop words specific to this study, we also lemmatize the document collection of SpaCy [13]. Finally, we remove the words that appear in more than 80% and less than 2% of the documents to get the corpus.

Topic modeling. Latent Dirichlet assignment (LDA) [14] is a traditional topic modeling model. The LDA model is a very basic and well-known model in NLP. The essence of the LDA model is a generative Bayesian probability model containing a three-layer structure of words, topics and corpora. We used LDA and the mallet tool [15]. The LDA settings are configured according to the guideline [16], followed by empirical experiments on our dataset. By constantly modifying the values of K and I_r and randomly selecting 50 posts for testing, in the end, we used K=18 topics, because each topic is described by 10 words, but these words cannot represent each topic clearly, Therefore, we combined the ROS development experience to analyze the descriptions of these words and get the corresponding topics. We got 18 subjects, combined with the previous 5 categories, we included these 18 subjects, and we can clearly see the proportion of different subjects. From it, we can see the proportion of different question posts in the total, and we can also see the proportion of subjects under the same category.

4. Empirical Study Results

- ***RQ1****: What are the problems related to Launch files in the SO and ROS Answers communities?*

According to the 18 subjects obtained by the IDA model, we include them in five categories. According to the results of figure 3, we can clearly see that the parameter configuration in the syntax specification and the wrong file syntax in the runtime error account for the largest proportion. This shows that when we write and use Launch files, there are many problems in the use of parameter configuration and internal syntax. In fact, this is also the key to the use of Launch files. Because the Launch files themselves are a kind of configuration files, there are many labels and parameters that need to be configured in the file. The node label is more important in the Launch files, and the parameter information in it is related to whether the node is running normally.

In other categories, we see that many questioners ask questions without thinking carefully after encountering problems. For example, the running command is wrong, which is a relatively simple error.

- ***RQ2****: What are the corresponding solutions for different types of problems?*

The main problems with Launch files are syntax problems and parameter configuration problems. In the corresponding answers, I removed the unanswered questions. In the answered posts, I removed the posts with only one answer and not accepted by the questioner, which to some extent guarantees the quality of the answer. For the answers to syntax questions and parameter configuration, most of them give similar correct examples for the questioner's reference, and few really give correct and operational answers.

For the problem of function implementation, usually give the answers are relatively clear, but the answers given by the respondents are not very clear to the functionally complex questions raised by some people. In the question of syntax specification, many answers give more practice and give the writing specification of launch files. In SO, because there are many questions about environment configuration, and there are few satisfactory answers. For a novice ROS developer, I recommend going to ROS Answers to search and learn more.

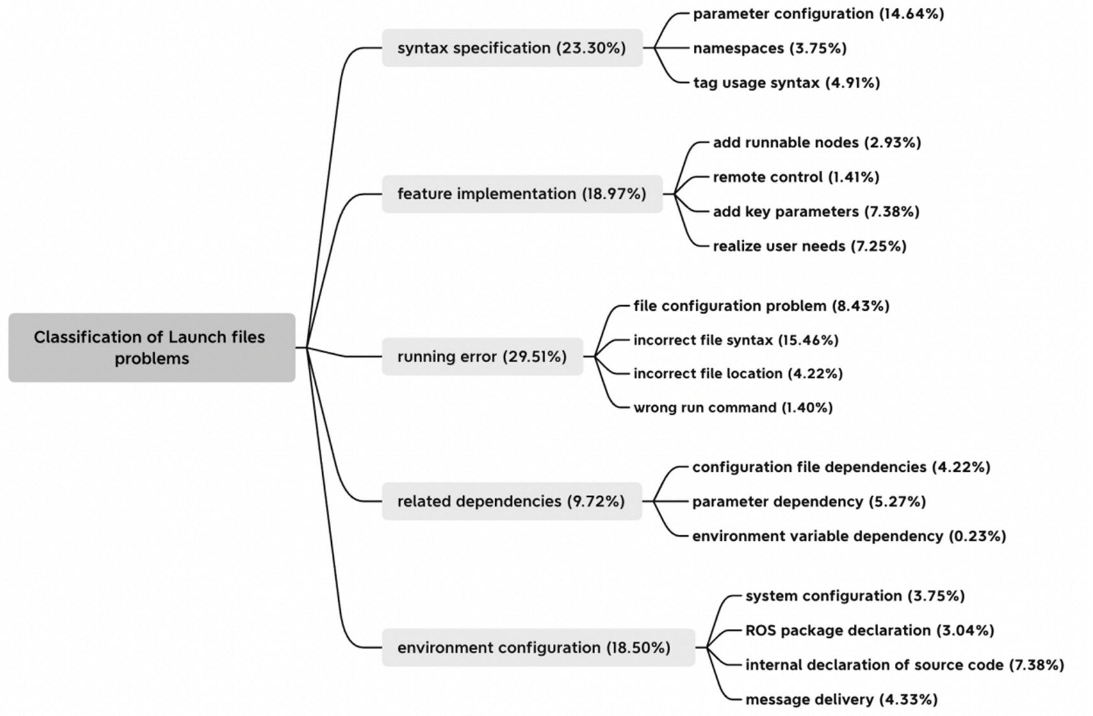
Classification of Launch files problems
syntax specification (23.30%)
parameter configuration (14.64%)
namespaces (3.75%)
tag usage syntax (4.91%)
feature implementation (18.97%)
add runnable nodes (2.93%)
remote control (1.41%)
add key parameters (7.38%)
realize user needs (7.25%)
running error (29.51%)
file configuration problem (8.43%)
incorrect file syntax (15.46%)
incorrect file location (4.22%)
wrong run command (1.40%)
related dependencies (9.72%)
configuration file dependencies (4.22%)
parameter dependency (5.27%)
environment variable dependency (0.23%)
environment configuration (18.50%)
system configuration (3.75%)
ROS package declaration (3.04%)
internal declaration of source code (7.38%)
message delivery (4.33%)

Figure 3: Categorization of Launch files-related problems.

- ***RQ3****: How to combine Launch files to improve the development efficiency of ROS developers?*

In response to the troublesome configuration of Launch files parameters, we can recommend template-based parameters based on some context-related parameter variables.

According to the nodes in Launch files are related, because they may be in a launch file to complete a certain task together, and the node name recommendation in the process of writing Launch files can be realized by combining the source code and probabilistic dependencies. Overall, community users are not very active in the answers to ROS-related Q&A posts, and may be closely related to the fact that there are not many people engaged in ROS development.

5. Conclusion

This article analyzes issues related to launch files on the Q&A community. We divided the 854 Launch files into five categories, including syntax specification, function implementation, operation Errors, dependencies, environment configuration and classified them using a topic modeling approach, resulting in 18 subjects. The results show that the wrong syntax in the runtime error of the Launch files and the parameter configuration in the syntax specification are the two main problems. At the same time, we did a rough analysis of the responses to some of the posts.

As future work, we will conduct empirical studies on other ROS profiles to study the impact of the current ROS ecosystem on ROS development, and intend to share and discuss our findings with the ROS developer community to validate and enrich our research. In the future, we will continue our work and further explore and research the problems existing in ROS in conjunction with the Q&A community.

6. References

[1] Cashmore M, Fox M, Long D, et al. Rosplan: Planning in the robot operating system.Proceedings of the International Conference on Automated Planning and Scheduling. 2015, pp.333-341.

[2] W. Curran, T. Thornton, B. Arvey, and W. D. Smart. Evaluating impact in the ros ecosystem, International Conference on Robotics and Automation, 2015, pp.6213–6219.

[3] Gerkey B, Vaughan R T, Howard A. The player/stage project: Tools for multi-robot and distributed sensor systems.Proceedings of the 11th international conference on advanced robotics. 2003, pp.317-323.

[4] Quigley, M., Conley, K., Gerkey, B., FAust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Mg, A., 2009. ROS: an open-source Robot Operating System.

[5] Estefó P, Robbes R, Fabry J. Code duplication in ros launchfiles. 2015 34th International Conference of the Chilean Computer Science Society (SCCC). IEEE, 2015, pp.1-6.

[6] S. Baltes, C. Treude, and S. Diehl. Sotorrent: Studying the origin, evolution, and usage of stack overflow code snippets, in Proceedings of the 16th International Conference on Mining Software Repositories (MSR 2019), 2019, pp.46-47.

[7] Wilbur W J ， Sirotkin K . The automatic identification of stop words. Journal of Information Science, 1992, 18(1), pp.45-55.

[8] M. Honnibal and I. Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing, 2017.

[9] Daud A ， Li J ， Zhou L , et al. Latent Dirichlet allocation (LDA) and topic modeling: models, applications, future challenges, a survey. Frontiers of Computer Science in China, 2010, 4(2).

[10] Moore A ， Sithole M . 1. The tools: Tool-making skills and equipment; Materials for tool-making，Mallet[M]. 1992.

[11] Chen L F ， Liao H ， Ko M T , et al. A new LDA-based face recognition system which can solve the small sample size problem. Pattern Recognition, 2000, 33(10), pp.1713-1726.

[12] Mens, T., Adams, B., Marsan, J. Towards an interdisciplinary, socio-technical analysis of software ecosystem health. In: CEUR Workshop Proceedings, 2017, pp.7–9.

[13] Mateos-Garcia, J., Steinmueller, W.E. The institutions of open source soft-ware: examining the Debian community. Inf. Econ. 2008, Policy 20 (4), pp.333–344.

[14] Mao Xin-Jun，Yang Shuo，Huang Yu-Hong，Wang Shuo. Towards software architecture and accompanying behavior mechanism of autonomous robotic control software based on multi-agent system．Journal of Software，2020，31(6), pp.1619-1637.

[15] Storey, M.-A., Zagalsky, A. Disrupting developer productivity one bot at a time. In: Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering. ACM, New York, NY, USA, 2016, pp.928–931.

[16] Wang, X., Huang, C., Yao, L. A survey on expert recommendation in com- munity question answering. J. Comput. Sci. Technol. 2018,33 (1), pp.1–29.