Tạp chí Tin học và Điều khiển học, T. 19, S. 2 (2003), 167–172

PHẦN MỀM ĐIỀU KHIỂN MÁY RÚT TIỀN TỰ ĐỘNG

LÊ HOÀI ANH

Viện Công nghệ thông tin - NCST

Abstract. The automation of the bank transaction is developing nowadays, for example using ATM (Automatic Teller Machine).This article presents a building method of the structure of an ATM Driver (The software which pilots ATM to carry out the bank transaction) which I have developed.

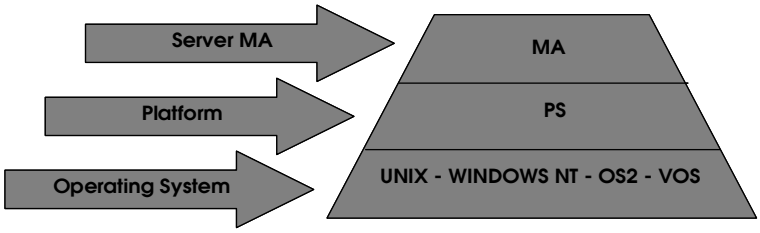
Tóm tắt. Ngày nay, việc tự động hóa các phiên giao dịch ngân hàng rất phát triển, ví dụ như việc sử dụng các máy rút tiền tự động ATM. Trong bài báo này, chúng tôi giới thiệu một phương pháp xây dựng cấu trúc cho một ATM Driver (Phần mềm điều khiển máy ATM thực hiện các phiên giao dịch ngân hàng).

1. GIỚI THIỆU

Việc ứng dụng tin học-điện tử trong ngân hàng và tài chính đang rất phát triển. Một trong số đó là lĩnh vực Monetic được hiểu như là việc tự động hóa các phiên giao dịch ngân hàng. Ví dụ như: Sử dụng các máy rút tiền tự động ATM (Automatic Teller Machine), sử dụng các máy thanh toán điện tử, sử dụng các thẻ thanh toán tự động trong nước (ví dụ như hệ thống blue card ở Pháp) hay là sử dụng các thẻ thanh toán tự động quốc tế (visa card), v.v

Để giúp cho hệ thống các máy ATM hoạt động cũng như trợ giúp cho việc tự động hóa các phiên giao dịch ngân hàng, cần xây dựng một phần mềm trung tâm để quản lý và điều khiển tất cả các hoạt động trên. Một phần mềm như vậy ta gọi là MA (Monetic Application). MA sẽ có nhiệm vụ đáp ứng tất cả các ứng dụng Monetic, MA là một môi trường cho phép chúng ta phát triển và tích hợp vào đó ngày một nhiều hơn các ứng dụng Monetic. MA gồm nhiều thành phần khác nhau, ví dụ như phần mềm điều khiển các phiên giao dịch ngân hàng cho máy rút tiền tự động ATM mà ta sẽ đề cập chính trong bài báo này (ATM Driver), phần mềm đảm bảo an toàn cho máy rút tiền tự động ATM (ATM security), phần mềm hỗ trợ thanh toán quốc tế bằng thẻ visa card, v.v...

MA là một hệ thống được môđun hóa và cấu hình cao, nó cho phép chúng ta xây dựng, phát triển và tích hợp thêm các ứng dụng Monetic một cách dễ dàng.

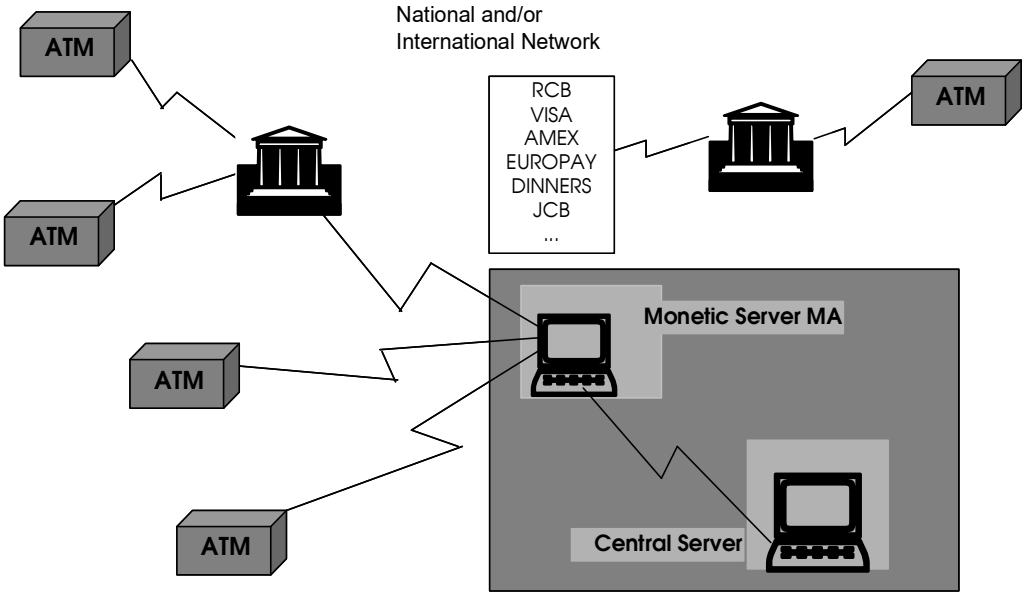


Hình 1. Server MA chạy trên các hệ điều hành khác nhau

MA được xây dựng trên một phần mềm nền gọi là PS (platform Software) giúp cho hệ thống MA có thể phát triển và chạy trên bất cứ một hệ thống mở nào như: (UNIX, OS2,

NT ...), việc xây dựng các ứng dụng Monetic trong hệ thống MA sẽ trở lên trong suốt hơn đối với các hệ điều hành khác nhau (Hình 1).

ATM Driver là một phần của hệ thống MA đảm nhiệm việc điều khiển các phiên giao dịch ngân hàng trên máy rút tiền tự động ATM. ATM Driver được cài đặt trên Monetic Server MA. Với một tiến trình ATM Driver có thể quản lý được vài trăm máy ATM, với vài tiến trình ATM Driver ta có thể quản lý tới một vài ngàn máy ATM trên chỉ một Server MA (Hình 2).



Hình 2. Sơ đồ mạng lưới ATM

Bài viết này sẽ trình bày một phương pháp xây dựng cấu trúc cho ATM Driver mà chúng tôi đã thực hiện.

2. NỘI DUNG

ATM Driver là một thành phần của hệ thống MA, nó giải quyết bài toán với nội dung được tóm tắt như sau:

Đầu vào: Ta có các máy ATM với nội dung cụ thể về các protocol riêng của nó. Các protocol này được cung cấp bởi các nhà sản xuất ATM, nó qui định các chuẩn trong việc giao tiếp với nó.

Bảng 1. Các hãng sản xuất ATM

Nhà sản xuất	ATM protocol
IBM	$473x$, $478x$
NCR	NDC+
DASSAUT	GABA, GABD

Đầu ra: Ta phải xây dựng phần mềm ATM Driver để điều khiển các máy ATM thực hiện các giao dịch ngân hàng như: rút tiền, gửi tiền, chuyển khoản, lấy thông tin, lấy bảng sao kê,... Mỗi ATM Driver có thể quản lý và điều khiển nhiều máy ATM khác nhau.

Cấu trúc của ATM Driver: Trên thực tế có rất nhiều nhà sản xuất máy rút tiền tự động ATM (Bảng 1), họ cho ra đời nhiều loại máy ATM khác nhau, mỗi loại sẽ hoạt động với một protocol riêng của mình. ATM protocol sẽ qui định các chuẩn trong việc giao tiếp với nó, trong đó có qui định cấu trúc của các message trao đổi giữa ATM và Server trung tâm. Các message chứa thông tin cần thiết cho các phiên giao dịch ngân hàng được yêu cầu bởi người sử dụng.

Do mỗi ATM Driver có thể hỗ trợ nhiều loại máy ATM với các loại protocol khác nhau vì vậy ATM Driver cần có một bộ phận làm nhiệm vụ chuẩn hóa các message trao đổi giữa ATM và Server trung tâm đó là các bảng Protocol. Dưới đây là phương pháp xây dựng cấu trúc của một ATM Driver mà chúng tôi đã thực hiện (Tại công ty ASTRIA - 137 - Đại lộ Voltaire- Paris - một công ty chuyên sản xuất phần mềm về lĩnh vực Monetic):

Mỗi ATM Driver gồm 3 bộ phận chính là: ôtômát trung tâm, các bảng Protocol và ôtômát xử lý phiên giao dịch.

• ***Ôtômát trung tâm:*** Đây là bộ phận chính của mỗi ATM Driver, bộ phận này có nhiệm vụ đồng bộ các sự kiện mạng yêu cầu giao tiếp hội thoại với ATM, các lời gọi xử lý các protocol riêng và việc thực hiện các phiên giao dịch ngân hàng chuẩn. Nó là một ôtômát toán học với các thành phần là các trạng thái và các sự kiện. Nó hoạt động theo nguyên tắc chung của một ôtômát toán học, ôtômát này khi gặp một sự kiện sẽ thay đổi trạng thái của mình và thực hiện một công việc tương ứng.

Các trạng thái của ôtômát trung tâm chỉ trạng thái của liên kết giữa Server trung tâm với ATM, ví dụ như: ATM được kết nối, ATM không được kết nối, (Bảng 2).

Bảng 2. Các trạng thái của ôtômát trung tâm

Tên	Ngữ nghĩa
et_dcxhs	ATM không kết nối (Server trung tâm không phục vụ)
et_dcxes	ATM không kết nối (tạm thời, Server trung tâm phục vụ)
et_cnx	ATM được kết nối (trạng thái nghỉ đợi message từ ATM)
et_cnxtr	ATM được kết nối và phiên giao dịch đang được thực hiện
...	...

Các sự kiện làm thay đổi trạng thái của ôtômát này là các yêu cầu đến từ ATM hoặc Server trung tâm. Ví dụ như: Server trung tâm yêu cầu kết nối với ATM, ATM gửi message về Server trung tâm,v.v (Bảng 3).

Bảng 3. Các sự kiện của ôtômát trung tâm

Tên	Ngữ nghĩa
ev_cnx	Yêu cầu kết nối với ATM
ev_dcx	Yêu cầu huỷ kết nối với ATM
ev_msg	Có message gửi từ ATM tới Server trung tâm
ev_tr	Có message trả lời từ Server trung tâm về ATM
ev_cmd	Có những yêu cầu vận hành gửi về ATM
...	...

Khi ôtômát trung tâm nhận một sự kiện nó sẽ thay đổi trạng thái và thực hiện những công việc sau:

+ Gọi bảng các hàm của ôtômát trung tâm: Một bảng các hàm xử lý được xây dựng trước như một thư viện hàm nhằm phục vụ cho quá trình vận hành của ôtômát. Khi một sự kiện

E1 gửi đến ôtômát đang ở trạng thái S1 nó sẽ gọi các hàm F1, F2,... liên quan trong bảng hàm để xử lý nhưng công việc tương ứng.

Trong khuân khổ của bài báo nhằm giới thiệu phương pháp xây dựng cấu trúc cho ATM Driver, chúng tôi không trình bày tất cả các hàm xử lý ôtômát, mà chỉ thể hiện qua một ví dụ: khi ôtômát ở trạng thái et_cnxtr tức là ATM được kết nối, mà gặp sự kiện ev_msg tức là có message gửi từ ATM tới Server trung tâm thì ôtômát sẽ vận hành như sau:

	et_cnx (ATM được kết nối)
ev_msg (Có message gửi từ ATM tới Server trung tâm)	⟶ state: **et_cnxtr** do *g_inim* if OK then do *p_valm* | if OK then do *p_iden* | if OK then do *g_gnext* | if OK then do *g_frep* | if not then do *p_apro* | if not then do *g_dcxp* ⟶ state: **et_dcxes** | if not then do *g_dcxp* ⟶ state: **et_dcxes**

g_inim, *p_valm*, *p_iden*, *g_gnext*, *g_frep*, *p_apro*, *g_dcxp* là các hàm được xây dựng trước trong bảng hàm nhằm mục đích như:

g_inim : khởi tạo message

p_valm : đánh giá sự hợp lệ của mesage

p_iden : nhận ra phiên giao dịch cần thực hiện

g_gnext : đọc thông tin đi kèm

g_frep : định dạng câu trả lời từ Server trung tâm về ATM

p_apro : thông tin lỗi không thực hiện được

g_dcxp : huỷ liên kết giữa Server trung tâm và ATM

Ta thấy ôtômát sẽ chuyển về trang thái et_cnxtr (ATM được kết nối và có phiên giao dịch đang được thực hiện) nếu không gặp lỗi hoặc chuyển về trạng thái et_dcxes (ATM không được kết nối tạm thời, Server trung tâm phục vụ) nếu gặp lỗi.

+ Gọi bảng Protocol để xử lý giao dịch: Các bảng Protocol sẽ giúp cho việc chuẩn hóa các ATM message, chuyển tất cả các mesage trao đổi giữa ATM và Server trung tâm về một dạng đồng nhất, giúp cho ôtômát xử lý phiên giao dịch trở lên trong suốt với các máy ATM khác nhau.

+ Gọi ôtômát xử lý phiên giao dịch để xử lý phiên giao dịch được yêu cầu ...

• ***Các bảng Protocol:*** chịu trách nhiệm chuẩn hóa các ATM message, bảng Protocol sẽ chuyển đổi các message gửi từ ATM về Server trung tâm (các message được cấu trúc dưới định dạng riêng của ATM protocol) thành các message được cấu trúc dưới định dạng qui ước bởi Server MA và ngược lại các bảng Protocol cũng sẽ chuyển đổi các message gửi từ Server trung tâm MA về ATM (các message được cấu trúc dưới định dạng quản lý bởi MA) thành các message được cấu trúc dưới định dạng riêng quản lý bởi ATM protocol. Các bảng Protocol giúp cho chúng ta chuẩn hóa việc định dạng các message chứa thông tin về các phiên giao dịch ngân hàng của các máy ATM khác nhau dưới dạng quản lý chung bởi Server MA. Nó giúp cho ôtômát trung tâm và ôtômát xử lý phiên giao dịch trở nên độc lập với các ATM

protocol. Các bảng Protocol cũng là những ôtômát nhưng không quản lý bất cứ một trạng thái nào, nó chỉ quản lý các sự kiện mà đến từ ôtômát trung tâm. Khi mà bảng Protocol nhận một sự kiện thì nó sẽ thực hiện một hàm tương ứng trong một bảng các hàm được xây dựng cho việc xử lý giao dịch như: đánh giá sự hợp lệ của một message, nhận dạng thông tin về phiên giao dịch ngân hàng trong message, xây dựng message trả lời, v.v... (Bảng 4).

Bảng 4. Các sự kiện của bảng Protocol

Tên	Ngữ nghĩa
ev_pvalm	Kiểm tra sự hợp lệ của các message
ev_piden	Lấy thông tin về phiên giao dịch cần thực hiện từ message
ev_frep	Định dạng câu trả lời từ Server trung tâm về ATM
ev_cmd_env	Gửi message đặc biệt cho yêu cầu vận hành
ev_cmd_imm	Quản lý trung gian những yêu cầu vận hành
...	...

• ***Ôtômát xử lý phiên giao dịch:*** Đây là bộ phận giúp cho việc xử lý các phiên giao dịch ngân hàng chuẩn được nhận ra bởi bảng Protocol như: rút tiền, gửi tiền, chuyển khoản, lấy sao kê,... Nhờ có việc bảng Protocol chuẩn hóa ATM message mà ôtômát xử lý phiên giao dịch trở nên độc lập với bất cứ ATM protocol nào. Nó cũng là một ôtômát toán học với các thành phần là các trạng thái và các sự kiện. Nó hoạt động theo nguyên tắc chung của một ôtômát toán học, ôtômát này khi gặp một sự kiện sẽ thay đổi trạng thái của mình và thực hiện một công việc tương ứng (ở đây là xử lý phiên giao dịch ngân hàng chuẩn).

Các trạng thái của ôtômát này chỉ ra phiên giao dịch nào đang được thực hiện bởi Server trung tâm (Bảng 5).

Bảng 5. Các trạng thái của ôtômát xử lý phiên giao dịch

Tên	Ngữ nghĩa
et_repos	Không có phiên giao dịch nào đang thực hiện
et_txxx	Phiên giao dịch xxx đang thực hiện

Ôtômát cũng bao gồm các sự kiện như là có phiên giao dịch ngân hàng yêu cầu xử lý hay có câu trả lời từ Server trung tâm, v.v... (Bảng 6).

Bảng 6. Các sự kiện của ôtômát xử lý phiên giao dịch

Tên	Ngữ nghĩa
ev_txxx	Phiên giao dịch xxx yêu cầu thực hiện
ev_trep	Nhận kết quả thực hiện phiên giao dịch

Ôtômát xử lý phiên giao dịch cũng có bảng hàm của nó, đó là các hàm xử lý các phiên giao dịch chuẩn, như:

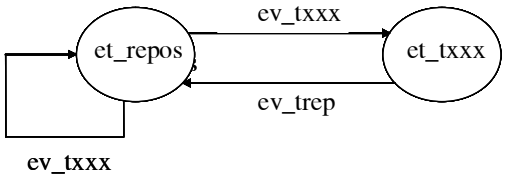
t_retrait: hàm xử lý yêu cầu rút tiền

t_depot: hàm xử lý yêu cầu gửi tiền

t_virement: hàm xử lý yêu cầu chuyển khoản

...

Khi ôtômát đang ở trạng thái et_repos mà nhận được sự kiện ev_txxx tức là có phiên giao dịch yêu cầu thực hiện (giả sử là yêu cầu rút tiền chẳng hạn) nó sẽ chuyển về trạng thái et_txxx đồng thời gọi đến hàm t_retrait để xử lý phiên giao dịch này (Hình 3).



Hình 3. Mô hình hoạt động của ôtômát xử lý phiên giao dịch

3. KẾT LUẬN

Với việc phát triển nhanh chóng của công nghệ thông tin, lĩnh vực Monetic cũng không ngừng phát triển, đòi hỏi việc xây dựng các phần mềm trung tâm quản lý các ứng dụng Monetic như hệ thống MA cũng không ngừng phát triển. Một trong các bộ phận không thể thiếu của những hệ thống như vậy là phần mềm điều khiển các phiên giao dịch ngân hàng cho máy rút tiền tự động ATM gọi là ATM Driver. Và với sự ra đời không ngừng của các loại máy ATM khác nhau làm cho thị trường máy ATM ngày càng đa dạng thì việc xây dựng một ATM Driver có cấu trúc 3 tầng gồm ôtômát trung tâm, các bảng Protocol và ôtômát xử lý phiên giao dịch là rất hợp lý. Với việc xây dựng các bảng Protocol khác nhau cho các máy ATM khác nhau sẽ giúp cho việc chuẩn hóa các ATM message và làm cho ôtômát trung tâm và ôtômát xử lý phiên giao dịch trở lên độc lập với các máy ATM khác nhau. Việc xây dựng cấu trúc phân tầng cho ATM Driver như trên cũng sẽ giúp ta dễ dàng nâng cấp các ATM Driver cho việc sử dụng các máy ATM mới.

TÀI LIỆU THAM KHẢO

[1] Andre Simon, *Automatas programables,* Hardcover, 1991.

[2] Goerge Pajari, *Writing device driver,* Addison - Wesley Publishing Company, Inc. 1992.

[3] John J.William, Clifford Williams, *Automatic Teller Machine III,* Consumertronics Co., 1997.

[4] Lê Hoài Anh, "Pilote GAB pour protocole NDC+", *Luận văn thạc sĩ,* Viện tin học sử dụng tiếng Pháp IFI, (1998).

Nhận bài ngày 10 - 1 - 2003
Nhận lại sau sửa 13 - 3 - 2003