Abstract

Rapid development for network software such as ATM and wireless data communication networks, needs more efficient networking software design methods. We have constructed an OOATM [Yan96] model and a corresponding C++ library for object-oriented design/implementation of ATM protocol machines. OOATM model explores the potential for producing reusable software modules by discovering the underlying generic class structures and behavior. Applying OOATM model to specifying/ coding does improve reuse for protocol implementation since each object specified has a corresponding (also easy use) category of class(es) and methods.

Visual programming [Burnett95] [Shu88], referring to any system that allows the user to specify programs in a multi-dimensional style, is intended to ease the programming process and reduce the efforts of specification. Object-oriented method is a better model to adapt visualization technique. In this report, we present a Visual Object-Oriented Protocol Development Environment (VOOP DE) based on OOATM model and the techniques of visual programming to facilitate the implementation of communication protocol.

The VOOPDE include a graphical protocol state diagram editor (PSDE), a PDU specification editor and a protocol code generator. The graphical protocol states diagram editor provides well-designed edit and display facilities that ease protocol specification. The PDU editor provides a bit-by-bit graphic editor to specify the PDU format (message coding format). The protocol
code generator generates a set of classes based on the OOATM model and the associated C++ library. The classes are a skeleton of the protocol implementation. Applying these tools not only can reduce the effort of specifications but also can generate C++/object code semi-automatically to ease protocol implementation.

**Keywords**: Protocol, Object-oriented, ATM, Program specification, Reuse, Visual programming

二、缘由与目的

Visual Programming 最近幾年來越來越普遍，其中一個原因為圖形方面相關的硬體技術使得使用者與電腦間可經由 graphics 來溝通。透過具體與合適的圖形可以使得使用者更容易與電腦溝通。一般而言 Visual Programming 可以看成由一些有意義的 graphics 和 graphical components 來構成的過程。

直接用文字來描述物件及其間的關係遠比用圖形(Graphics)來的困難，換句話說，用 VISUAL PROGRAMMING 的技術來撰寫物件導向規格似乎比用文字語言容易。如果能夠運用 Visual Programming 技術來撰寫 protocol specification，不但可以減少 specification 時的努力，也應可以幫助撰寫者更容易表現出 protocol 中各個 protocol state 間的關係使得 protocol specification 更方便 maintain。

我們於計劃的前兩年時發展了一套物件導向模式 OOATM Model(如圖一)，我們也已製作一套相對應的 C++程式庫。由於在 OOATM Model 中每一 object 有一對應的 class(es) 及其 method(s) 目錄，應用 OOATM Model 來設計或撰寫 ATM PROTOCOL 機器的程式確實可以改善再利用。這些我們歸納整理出來的 objects 與 classes 讓我們決定設計一些相對應的 graphical components 並訂定一些 graphical components 間相互的關係的 rules 加以規範，以達成使用 graphics 的方式撰寫 protocol specification，即所謂的 Visual Programming。利用這種 graphical components 和 objects 對應關係，我們可以進一步分析這些 graphical components 和 C++ library 的關係進而設計一個 Code Generator，能將 Visual specification 自動依循 C++ library 的 classes 轉譯成 C++ program Code。再進一步我們整合這些工具發展一套通訊協定發展環境幫助使用者實作通訊協定。
The VOOPDE includes a graphical protocol state diagram editor (PSDE), a PDU specification editor and a protocol code generator. The architecture of the VOOPDE is shown in Figure 1.

The graphical protocol state diagram editor provides well-designed editing, displaying facilities that ease protocol specification. Using PSDE to edit protocol state transition diagram can enhance the understandability and readability of specification. For example, as shown in Figure 2, one uses the PSDE to edit the ATM signaling protocol [ATM Forum 95]. The round rectangles represent states and the arrow lines representing transitions.

The PSDE provides two views for a state diagram. One view provides a graph for the state diagram. User can move, resize, rename and edit a state and
specify transitions between states at the graph view. When user issues the command called "Edit" on a state, the PSDE display a text editor of the state. The text editor shows the generated codes of the state and user can modify and extend the codes.

An another view is a tree structure view of the state diagram. The tree structure shows the relationship between states and transitions more clearly.

Figure 2. ATM point-to-point signaling processing
The PDU editor (as shown in Figure 3) provides a bit-by-bit graphic editor to specify the PDU format (message coding format). Figure 3 shows an example of the PDU, general message organization of UNI signalling protocol. The protocol code generator generates a set of classes based on the OOATM model and the associated C++ library. The classes form a skeleton of the protocol implementation. Using those tools can not only reduce the effort of specifications but also generate C++/object code semi-automatically to ease protocol implementation.

**Conclusion and Future Work**

We have proposed a Visual Object-Oriented Protocol Development Environment (VOOPDE) to facilitate the implementation of communication protocol. Applying these tools not only can reduce the effort of specifications but also can generate C++/object code semi-automatically to ease protocol implementation.

Our future study includes:

- Applying VOOPDE to the development of more complex, larger communication software
- Make VOOPED usable in a distributed environment (groupware)

**Reference**


