

## RESEARCH ON CROSS PLATFORM DEVELOPMENT MODEL BASED ON QUICK QT

Xiaohua Zhang<sup>1, a</sup>, Bo Huang<sup>2, b</sup>

<sup>1</sup>Department of Computer Science and Technology, Chengdu Neusoft University, Chengdu, 611844, China

<sup>2</sup>Department of Computer Science and Technology, Chengdu Neusoft University, Chengdu, 611844, China

<sup>a</sup>email: 137136612@qq.com, <sup>b</sup>email: 569960256@qq.com

**Keywords:** Intelligent Vehicle; Development Model; Cross Platform; Qt Quick

**Abstract.** CPADM is a new cross platform application development model proposed by the author, It is based on QT Quick technology and be used in the development of all applications. In this paper, the shortcomings of the traditional development model is analyzed to explain the significance of proposing CPADM model firstly; Secondly briefly introduces the relevant Quick Qt technology used in CPADM model; Thirdly, mainly introduces the using scenes, the function of each component and the communication mechanism between each component in basic CPADM model and the advanced CPADM model; Lastly, introduces the causes of why CPADM model can cross platform and the results of using it in application development. Compared with the traditional development mode, the CPADM model is easy to use and reduces the complexity and cost of application development. Therefore, CPADM model is feasible to develop the cross platform application, and is very suitable for the development of the cross platform application.

### Introduction

“Intelligent electric bicycle vehicle mounted system” requires being able to run on the mainstream mobile platform, such as Android, IOS. At present, the traditional solution is to develop a version for each mobile platform, this program can be called non-cross platform development model. Non-cross platform development model increases the difficulty and time of project development and maintenance, which leads to the increase of the cost. Through consulting a large number of cross platform development documents, this paper proposes a cross platform development model CPADM, which based on the Quick Qt technology, and is used for research and development of “intelligent electric bicycle vehicle systems”.

### Qt Quick Summary

Qt is a cross platform framework for creating wonderful user interfaces and powerful native applications. Quick Qt based on the inherent advantages of Qt, can quickly and easily create a dynamic touch interface and lightweight applications for mobile and embedded devices [1].

Qt is composed of ten major components, as shown in figure 1. Core Qt is the core component, is the basis of other components, other components are not necessary, can be added and removed according to the actual needs. Qt Quick、 Qt Quick Layouts、 Qt Quick Dialogs、 Qt Quick Controls and Qt QML is used for UI (User Interface) design. Qt Multimedia is used for multimedia programming, Qt Network is used for network programming, Qt SQL is used for database programming, Qt Test is used for project testing.

QML (Meta-Object Language Qt) is the core and foundation of Quick Qt technology, is a kind of markup language that supports JavaScript, and is used to describe the user interface of the program.

All API (Application Program Interface) in Qt are implemented with the C++ class. In Qt, you can use the `qmlRegisterType` global function to register any C++ class as a QML type, so that the QML page can call the C++ method by the QML object. On the other hand, you can use the

findChild method to find any object on the QML page in the C++ class.

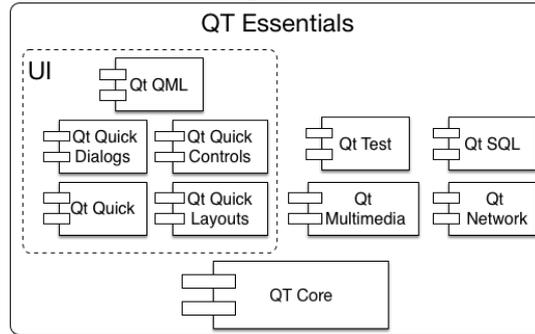


Fig.1. Qt module composition

In QT, the communication between the objects is use by "signal and slot" mechanism. "Signal and slot" is a communication mechanism defined by QT, when the specified signal is sent out, all the slots connected with the signal function will be executed immediately [2-4]. The correlation between signal and slot is realized by calling the connect function of the QObject object in the C++ class, which is a bridge between the signal and the slot [5-6]. In the QML page, the correlation between the signal and the slot is achieved through the target attribute and the onXXX slot function in the Connections QML type. In above, target is used to specify the object to receive the signal, the onXXX slot function is used to receive and process the XXX signal, and XXX is the name of the signal.

### Basic CPADM Model

In this paper, the CPADM model can be divided into the basic CPADM model and the advanced CPADM model according to the usage scenarios. When the application needs not integration with the native code which coding by a special platform SDK (Software Development Kit), you can use the basic CPADM model shown in Figure2 to develop cross platform applications based on Quick Qt. Otherwise, you should use the advanced CPADM model shown in Figure3 to develop cross platform applications based on Quick Qt.

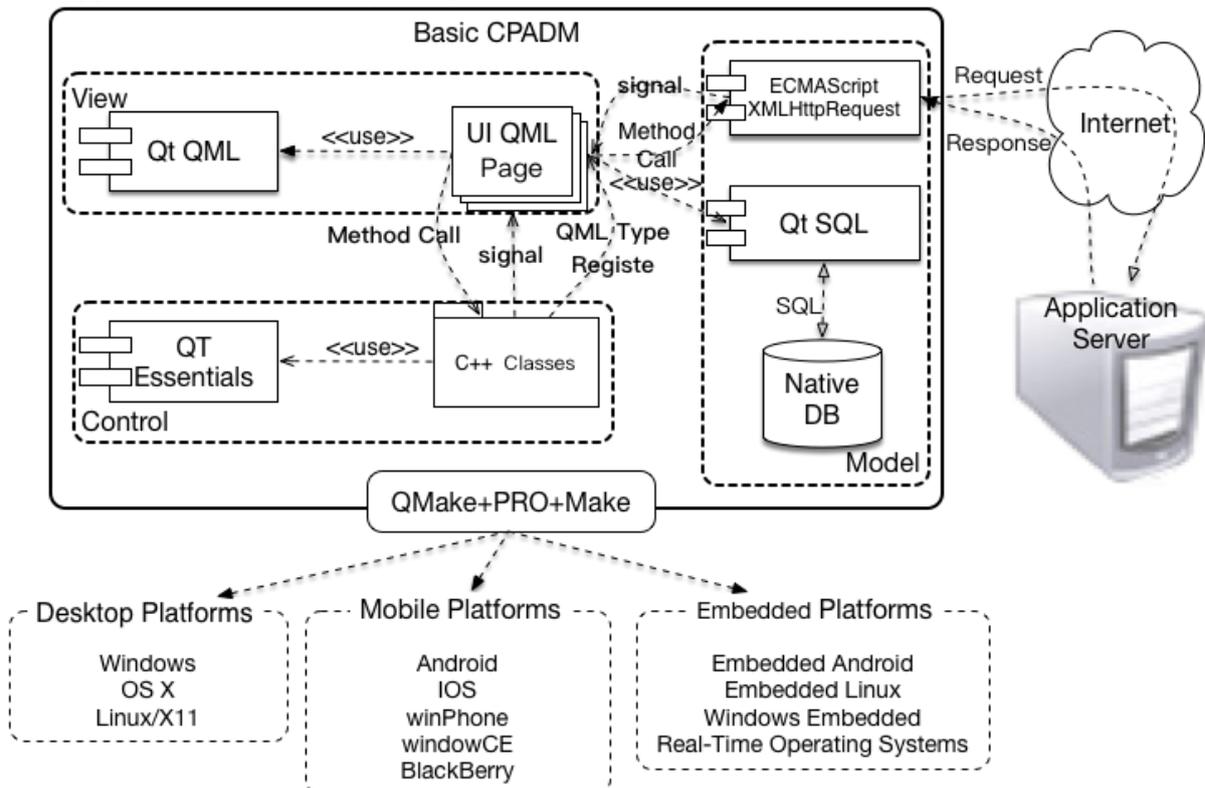


Fig.2. Basic CPADM model

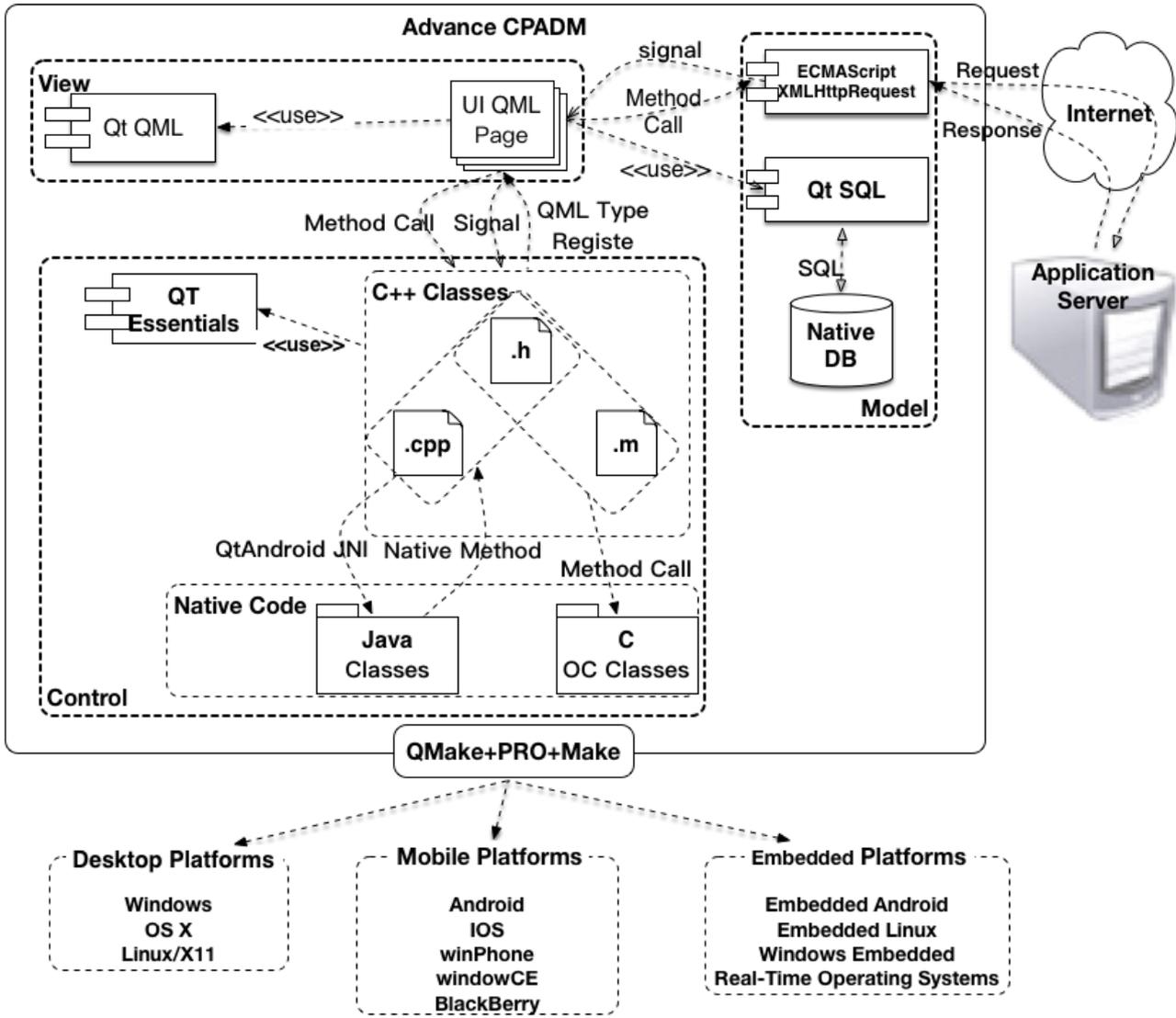


Fig.3. Advanced CPADM model

In Figure 2, the basic CPADM model is designed using the MVC model. The model layer of the basic CPADM models responsible data request and processing, you can use the XMLHttpRequest to asynchronously request data on a remote server, also can use the QT SQL to synchronously request the data in the local database; The View layer is composed of several QML pages, which is responsible for the design and implementation of the UI interface; Controller layer can be placed in the event processing section of QML page, also can be placed in a separate C++ class. Controller layer is the bridge between the view layer and the model layer, which controls the application logic [7].

There are two kind of development pattern to use basic CPADM model development application, one is non-layered mode, that all of the codes are placed in the QML page; another is layered model, that three layer codes are placed in three different resources.

Non-layered development model is mainly used for small simple applications; the purpose is to pursue rapid development. Its development process is: A. design the UI interface in the QML page by using QML controls; B. using the onXXX slot function to receive the signal in control which need to respond to event, where XXX is the event signal name. C. In the onXXX slot function implementation code block, using XMLHttpRequest object asynchronously request remote server data; D. in the end, update the component in QML page with data from step C.

Layered development model is mainly used in the development of large, multi person collaborative development. Layered development model is based on the Non-layered development model. Compared with the non-layered development mode, the layered development model

separates data logic code and application logic code from the QML page to C++ class, and respectively as the model layer and controller layer. In the end, the QML page that only contains the display logical code acts as the view layer. The key point of the layered development mode is how to communicate between each layer, so that each layer becomes an organic whole. In this paper, the CPADM model uses the Qt signal and slot mechanism to integrate each layer; the signal processing flow is shown in Figure 4.

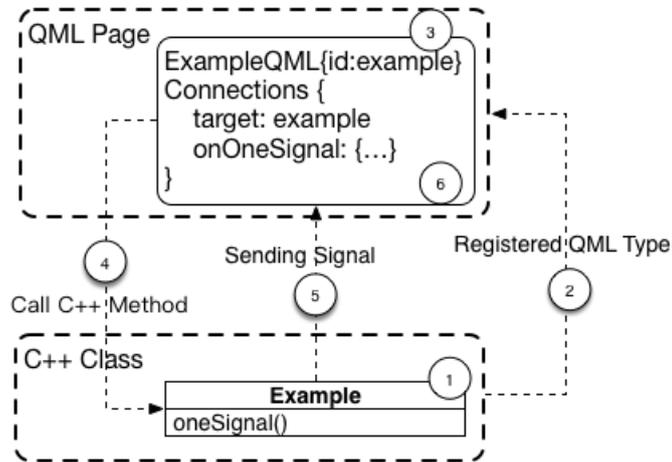


Fig.4. Communication process between QML and C++ class

- A. Code the C++ class of controller layer, and define the signal method.
- B. Register C++ class in step A as a QML type by the global function `qmlRegisterType`.
- C. Declare a QML component object with The QML type registered in the step B in QML page.
- D. Call the method of QML object declared in step C, and call the emit Qt macro to send a signal before the end of the method, such as `emit oneSignal ()`.
- E. Declare Connections object to receive and process the signal of the target object in in the QML page.

If you send a signal to carry data, then the data can be used as parameters of signal method, when the signal is sent by the emit macro, all the slot functions associated with the signal can receive the parameter data.

### Advanced CPADM Model

Advanced CPADM model as shown in Figure 3 is mainly used to integrate the platform native code. Compared with the basic CPADM model, it extends the control layer of the basic CPADM model to integrate native code, while the other layer has not changed. The control layer of advanced CPADM model is composed of two parts: C++ class and native code. The C++ class is an intermediate conversion layer between the QML layer view page and the underlying native code. Specifically, C++ class is responsible for receiving the QML page data, calling the native code, and sending the processing results of the native code back to the QML page in a synchronous or asynchronous way.

C++ class is composed of header file (. H) and implementation files (.CPP or .mm), the header file declares public interface, and the implementation file implements the public interface. In order to integrate the native code of multiple platforms, all platforms have a shared C++ header file, and every platform has independent C++ implementation file. According to whether the native code can be compiled by C++ compiler, the native code can be divided into two categories. One is the native code can be compiled by C++ compiler, another is the native code cannot be compiled by C++ compiler. This paper will be used the C language code and Objective-C language code as a representative to analyze how to integrate the native code that can be compiled by the C++ compiler in the advanced CPADM model. Otherwise, This paper will be used the Java language code as a representative to analyze how to integrate the native code that cannot be compiled by the C++ compiler in the advanced CPADM model.

For the C language code, the C++ class can call its function directly. For Objective-C native code, C++ objects and Objective-C objects can call each other's methods. So, when the advanced CPADM model integration the native code that can be compiled by the C++ compiler, the native code can be as the C++ code to call, compile and run.

Because the Java code can not be compiled by the C++ compiler, and therefore it can not be mixed with the C++ code in control layer. Therefore in the advanced CPADM model, Java code needs to be placed separately in the Java source file, and need to use the Java compiler to compile and need to use Java virtual machine to run. In Qt, the C++ object cannot directly call the method of Java object, but can use the JNI (Java Native Interface) technology to achieve communication between C++ object and Java object. Qt extends JNI interface to QAndroidJniEnvironment class and QAndroidJniObject class, and put them in the QtAndroid namespace. QAndroidJniEnvironment class represents JNI environment, packages the JNIEnv in the JNI. QAndroidJniObject class represents a Java object, which provides a lot of methods for developers to create Java objects and call Java methods [8]. Methods in the QAndroidJniObject class can be divided into three categories according to their functions. One is the QAndroidJniObject construction method, which is used to construct the Java object; two is the callObjectMethod method, used to call the Java instance method; three is the callStaticMethod method, used to call the Java static (class) method. In the control layer of the advanced CPADM model, communication from C++ class to Java class use callStaticMethod of QAndroidJniObject, communication from Java class to C++ class use Java Native method.

### The principle of cross platform mechanism of CPAMD model

The root cause why the CPAMD model can be used for cross platform development is that Quick Qt technology is a cross platform. Qt Quick use Make tool for cross platform package and deployment.

Every Qt project has a global configuration file, that is, the pro file. The configuration file is used for configuring shared project resources of all platform, is used for configuring exclusive project resources of every platform. These project resources include QML files, header files, implementation files and library files. All project resources of one platform are consist of shared project resources for all platform and exclusive project resources for the platform in the global configuration file. In the project configuration file, Qt uses “conditional test configuration block” to configure exclusive project resources for each platform [8]. An example of “conditional test configuration block” is shown in Figure 5.

```
HEADERS += Tuolukenative.h
android{
    SOURCES += Tuolukenative.cpp
}
ios {
    OBJECTIVE_SOURCES +=
    TuolukeNative.mm
}
```

Fig.5. An example of conditional test configuration block

Figure 5 shows that all platforms share the "Tuolukenative.h" header file in the project, and the implementation file of this header file is "Tuolukenative.cpp" on the Android platform, is "TuolukeNative.mm" on the IOS platform. After the completion of the project development and configuration of the configuration file, it will enter the project's build and run stage. In the building stage and Running stage, QT uses the QMake tool to read the configuration files, and then compile, package, install, run the application to a variety of target devices.

### Test results

The CPADM model is used in the development of "intelligent electric bicycle vehicle mounted

system" that should be run on Android platform and IOS platform, and the application results are shown in Table 1.

Table.1. The application results of CPADM model

The requirements module	Qt development module	QTCMP model type	Platform
UI	Qt QML	Basic model	Android/IOS
Web request	XMLHttpRequest	Basic model	Android/IOS
Photograph	Qt Multimedia	Basic model	Android/IOS
Bluetooth communication	Qt Bluetooth	Basic model	Android/IOS
Local DB storage	Qt SQL	Basic model	Android/IOS
Baidu map		Advanced model	Android/IOS
Baidu navigation		Advanced model	Android/IOS

From Table 1, we can know that the source code of “intelligent electric bicycle vehicle mounted system” developed by QTCMP model is only one copy, the robot a throw-in, but it can be perfect to build, run on the Android platform and IOS platform.

### Conclusion

With the rapid development of mobile Internet technology and the Internet of things, there has been numerous hardware and software platform, which leads to the use of native application development model development for all platform application becomes very difficult, and development costs increased sharply. This paper proposed the CPADM development model based on the QT quick technology, and this model is used for research and development of “intelligent electric bicycle vehicle mounted system”. The model provides new solutions for cross platform application development.

### References

- [1] Xiaohui An. Qt Quick core programming[M].Beijing:Electronic Industry Publisher, 2015:10-15.
- [2] JasminBlanchete. C++ GUI Qt4 programming [M]. Beijing: Electronic Industry Publisher, 2008:1-35-43.
- [3] Juan Zhang, Xuelan Zhang. Implementation of GUI application based on Linux embedded [J]. Computer application, 2003,23 (4): 115-117.
- [4] Jian Xu. The realization of embedded browser and GUI based on QT [D].Xi'an: Xi'an Electronic Science and technology, 2008:20-24.
- [5] Chao Huang. Research and implementation of embedded GUI based on Qt [D]. Jilin: Jilin University, 2011:20-26.
- [6] Yingge Chen, Li Ma, Xiaoyin Wang. Analysis and teaching practice of signals and slots mechanism in Qt/E [J]. Chang Shu Polytechnic College Journal, 2008,22 (10): 108-112.
- [7] Chunhong Li, Jianhua Gao. An improved MVC design architecture using hierarchical model [J]. Computer engineering and design, 2007, 28 (4): 766-769.
- [8] Xiaohui An. QT on Android core programming [M]. Beijing: Electronic Industry Publisher, 2016:67-76. 2010.