

# The Research of a new Software Framework for cross-platform development Field

Wenfeng Lin<sup>1,a</sup>, Gaoxuan Zhang<sup>1</sup>, Zhoucan He<sup>1</sup>, Xinyu Liu<sup>1</sup> and Wenchao Dang<sup>1</sup>

<sup>1</sup>Institute of Electronic Engineering, China Academy of Engineering Physics, Mianyang 621900, China

**Abstract.** Software framework is a very important approach to realize software reuse and improve the software quality. In this paper, we firstly introduce a new software framework named QFusion to complete cross-platform desktop software development, which is based on function plug-in module. The QFusion framework can realize information interaction by use of event bus, data bus and service, which can separate the graphical user interfaces (GUI) from data processing completely. Based on the framework-related develop process, the complexity is largely reduced. Secondly, we give the practice of QFusion framework in modern military field to implement the feature extraction and data processing algorithm. Benefit from the QFusion framework, the signal processing task is completed by the collaboration among three groups, including data generation plug-in development group, data processing plug-in development group and GUI plug-in development group, respectively. Finally, the signal processing results are presented to prove the validity of the QFusion framework, which lays a solid foundation for solving the practical problem in engineering application.

## 1 Introduction

With the development of software application in different kinds of domains, the software environment is becoming more and more diversified and complicated. At the same time, the software functions need to continuously strengthen and the scalability of the software becomes increasingly important because that the requirements of the users varies [1]. Traditional software develop process is difficult to satisfy the different demands such as the collaboration among different groups, the continuously increasing of the software scale, the complicated logic of the operation and the anfractuous function module. In face of the continuously changing environment and user's requirements, we need an efficient method to software construction. It is thus necessary to write a modular code with a predefined data flow from the very beginning. Such a program modularization and the definition of a corresponding data flow are the essential elements provided by a software framework [2]. Software framework is an important approach to large-grained software reuse. In order to realize the design reuse and code reuse of the software in specific domain, kinds of different software framework appeared [3-7].

Generally speaking, software framework provides templates and services at multiple levels of abstraction for the construction of software applications. That is to say, the framework is including a set of rules, interfaces and services provided to the programmer, who can use it to perform a set of tasks [8]. Mostly, the user's requirement is simply to implement or change a single

algorithm, not need changing the rest of the reconstruction software [9]. The programming logic is predefined in the framework, which makes it easier to extend. Based on the framework-related develop process, the programmer only needs to implement algorithms, whereas the framework is to specify how these algorithms interact [10]. The programmers do not need to know the every detail of a particular program to be able to extend it by using of software framework [11-12].

The rest of this paper is organized as follows. Section II we propose a new software framework called QFusion, which is used for cross-platform desktop software development. Based on the idea of layered to create a general software framework and using a configuration file to realize the logical relationship of software structure, QFusion can implement the separation of software framework from business logic. QFusion framework can create the object with the configuration file to complete dynamic construction of the software system, which can effectively avoid the logic of the software structure to be fixed in the software implementation. Benefited from the QFusion framework, the complexity is largely reduced. Section III we give the practice of QFusion framework to implement the feature extraction and data processing algorithm for telemetry data in modern military field. The results are presented to prove the validity of the QFusion framework.

## 2 Framework of QFusion

<sup>a</sup>Corresponding author: [linwenfeng\\_job@163.com](mailto:linwenfeng_job@163.com)

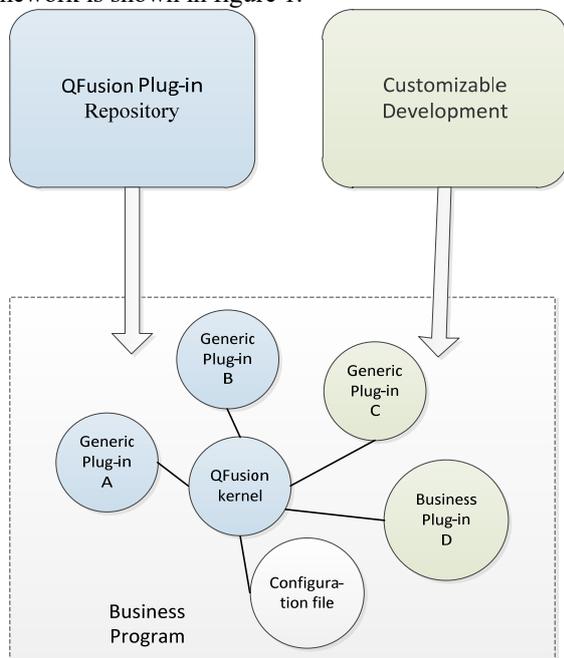
## 2.1 Requirements Analysis

In the subdivision business domain, the function of a large number of business software is similar, but it is difficult to integrate, which not only causes technical risks, but also has great difficulty in code version management and functional maintenance. In order to improve the production efficiency, reduce the cost of software, achieve software pedigree and product line, we have developed a set of software middleware system for rapid development and reconstruction of subdivision business domain software. We refer to this system as QFusion Fast Business Building Framework Platform, which is called QFusion framework for short.

## 2.2 The overview of QFUSION

QFusion framework is mainly used for desktop software development, which uses plug-in development model to provide software product line integrated development platform, business plug-ins and business-independent graphics, tables, trees and other generic display plug-ins for the subdivision business domain.

The QFusion framework uses the kernel and framework structure to build applications based on the event bus. The framework bus provides the basic services. The plug-in implements the specific business functions, and joins the framework with the standardized interface. Multiple plug-ins are assembled into a complete application through a configuration file. The plug-in business program development based on the QFusion framework is shown in figure 1.

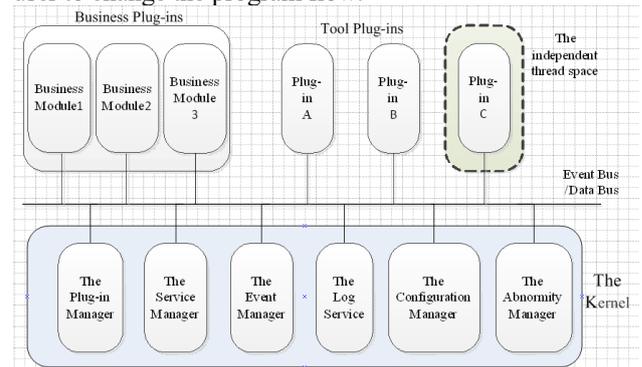


**Fig. 1.** The plug-in business program development based on the QFusion framework

The QFusion framework completely separates the interface from the backend services, and the plug-ins interact with each other through the bus mechanism, which is shown in figure 2. This design pattern enables the logical separation of presentation and processing, rapid configuration and change of software functions,

and effectively reduce the complexity of business, enabling multi-person collaborative development and binary multiplexing of business plug-ins.

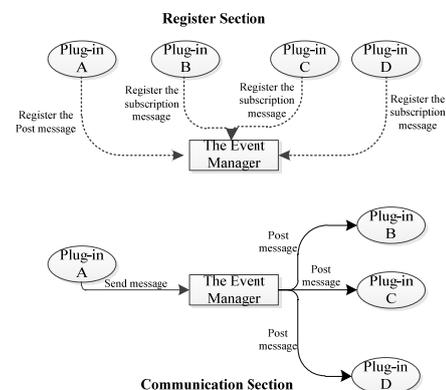
The QFusion framework saves the developer a lot of time in learning how to use a particular software package every time she/he wants to perform a new task on a set of data. All algorithms are implemented in modules having the exact same structure. This makes the code easier to read and understand. The software user can thus start to work on business problems with a significantly shorter learning period. A modular structure also leads to more flexibility. Systematic analyses can be performed more easily, as the QFusion framework explicitly allows the user to change the program flow.



**Fig. 2.** The bus mechanism in QFusion framework

## 2.3 The principle of QFUSION

In order to design the application as a loosely coupled system, the framework references the message-oriented middleware model, uses the event bus as the hub, and maintains plug-in communication through the subscription/publishing model. In the subscription/publishing model, the plug-in firstly registers with the event manager. The plug-in that posts the message simply sends the message out thematically and the event manager is responsible for passing the message to all plug-ins subscribing to the topic. The subscription/publishing model is shown in figure



**Fig. 3.** The subscription/publishing model

The QFusion framework defines each business module as a plug-in for a specific function, and the plug-in communicates with the outside through the registered interface. Plug-ins form a complete application through

collaboration. All services based on the framework run in the same process space. For asynchronous requirements, a single service can work in a separate thread. This architecture can meet both synchronous and asynchronous communication between plug-ins. Communication between plug-ins is carried out by event manager.

## 2.4 The features of QFUSION

### 1) Cross-platform

The QFusion framework can support for generic platforms such as Windows, Linux, Unix, Mac, and the NeoKylin.

### 2) Customizable

The QFusion framework does not rely on any closed-source third-party technologies, tools or code, and can be customized locally according to the platform, scenarios and the needs of users to meet different business requirements.

### 3) Plug-in

By use of plug-in, the complexity of business logic and the risk of project delivery are reduced by functional decomposition of the system. Each plug-in directly uses interface, bus and other ways to interact, which can reduce the complexity of business logic. Through the plug-in packaging of basic and universal business functions, it is not necessary to develop code to achieve binary multiplexing. The plug-in supports hot-plugging functions, which can be added or deleted dynamically at run time as needed.

### 4) Business-atomized

Business atomization refers to the separation of complex business functions into simple basic services and the use of plug-ins to complete independent, simple basic business functions. Then the basic business plug-in is integrated into different complex business processing plug-in sets. Service atomization is helpful to improve the reuse rate of services and is the core value of the QFusion framework.

### 5) Development normalization

By making plug-in development specifications, such as plug-in requirement specification, naming conventions, plug-in interface specification, and plug-in interaction specification, to constrain the behavior of developers, and gradually lead developers to form a modular, plug-in

development thinking, so that every developer can use to deal with problems as the exact same way.

### 6) Testability

The plug-in can be measured well and can be tested according to the interface, and the integration test of the access framework conforms to the test requirements of GJB5000A.

### 7) Plug-in repository

As a result of business atomization, a large number of business plug-ins are generated. In order to better manage and use the existing business plug-ins, you can use some management tools such as Tortoise SVN to classify the various business plug-ins.

### 8) Integrated Development Environment

The QFusion framework provides a secondary development integration environment for business plug-ins, which helps users quickly develop and generate business plug-ins. At the same time, the QFusion framework provides visualization business integration tools to help users quickly combine business functions and publish applications.

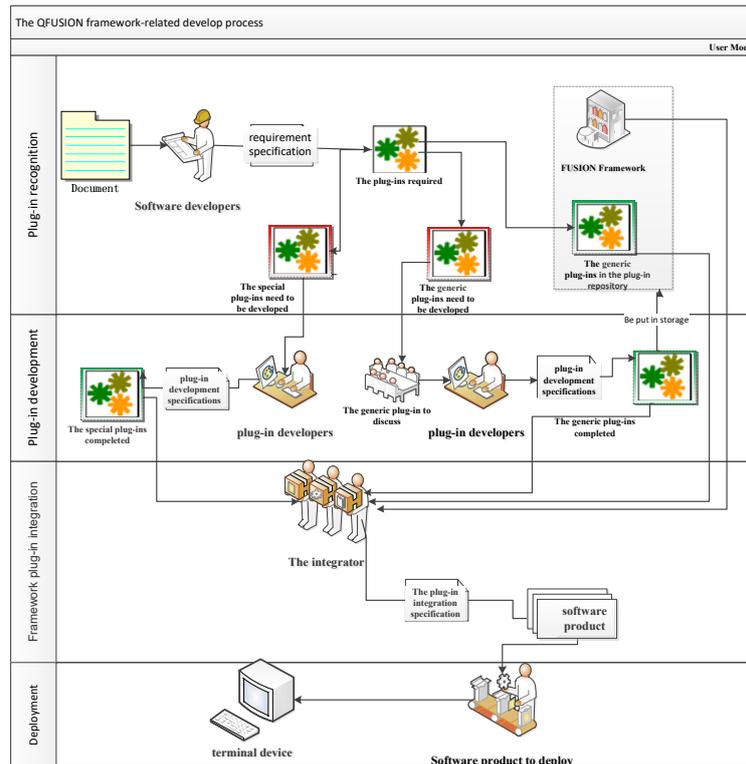
### 9) Flexible plug-in interface

Based on generality, the generic plug-ins can be designed with a large number of interfaces and redundancy in use. In order to block the redundant signals and redundant interfaces of generic plug-ins, the selection of signals and interfaces can be made in the configuration file to avoid unnecessary impact on business software.

## 2.5 The QFUSION framework-related develop process

The QFusion framework facilitates the collaboration among different groups doing software development. As all modules have the same format, they can be easily exchanged. This is especially advantageous when doing cross-checks and also helps implementing an overall quality control scheme. Code that is broken down into small independent plug-ins can be more easily reviewed by others than a single monolithic program.

The QFusion framework-related develop process includes plug-in recognition, plug-in development, framework plug-in integration, and deployment, which is shown in figure 4.



**Fig. 4.** The QFusion framework-related develop process

1) Plug-in recognition

Software developers identify the plug-ins required for the software according to the requirements of the software, and then identify the generic plug-ins and special plug-ins that need to be developed according to the generic plug-ins in the plug-in repository.

2) Plug-in development

The special plug-ins can be developed directly by plug-in developers following plug-in development specifications. The generic plug-in requires the relevant domain experts to discuss the generic plug-in and develop the plug-in. After the generic plug-in development is completed, you need to enter the plug-in library for other project applications.

3) Framework plug-in integration

The integrator integrates the development of the special plug-in and the generic plug-in based on the QFusion framework, and generates the software product according to the plug-in integration specification.

4) Deployment

The deployers deploy the software product to the corresponding terminal device.

### 3 The practice of QFusion

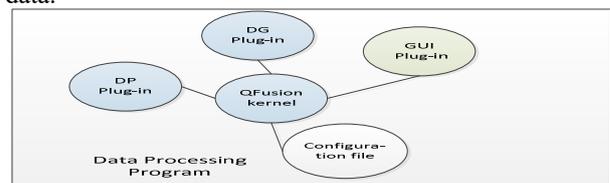
This section introduces the practice of QFusion framework in modern military field, which has practical meaning in the engineering application.

As we know, telemetry has important application on the task of ballistic trajectory measurement in the missile weapon tests, and processing and analysis of telemetry signal is very important to evaluate the missile weapon performances. However, the telemetry measurement parameters are affected by many factors, and uncertainty

exist in the tests, including environments, objects and equipments, and so on. The credibility of telemetry signal measured by ground stations is reduced because of the interference and error, so it's necessary to process the telemetry data. In this paper, we use the feature extraction and data processing algorithm to recognize the valid signal, which is very meaningful for telemetry data processing.

In this section, we firstly summarize the feature extraction and data processing algorithm, which is shown in figure 5. The original data of the input includes the valid signal and the disturbing of pump noise in the data transmission. In training phase, the original signal is preprocessed by frame division and noise reduction. Then, feature extraction is performed, and training algorithm is applied to establish an effective feature for each segment of signal. After modeling all the samples of training set, the model set is to form a model library. In the recognition stage, the rest samples are processed by the same pretreatment and feature parameters, matching with models in the model base according to certain rules. Finally, the operation results are output.

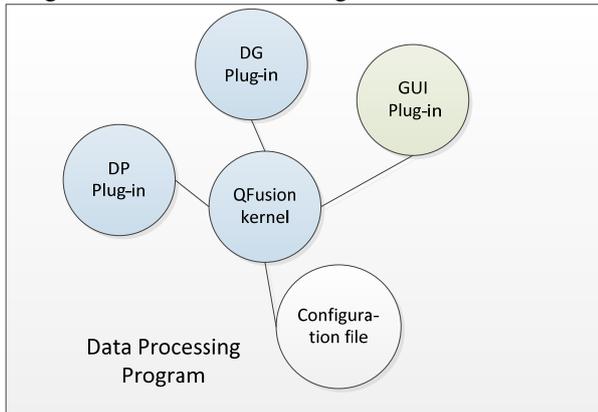
Afterwards, we implement the practice of QFusion framework-related develop process of the feature extraction and data processing algorithm for telemetry data.



**Fig. 5.** The feature extraction and data processing algorithm

Based on the QFusion framework-related develop process, the task is completed as follows.

Firstly, according to plug-in recognition, the data processing task is divided into three parts, including data generation plug-in (DG plug-in), data processing plug-in (DP plug-in) and graphical user interfaces plug-in (GUI plug-in). Where the DG plug-in and DP plug-in are the special business plug-ins, and the GUI plug-in is the generic plug-in. The configuration file is used to realize the logical relationship of software structure. The plug-in recognition result is shown in figure 6.

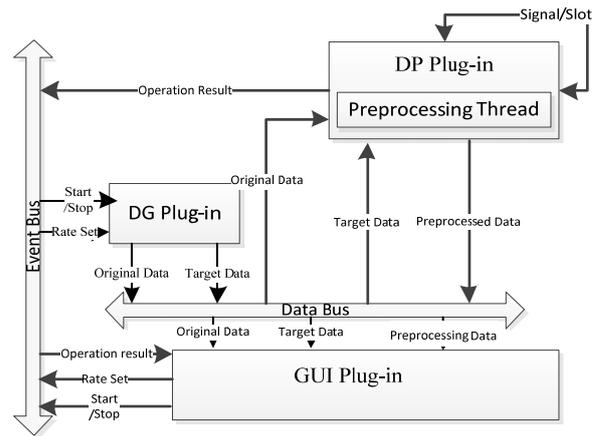


**Fig. 6.** The plug-in recognition result

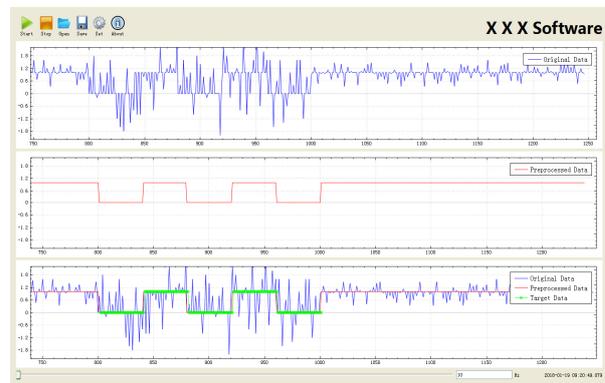
Secondly, the signal processing task is completed by the collaboration among three groups, including data generation plug-in development group, data processing plug-in development group and GUI plug-in development group, respectively.

Then, after the plug-in development is completed, the DG plug-in, the DP plug-in and the GUI plug-in are integrated to generate the software product according to the plug-in integration specification. Where the three plug-ins are assembled into the feature extraction and data processing application through the configuration file mentioned above. The data stream of the signal processing task is shown in figure 7. The DG plug-in, the DP plug-in and the GUI plug-in firstly register with the event manager. The three plug-ins that post the message simply send the message out thematically and the event manager is responsible for passing the message to all plug-ins subscribing to the topic.

Finally, the software product of feature extraction and data processing algorithm is deployed to the corresponding terminal device. The signal processing results are shown in figure 8.



**Fig. 7.** The data stream of the signal processing task



**Fig. 8.** The data stream of the signal processing task

The blue line is the original data generated by the DG plug-in, which is used to simulate the telemetry signal contaminated by pump noise in the data transmission. The red line is the preprocessed data. It is observed that the noise is largely reduced after the preprocessing. The green line is the recognition result, which shows that feature extraction and data processing algorithm can recognize the target data correctly. The results prove the validity of the QFusion framework.

## 4 Conclusions

We have presented a novel software framework named QFusion for cross-platform desktop software development. The QFusion framework uses the kernel and framework structure to build applications, which defines each business module as a plug-in for a specific function. The plug-in communicates with the outside through the registered interface, and the plug-ins can form a complete application through collaboration among different groups. For asynchronous requirements, a single service can work in a separate thread. The proposed framework can meet both synchronous and asynchronous communication between plug-ins. Communication between different plug-ins is carried out by event manager. Benefit from the QFusion framework, the production efficiency is improved, the cost of software is reduced, software pedigree and product line are achieved. The modular structure also leads to more flexibility. Based on the framework-related develop

process, the complexity is largely reduced. The practice of QFusion framework in modern military field is given in this paper. The operation results have demonstrated the effectiveness of QFusion framework.

## Acknowledgment

First of all, we thank all the collaborators of the joint work presented in this paper for their great contribution.

The work in this paper has been supported mainly by major project of fund of China Academy of Engineering Physics (Grant No. 060608-2017).

## References

1. W. D. Yang, Q. Chen, and N. Li, "A Framework-based Model of Software Construction," *2011 IEEE 2nd International Conference on Software Engineering and Service Science*, pp. 286-289 (2011).
2. R. Johnson, B. Foote. "Designing Reusable Classes," *Journal of Object-Oriented Programming*, vol. 6(1991).
3. E. Gamma, "Objekt orienterte software entwicklung am Beispiel von ET++," *Springer Verlag*(1992).
4. D. G. Firesmith, "Frameworks: the golden path to object, Nirvana," *Journal of Object-Oriented Programming*, vol. 7 (1994).
5. F. Buschmann, R. Meumier, H. Rohnrt. "Pattern-oriented software architecture-A system of patterns," *John Wiley and Sons*(1996).
6. W. D. Ma, Z. Liu, Y. Deng, L. X. Xu and W. F. Lin, "A Method of Software Reuse and Domain Modeling Based on Control-Oriented System Requirement Described Language", to be unpublished (2017).
7. W. D. Ma, Y. Deng, Z. Y. Wang, "COSRDL: A Control-Oriented System Requirement Described Language Based on Directed Acyclic Graph, " *2017 IEEE 8th International Conference on Software Engineering and Service Science*, pp. 286-289 (2017).
8. C. Kopper, "A software framework for KM3NeT," *Nuclear Instruments and Methods in Physics Research A*, vol. **602**, pp. 107-110 (2009).
9. D. Kirk, E. Tempero, "A lightweight framework for describing software practices, " *The Journal of Systems and Software*, vol. **85**, pp. 582-595(2012).
10. E. A. Stout, R. W. Carey, C. M. Estes, J.M. Fisher, L.J. Lagin, D. G. Mathisen, C. A. Reynolds, R. J. Sanchez, " CORBA- Based Distributed Software Framework for the NIF Integrated Computer Control System," *UCRL-CONF-236998*, pp. 1-5 (2007).
11. B. Ding, H. Wang, D. X. Shi, J. N. Cao, "Taming Software Adaptability with Architecture-Centric Framework," *IEEE*, pp. 145-151 (2009).
12. C. Y. Wang, X. F. Zheng, X. Y. Tu, "Coordination Framework Based Software Development Approach and Its Applications," *2008 International Conference on Computer Science and Software Engineering, IEEE Computer Society*, pp. 537-540 (2008).