LỜI CẢM ƠN

Trước tiên em xin được bày tỏ lòng biết ơn chân thành tới thầy giáo hướng dẫn, Ths Trần Ngọc Thái, Khoa Công nghệ thông tin trường Đại học Dân lập Hải Phòng đã tận tình hướng dẫn em trong suốt thời gian thực hiện đồ án tốt nghiệp.

Em cũng xin chân thành cảm ơn các thầy giáo, cô giáo Khoa Công nghệ thông tin trường Đại học dân lập Hải Phòng đã dạy và truyền đạt những kiến thức cần thiết và bổ ích trong suốt thời gian em học tập tại trường.

Cuối cùng em xin chân thành cảm ơn gia đình và tất cả bạn bè đã đóng góp ý kiến và hỗ trợ em trong quá trình thực hiện đồ án tốt nghiệp.

Hải Phòng, tháng 7 năm 2009

Trần Thị Thu Trang

MỤC LỤC

Chương 1:

Tổng quan về PDA và Hệ điều hành Windows CE

1.1 Tổng quan về PDA

1.1.1 Giới thiệu về các thiết bị PDA

Ngày nay Công nghệ thông tin đang ngày càng phát triển, có tác động ngày càng mạnh mẽ đến công việc cũng như cuộc sống của con người. Bắt đầu từ chiếc máy tính đồ sộ vào đầu thế kỷ 20 rồi đến chiếc máy vi tính và sau này là chiếc máy tính cá nhân (PC) đã tạo nên một cuộc cách mạng trên tất cả các lĩnh vực của cuộc sống và khoa học giúp cho con người tăng được đáng kể tốc độ và năng suất làm việc của mình trong công sở. Hơn thế nữa, sự ra đời của các thiết bị cầm tay (handheld devices) trong những năm đầu thập kỉ 80 còn tạo ra những điều kỳ diệu mới cho cuộc sống và công việc của con người. Việc phải mang 1 cái máy Fax cồng kềnh hay 1 quyển sách nhỏ để ghi số điện thoại và những công việc sẽ phải làm khi đi công tác đã khiến cho các nhà kinh doanh phải rất khó khăn trong việc liên lạc với thế giới xung quanh bằng những chiếc máy điện thoại cố định hay việc phải xử lý các công việc cần sự linh động hoặc với những công việc cần phải chia nhỏ để có thể làm việc với nó mọi lúc mọi nơi. Và công nghệ di động ra đời giúp giải quyết các vấn đề này. Các thiết bị tính toán di động có kiến trúc giống như máy để bàn hoàn toàn tương thích các phần mềm có sẵn và có thể làm việc không cần đến nguồn điện trực tiếp trong nhiều giờ liền. Và trong số những thiết bị đó thì PDA nổi lên như những đại diện mang đầy đủ các đặc tính thích hợp nhất để đáp ứng các nhu cầu của người sử dụng.

1.1.1.1 Khái niệm

PDA là gì? PDA - Personal Digital Assistant – là thiết bị trợ giúp kỹ thuật số cá nhân, được xây dựng trên nền tảng là một máy tính cá nhân bỏ túi đầy đủ phần cứng và phần mềm dễ dàng sử dụng, khởi động ngay lập tức khi bật máy, làm việc mọi lúc mọi nơi. Điều đã làm PDA trở nên gần gũi là sự kết hợp gắn bó giữa sức mạnh của máy tính để bàn và khả năng di chuyển cao của PDA. Giữa chúng có một mối liên hệ chặt chẽ qua hàng loạt các thiết bị phụ trợ trao đổi thông tin dễ dàng sử dụng.

Hình 1.1: Thiết bị PDA

1.1.1.2 Các thế hệ máy

Lần đầu tiên xuất hiện PDA chỉ đơn giản là một chiếc máy tính cầm tay với các ứng dụng cơ bản như đồng hồ, máy tính, danh bạ điện thoại, lịch làm việc, sổ địa chỉ…Chiếc máy PDA thật sự đầu tiên ra đời năm 1984 là Psion 1 từ hãng UK Technologies có kích thước 142mm x 78mm x 29.3mm, và chỉ nặng 225 grams, được làm từ công nghệ 8 bit, với 10K bộ nhớ, màn

hình 16-ký tự LCD, có một đồng hồ và lịch kèm theo một bộ các hàm tính toán toán học. Các hỗ trợ kèm theo chiếc máy này là các thư viện toán học và lập trình với OPL. Đến cuối thập niên 80 Psion 2 ra đời có 64K ROM, 32K RAM màn hình 4x20 kí tự. Thế hệ Psion Seria 3a ra đời vào năm 1993 được xây dựng trên nền tảng công nghệ 16 bit có màn hình 40 kí tự và 8 dòng LCD với bàn phím 58 phím. Đây là sự đột phá lớn của PDA khi nó có khả năng chuyển giao và đồng bộ hoá dữ liệu với máy tính để bàn. Cùng với sự phát triển của thị trường máy tính, năm 1997 Seria 5 ra đời với khả năng tính toán 32 bit đánh dấu bước ngoặc của PDA.

Phát triển từ thị trường của Psion, năm 1993, Apple ra đời sản phẩm Newton MessagePad. Việc nhập liệu bằng các bàn phím tí hon đã hạn chế rất nhiều sự phát triển của PDA. Do đó Apple đã cải tiến, áp dụng nhiều công nghệ mới như đưa ra công nghệ màn hình điều khiển trực tiếp bằng tay, và công nghệ nhận dạng chữ viết tay phát triển một cách nhanh chóng.

Tháng 3 năm 1995, Palm Pilot, một PDA được thiết kế để làm việc một cách hợp lý khi di chuyển, được Palm Computing Corp giới thiệu. Thay vì theo bước Apple trong việc tạo nhiều tính năng cho Newton, Palm quyết định gây sự chú ý của thị trường bằng một chiếc máy có tốc độ cao và hiệu quả dựa trên những tính năng cơ bản như việc ghi chú, quản lý các mối quan hệ, thời gian và công việc một cách tốt nhất. Palm Pilot với công nghệ nhận dạng chữ viết tay Graffiti đã trở nên thật sự phổ biến như là một chiếc máy tính bỏ túi với màn hình nhạy cảm có thể ghi lại những hoạt động hàng ngày của bạn và kết nối với PC. Palm Pilot đã trở thành chuẩn mực của thế hệ PDA thứ 2, có khả năng kết nối với PC, màn hình nhạy cảm, nhận dạng chữ viết tay. Các modul của Pilot được thiết kế cho phép dễ dàng gắn thêm hay gỡ bỏ các thiết bị phụ trợ để tạo dáng vẻ hấp dẫn như 1 thứ đồ trang trí. Nó nổi bật ở tính thiết thực, dễ sử dụng, và thoải mái khi di chuyển.

Năm 1997, Microsoft cho ra đời PDA đầu tiên chạy hệ điều hành Microsoft Windows CE. Những chiếc PDA đầu tiên này có hình dáng to

lớn, giống như 1 chiếc mini-laptop nhưng dần dần kích thước được thu nhỏ lại và được gọi là Handheld PC. Chiếc PDA đầu tiên dùng Windows CE không được sử dụng rộng rãi như Palm vì thiếu tính di động và quá phức tạp. Đến năm 2000, Microsoft đưa ra phiên bản PDA mới là Pocket PC. Pocket PC với giao diện thân thiện, dễ sử dụng, đòi hỏi ít thao tác hơn đã nhanh chóng được nhiều người sử dụng. Các thế hệ Pocket PC tiếp theo đã được trang bị phần cứng mạnh hơn và nhiều công nghệ mới đã thực sự trở thành thiết bị hỗ trợ cá nhân tiện lợi và trung tâm giải trí.

PDA sẽ sử dụng SD (Secure Digital) để phát triển tiềm năng trong tương lai. Thị trường PDA thật sự rất hứa hẹn. Kích thước nhỏ gọn và tiết kiệm điện năng, những lợi thế của PDA, tỏ ra rất phù hợp với việc truyền dẫn không dây và việc sử dụng máy dựa trên máy chủ. PDA sẽ ngày càng nhỏ và nhẹ hơn, thực hiện được nhiều chức năng hơn. Rất có thể trong tương lai PDA sẽ sử dụng năng lượng mặt trời, cho phép làm việc ngay cả khi đang di chuyển với việc truy nhập Internet không dây. Dữ liệu sẽ được đảm bảo hơn với việc lưu trữ từ xa. Trên thực tế, chúng ta đã thấy các thiết bị dùng công nghệ BlueTooth và WAP cho phép trao đổi thông tin, truy cập Internet không dây với các thiết bị BlueTooth khác mà không phải lo nghĩ gì về sự tương thích đang là một trở ngại ở các tia hồng ngoại đang dùng trong các máy PDA. Sự phát triển của Personal Area Network (PAN) của cơ quan nghiên cứu mối quan hệ giữa máy tính với con người của IBM (IBM Research's Human Computer Interaction) chia thành các bước khác nhau trong việc tái phát minh ra PDA. PDA có thế trở thành một phần của cơ thể con người, cho phép trao đổi, truy cập dữ liệu với những thao tác đơn giản hay truy cập Internet thông qua ý nghĩ có thể trở thành hiện thực trong tương lai gần khi mà con người có thể cấy ghép các thiết bị điện tử vào cơ thể. Sự tích hợp nhiều tính năng khác nhau trong PDA sẽ mang đến cho người sử dụng nhiều tiện như sử dụng PDA như là một thiết bị điều khiển từ xa tất cả các thiết bị trong nhà...Việc nhận dạng giọng nói và chữ viết cũng sẽ được cải tiến đáng kể.

1.1.1.3 Các thành phần

1.1.1.3.1 Màn hình

Kế thừa các tính năng ưu việt của công nghệ điện tử di động, PDA được trang bị màn hình tinh thể lỏng (TFT) tốt nhất, chịu đựng được môi trường rung và va đập, màu sắc và ánh trung thực, tiêu tốn ít năng lượng nhất. Hiện có hai công nghệ khác nhau để sản xuất màn hình. Thông thường là công nghệ chiếu sáng nền, các máy sử dụng công nghệ này cho phép người dùng dễ dàng quan sát các ứng dụng trên màn hình, nhưng tốn pin. Trong khi đó với công nghệ màn hình phản chiếu, dù vẫn có chiếu sáng nền nhưng máy chỉ hoạt động tốt khi ở ngoài trời hoặc những nơi có ánh sáng tốt. Pocket PC có kích thước màn hình chuẩn là 320x240. Trong khi đó các máy Palm có độ phân giải đa dạng hơn: từ độ phân giải căn bản là 160x160 pixel và tối đa là 320x480 pixel.

1.1.1.3.2 Pin

Hầu hết các máy sử dụng pin có thể nạp lại, tiêu biểu trong số này có pin lithium-ion, là loại pin có hiệu suất cao nhất hiện nay, nhưng đa số cá loại máy đơn sắc và các model rẻ tiền đều dùng pin AAA. Với các máy có màn hình đơn sắc, có thể dùng hơn một tháng mới hết pin, trong khi chỉ dùng được khoảng hơn 10 giờ đối với các loại máy có màn hình màu.

1.1.1.3.3 Nhận dạng chữ viết tay và nhập dữ liệu

Đây là một trong những yếu tố quyết định của PDA, công nghệ này xây dựng dựa trên việc người dùng sử dụng cây bút gọi là stylus viết trực tiếp lên màn hình và PDA nhận dữ liệu chuyển chúng thành các văn bản hoặc lưu trữ chúng giống như các cuốn sổ tay điện tử, công nghệ Graffiti được ứng dụng rộng rãi. Người dùng cũng có thể nhập liệu bằng một bàn phím vật lý nhỏ được thiết kế rời hay bằng bàn phím ảo (Onscreen Keyboard) trên màn hình cảm ứng.

Hình 1.2: Bàn phím ảo của Pocket PC

1.1.1.3.4 Liên lạc, kết nối

Phụ thuộc vào các hãng sản xuất và model. Các PDA hiện nay đều có thể kết nối với nhau hoặc với PC qua cổng serial, hồng ngoại, modem trong kết nối qua đường điện thoại và cả điện thoại di động. IrDA và Bluetooth sẽ là giao tiếp chuẩn cho PDA.

1.1.1.3.5 Thiết bị mở rộng

Hầu hết các PDA đều có khe cắm mở rộng dùng cho việc nâng cấp bộ nhớ hay mở rộng chức năng như modem, wire Ethernet và Wifi, máy ảnh số. Các PDA thông thường dùng SD card (Security Digital) và một số ít khác dùng CF (Compact Flash) hoặc có cả hai loại.

1.1.1.3.6 Hệ điều hành

Các PDA sử dụng hệ điều hành Palm chiếm tỉ lệ lớn, được số lượng ngày càng tăng với các sàn phẩm của Sony, IBM, Handspring...Từ các phiên bản hệ điều hành nhúng Windows CE ban đầu, Microsoft đã cải tiến và cho ra đời hệ điều hành Pocket PC với nhiều cải tiến và đang dần được sử dụng rộng rãi trong các PDA. EPOC là hệ điều hành truyền thống trên PDA của Psion chiếm 70% thị trường (1999). EPOC với những ưu điểm đã được sự ủng hộ mạnh mẽ của Nokia, Motorolla, Erisson và Symbian đang hoà nhập PDA và điện thoại di động qua hàng loạt các công nghệ không dây tiên tiến. Một số ít PDA được cài đặt hệ điều hành Linux.

1.1.1.3.7 Các phần mềm ứng dụng

Bộ phần mềm quản lý thông tin cá nhân, còn gọi là PIM (Personal Information Management), là linh hồn của PDA, bao gồm các chương trình nhỏ về các công việc chủ yếu như: lập lịch làm việc, danh bạ điện thoại, ghi chú, thư điện tử. Ngoài ra còn có nhiều ứng dụng tiện ích khác như: quản lý tập tin, đồng hồ, máy tính, soạn thảo văn bản, bảng tính, phần mềm tài chính, Từ điển… Ngoài ra còn có hàng loạt sản phẩm phần mềm về các công việc chuyên môn được viết riêng cho PDA như tìm đường bằng GIS kết hợp GPS, thu thập các số liệu điều tra hiện trường, điều khiển các dây chuyền sản xuất...

1.1.2 Các hạn chế của PDA

Các PDA được thiết kế nhỏ gọn, tiết kiệm pin nên có nhiều hạn chế như dung lượng bộ nhớ nhỏ, tốc độ xử lý chậm, tương tác người dùng không tiện lợi. Các hạn chế này gây nhiều khó khăn cho việc phát triển ứng dụng phần mềm cho nó.

Tuy vậy, với nhiều tiện lợi, tính nhỏ gọn, các PDA vẫn đang ngày càng được sử dụng rộng rãi. Đối với nhiều người, PDA là thiết bị không thể thiếu để quản lý thông tin cá nhân, lập lịch làm việc, hỗ trợ công việc, giải trí…

1.2 Một số hệ điều hành nhúng cho thiết bị PDA

Khi nói đến thiết bị máy tính thì ta không thể không nhắc đến các hệ điều hành được sử dụng trên thiết bị đó. Hệ điều hành của máy tính được ví như dòng máu chảy trong cơ thể của một con người. Nếu không có hệ điều hành thì máy tính không thể vận hành được. Các thiết bị PDA cũng vậy. Hầu hết các PDA sử dụng một trong 3 hệ điều hành: Windows CE (Microsoft), EPOC (Symbian), PalmOS. Đã bắt đầu có một số sản phẩm PDA được giới thiệu cùng với Linux. Trong đó, Windows CE và EPOC là hai hệ điều hành được sử dụng nhiều nhất trong các thiết bị PDA.

1.3 Tổng quan về hệ điều hành Windows CE

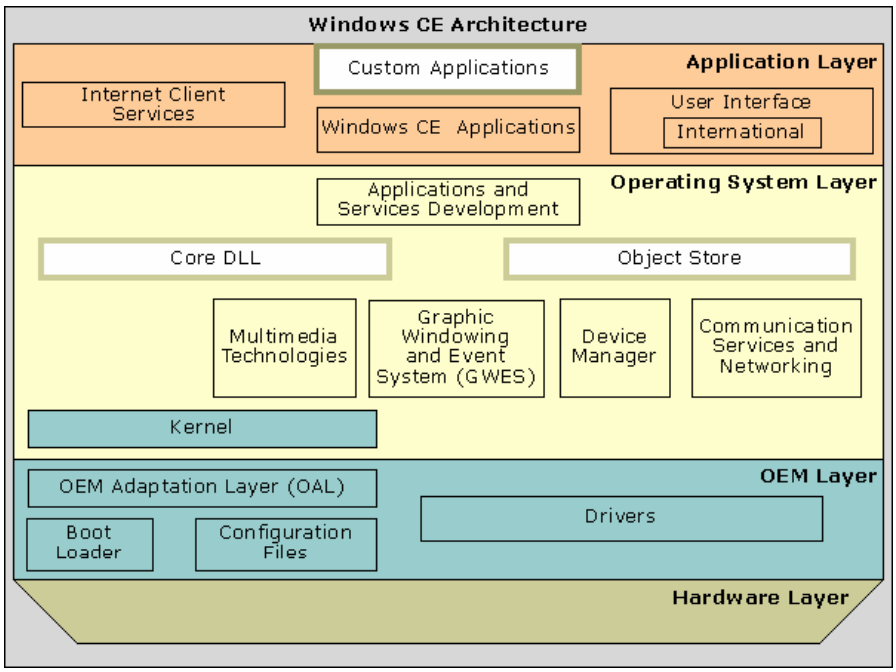
Hình 1.3: Biểu tượng của Windows CE

1.3.1 Giới thiệu

Windows CE là một hệ điều hành nhúng do Microsoft phát triển năm 1996, được tích hợp vào các thiết bị giải trí, các máy subnotebook, máy tính cầm tay (handheld PC, palm-size PC…); các điện thoại di động; các hệ thống thông tin, giải trí trên xe hơi (AutoPC); cũng như các thiết bị công nghiệp, …

Do được thiết kế như là một phiên bản hệ điều hành Windows 32 bit thu nhỏ, Windows CE rất quen thuộc đối với các hãng phát triển phần mềm, các lập trình viên cũng như đối với người sử dụng Windows. Windows CE là một trong hai hệ điều hành nhúng chiếm thị phần cao nhất hiện nay.

1.3.2 Đặc điểm



Hình 1.4: Kiến trúc của hệ điều hành Windows CE .Net

1.3.2.1 Tính nhỏ gọn

Đây là đặc điểm quan trọng nhất của hệ điều hành Windows CE. Mục đích của việc tạo ra một hệ điều hành Windows CE nhỏ gọn là để giảm bớt những phần cứng cần thiết (như RAM, ROM, CPU và vô số các thành phần khác) sao cho phù hợp với những thiết bị điện tử giá thành rẻ, tính năng cao chẳng hạn như PDA, … Hệ điều hành Windows CE nhỏ nhất chỉ dưới 500K (không có màn hình hiển thị và các trình điều khiển thiết bị). Mặc dù nhỏ gọn nhưng Windows CE thực sự là một hệ điều hành giàu tính năng và có thể cấu hình lại.

1.3.2.2 Tính khả điều chỉnh lại

Windows CE là một hệ điều hành có tính "lắp ráp", có thể điều chỉnh lại. Không giống như phiên bản Windows trên desktop được phát triển như là một tập cố định các tập tin, Windows CE được tạo nên từ các *module* (là các tập tin chương trình .exe, và các tập tin thư viện .dll), và một số module này được tạo ra từ hai hay nhiều *component* (bao gồm các hàm API hay các tính năng của hệ điều hành).

Để tạo ra một phiên bản Windows CE đáp ứng một mục đích sử dụng nào đó (như để tích hợp vào một thiết bị mới), những nhà phát triển có thể sử dụng công cụ *Platform Builder* của Microsoft để điều chỉnh lại hệ điều hành bằng cách thêm hay bớt các module khác nhau.

1.3.2.3 Tính khả chuyển đổi

Cũng giống như phiên bản Windows trên desktop, hầu hết các chương trình ứng dụng lẫn các trình điều khiển thiết bị (hai thách thức chính dẫn đến sự thành công của một hệ điều hành) của Windows CE đều được xây dựng trên nền tảng giao diện lập trình Win32 API . Hơn nữa, phần lớn chúng được kế thừa, đơn giản hóa từ phiên bản hệ điều hành Windows trên desktop. Do đó, có thể chuyển mã nguồn từ desktop sang Windows CE, cũng như có thể chuyển mã nguồn giữa các thiết bị được xây dựng trên các CPU khác nhau nhưng cùng sử dụng hệ điều hành Windows CE.

1.3.2.4 Tính tương thích

Thông thường thì một hệ điều mới luôn duy trì tính tương thích với các hệ điều hành trước nó. Windows CE không phải là một trường hợp ngoại lệ. Để đạt được điều này thì tính chuyển đổi của Windows CE được nâng lên một bước, đó là có thể *chia sẻ* mã nguồn giữa desktop và các thiết bị thông minh càng dễ dàng càng tốt.

Hơn nữa, tính tương thích của Windows CE còn thể hiện ở việc tạo các giao diện lập trình có tính tương thích, nghĩa là giữ cho các giao diện lập trình trên thiết bị càng nhất quán với trên desktop càng tốt. Chẳng hạn, mặc dù Windows CE hỗ trợ một số lượng các hàm Win32 ít hơn desktop nhiều nhưng tất cả những hàm được hỗ trợ có thể thực hiện những công việc tương đương trên desktop càng nhiều càng tốt.

1.3.2.5 Tính kết nối

Windows CE làm cho các thiết bị thông minh có thể kết nối tới các thiết bị dùng hệ điều hành Windows CE khác, tới các mạng cục bộ (cả kết nối có đường dẫn lẫn kết nối không dây), và kết nối vào mạng Internet. Hơn nữa, các thiết bị chuyên biệt cho Windows CE còn có thể kết nối tới các mạng cá nhân (PAN – Personal Area Network), các mạng nội bộ (LAN – Local Area Network), và các mạng diện rộng (WAN – Wide Area Network).

Khi đề cập đến kết nối thì tính bảo mật luôn có tầm quan trọng nhất. Do đó, các thiết bị chuyên biệt cho Windows Ce cũng cho phép thiết lập các kết nối riêng, an toàn, bảo mật tới một mạng LAN thành viên ở xa qua Internet sử dụng giao thức Point – to – Point Tunneling Protocol (PPTP) để thiết lập một mạng riêng ảo có tính bảo mật (Virtual Private Network – VPN). Ngoài ra, Windows Ce còn cung cấp các tính năng khác cho việc truyền thông an toàn trên mạng như: Secure Socket Layer (SSL), hỗ trợ Cryptography API; xác nhận Kerberos and NTLM, và hỗ trợ tường lửa IP. Nói chung, khi có mối quan hệ client/server thì Windows Ce hỗ trợ kết nối *ở phía client.*

1.3.2.6 Hỗ trợ phát triển hệ thống thời gian thực

Bắt đầu từ phiên bản Windwos CE 3.0, thì Windows CE được tích hợp một tập các tính năng quan trọng để hỗ trợ cho việc phát triển các hệ thống thời gian thực như: hỗ trợ 256 độ ưu tiên cho tiến trình (Windows CE luôn hỗ trợ lập trình đa tiến trình), hỗ trợ các yêu cầu ngắt lồng nhau. Có thể nói Windows CE là hệ điều hành hỗ trợ mạnh các tính năng về thời gian thực như:

- Đảm bảo các chặn trên cho việc lập lịch tiến trình có độ ưu tiên cao – chỉ đối với tiến trình có độ ưu tiên cao nhất trong tất cả các tiến trình được lập lịch.
- Đảm bảo chặn trên trễ cho việc thực hiện các chuỗi dịch vụ ngắt có độ ưu tiên cao (ISRs – Interrupt Service Routines). Nhân hệ điều hành có một vài nơi ở đó các ngắt bị khóa trong một khoảng thời gian ngắn, có giới hạn.
- Kiểm soát chặt chẽ bộ lập lịch và cách mà nó lập lịch các tiến trình.

1.3.3 Một số phiên bản của Windows CE

Hiện tại, có khá nhiều sự lẫn lộn quanh các phiên bản của Windows CE cũng như cách gọi tên. Sau đây là một vài phiên bản hiện tại của Windows CE:

- Windows CE .NET 4.2: Phiên bản mới nhất hiện nay cung cấp nhiều hàm thư viện hơn nhưng đòi hỏi cấu hình phần cứng cao hơn. Một trong những tính năng mới của Windows CE .NET là tích hợp sẵn .NET Compact Framework, cho phép phát triển ứng dụng không phụ thuộc phần cứng và hệ điều hành.
- Windows CE 3.0: Phiên bản này được thiết kế để cung cấp các đặc tính của một hệ điều hành thời gian thực và một số phát triển khác. Thiết bị Pocket PC (phiên bản 2002 trở về trước) sử dụng một dạng biến thể phiên bản này.
- Windows CE 2.12: Được sử dụng chủ yếu bởi các nhà sản xuất thiết bị nhúng dùng Microsoft Platform Builder.

- Windows CE 2.21: Phiên bản của Windows CE dùng cho các thiết bị Windows Handheld và Palm-size.

1.3.4 Các biến thể của Windows CE

Hiện nay, Windows CE có nhiều biến thể cho phù hợp với từng loại thiết bị PDA: Handheld PC, Pocket PC, SmartPhone. Trong năm 2003, Microsoft đã cho ra đời hai phiên bản biến thể của Windows CE là Pocket PC 2003 và Smartphone 2003. Cũng trong năm 2003 một biến thể khác của Windows CE là Windows mobile ra đời. Một số phiên bản của Windows mobile:

- Windows mobile 2003:được phát hành vào 23/6/2003, dựa trên windows CE 4.2. cung cấp nhiều hàm thư viện hơn nhưng đòi hỏi cấu hình phần cứng cao hơn, tích hợp sẵn .NET Compact Framework, cho phép phát triển ứng dụng không phụ thuộc phần cứng và hệ điều hành.
- Windows mobile 2003 SE: được phát hành vào 24/3/2004, đây là phiên bản cuối cùng cho phép người dùng sao lưu và khôi phục toàn bộ một thiết bị thông qua ActiveSync.
- Windows mobile 5:được phát hành 12/5/2005.
- Windows mobile 6: được phát hành vào 12/2/2007, được hỗ trợ bởi windows CE 5.0
- Windows mobile 6.1: được công bố 1/4/2008, nâng cấp từ windows mobile 6, mang tính năng nâng cao hiệu suất.

Chương 2:

Tổng quan về Pocket PC và môi trường lập trình .Net Compact Framework

2.1 Tổng quan về Pocket PC

2.1.1 Giới thiệu

Khi nói về Pocket PC ta cần phân biệt hai khái niệm. Đó là hệ điều hành Pocket PC (Pocket PC Operating System) và thiết bị Pocket PC (Pocket PC– device).

- Hệ điều hành Pocket PC: là một phiên bản của hệ điều hành Windows CE cho các thiết bị di động được Microsoft giới thiệu vào đầu năm 2000.
- Thiết bị Pocket PC: là một PDA, là một thiết bị cầm tay (palm–size) sử dụng hệ điều hành Pocket PC.

Một thiết bị Pocket PC luôn đi kèm với một phiên bản của hệ điều hành Pocket PC.

2.1.2 Hệ điều hành Pocket PC

Hệ điều hành Pocket PC được thiết kế với các tính năng và giao diện dành riêng cho các thiết bị hỗ trợ cá nhân PDA và máy tính cầm tay (handheld PC). Có thể hiểu hệ điều hành Pocket PC là một phiên bản của hệ điều hành Windows CE được cài đặt nhằm tối ưu cho các thiết bị này.

Hệ điều hành Pocket PC giải quyết được nhiều thiếu sót đã làm giảm thành công của hệ điều hành Windows CE, như giao diện quá phức tạp, tốc độ chậm, khả năng lưu trữ kém, nguồn cung cấp năng lượng không tốt, …

Phiên bản mới nhất là hệ điều hành Pocket PC 2003, một thể hiện của hệ điều hành Windows CE 4.2.

Hình 2.1: Giao diện Pocket PC 2003

2.1.3 Thiết bị Pocket PC

Ra đời vào những năm 90 của thế kỉ 20, thiết bị Pocket PC là một dạng thiết bị PDA sử dụng hệ điều hành Pocket PC. Với thiết bị Pocket PC thì mục đích của Microsoft là tạo ra một máy tính đa năng mà có thể đặt trong lòng bàn tay. Các thiết bị Pocket PC do nhiều hãng như Compag, HP, Casio sản xuất. Thiết bị Pocket PC có tốc độ và bộ nhớ gấp vài lần so với các thiết bị sử dụng hệ điều hành PalmOS. Chúng cũng có độ phân giải màn hình lớn hơn (320x240) và gần như hiển thị được tất cả các màu. Khả năng thể hiện và ghi âm đã trở thành chuẩn. Hơn nữa, thiết bị Pocket PC sử dụng các chuẩn cắm công nghiệp, có tính tương thích và có một số hình thức thêm các phần (module) mở rộng (thường dưới hình thức các thẻ Compact Flash) như các thẻ nhớ (storage card), hay modem kết nối Internet, …Tất cả các hệ thống

có thể giao tiếp qua cổng hồng ngoại (IR– Infrared), và chúng cũng có thể kết nối với desktop qua cổng USB dùng cáp nối.

Microsoft đã thêm các tính năng mới cho thiết bị Pocket PC với hệ điều hành Pocket PC (lưu trong ROM) chứa phiên bản thu gọn của một số phần mềm như: Pocket Internet Explorer, Pocket Word và Excel, Outlook, Microsoft Reader, Media Player, File Manager, Notepad và Calculator, …cũng như gói phần mềm ActiveSync giữa thiết bị Pocket PC và các máy trạm. Các phần mềm khác phải được lưu trong Systems RAM (đóng vai trò là sự kết hợp giữa hệ thống tập tin và bộ nhớ hỗn tạp). Microsoft cũng cung cấp một số phần mềm miễn phí như: Pocket Streets, Transcriber (bộ giải mã chữ viết tay) và Games (như Freecell). Nhiều phần mềm khác của hãng thứ ba cũng được tích hợp sẵn như: Databases, Picture Viewers, …

Với tất cả những tính năng trên thì Pocket PC thực sự là một trong những thiết bị PDA được sử dụng phổ biến nhất ở Việt Nam hiện nay, cũng như trong tương lai. Nhưng do điều kiện không cho phép nên ứng dụng Từ điển chỉ được phát triển trên Pocket PC 2002.

Hình 2.2: Một số thiết bị Pocket PC

2.2 Một số công cụ phát triển trên Pocket PC 2002

Là một lập trình viên thì khi tìm hiểu một thiết bị mới, cũng như một hệ điều hành mới, vấn đề được quan tâm nhất chính là khả năng lập trình, phát triển ứng dụng trên thiết bị, hệ điều hành đó. Đối với Pocket PC 2002 thì hiện nay tất cả các phần mềm hay ứng dụng đều được phát triển bằng hai công cụ chính là:

- Microsoft eMbedded Visual C++ 3.0.
- .Net Compact Framework.

Mỗi công cụ đều có những điểm mạnh, yếu đặc trưng của nó. Vì vậy, khi phát triển ứng dụng trên Pocket PC cần xem xét, cân nhắc việc kết hợp giữa 2 bộ công cụ này.

2.3 Công cụ lập trình Microsoft eMbedded Visual C++ 3.0

2.3.1 Khái quát

Microsoft eMbedded Visual Tools 3.0 là một môi trường "tất cả trong một" cho sự phát triển cơ sở Windows CE (tương tự như bộ Visual Studio). Nó cuộn sang một gói đơn tất cả hỗ trợ mà thường đòi hỏi 4 sản phẩm riêng rẽ. Không giống như sản phẩm phát triển chương trình trước đó của Windows CE, nó không đơn giản một add-on vào công cụ tồn tại cho Visual C++ và Visual Basic. Thay thế vào đó, nó cung cấp tất cả những gì bạn cần từ cả 2 môi trường đơn và gói độc lập.

2.3.2 Một số đặc điểm nổi bật

Microsoft eMbedded Visual C++ 3.0 là một chương trình mạnh nhất cho các nhà lập trình xây dựng chương trình phầm mềm ứng dụng cho các thiết bị sử dụng Windows CE. IDE đứng độc lập mang đến một mức độ mới cho sản phẩm cho sự phát triển Windows CE, không có sự thỏa hiệp mềm dẻo nào, thực thi, hoặc kiểm soát.

Với eMbedded Visual C++, các nhà phát triển có thể đạt được các điều sau:

- Có một sự thuận lợi của một môi trường phát triển quen thuộc bằng việc xây dựng các chương trình ứng dụng trên Windows CE sử dụng bộ tích hợp điện tử độc lập được thiết kế nhắm tới sự phát triển Windows

CE.

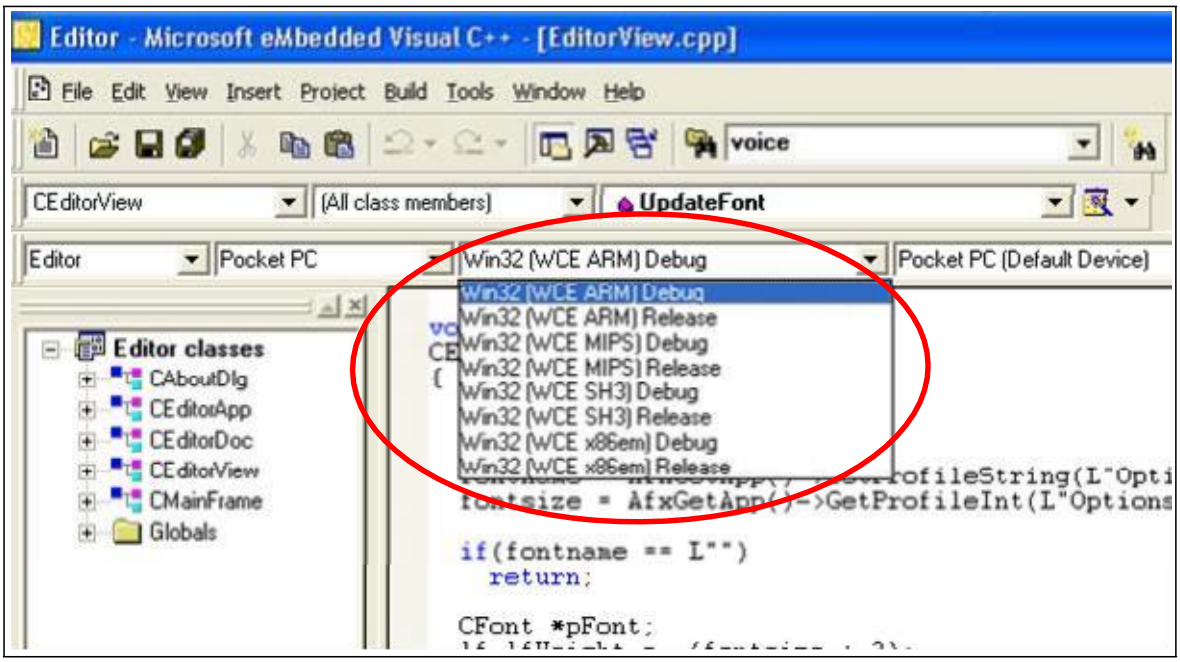
- Truy nhập Windows CE - cung cấp những tài liệu cụ thể nhằm tạo ra các bộ phát triển phần mềm nền mà bạn đã cài đặt ở nơi làm việc của bạn.
- Tiết kiệm thời gian và tiền bạc bằng việc sử dụng các phiên bản Windows CE của Microsoft Foundation Classes và Active Template Library.
- Xây dựng những giải pháp công nghệ với các khả năng qua các ADO cho Windows CE, xử lý các tác vụ qua MTS - Microsoft Transaction Server, và sự tích hợp gần gũi hơn với các dịch vụ của hệ điều hành Windows CE.

Đạt tới sự truy nhập trực tiếp vào các tính năng dưới hệ điều hành không cần sự mã hóa thêm vào, cung cấp điều khiển đầy đủ trên các thiết bị phần cứng và hệ điều hành chủ. Truy nhập vào mọi tính năng của mọi sự hoán vị của hệ điều hành Windows CE để xây dựng nhanh nhất, thiết thực nhất cho các chương trình ứng dụng Windows CE. Là công cụ đầu tiên để lập trình cho các thiết bị mới nhất và thú vị nhất với Windows CE, sử dụng các giả lập SDK Windows CE cho eMbedded Visual C++. Tham gia lập trình ngay từ đầu và có thể xây dựng các chương trình tốt cho hệ điều hành tiếp theo. Mở rộng các lựa chọn phát triển hướng tới toàn bộ những nhóm người dùng mới và trang bị những chương trình tương tự như trong máy tính để bàn nhưng chạy trong Windows CE như trình duyệt Internet, các xử lý giao dịch công việc cụ thể (task-specific business processes), hoặc chương trình giải trí. Xây dựng các chương trình phục vụ sự lưu động cao, với tính năng có thể truy nhập từ xa dữ liệu lưu trữ và truyền tải với các mạng chủ.

Với những tính năng như vậy thì Microsoft eMbedded Visual C++ 3.0 thường được sử dụng để phát triển các ứng dụng đòi hỏi phải can thiệp sâu xuống hệ thống, có tốc độ xử lý đặc biệt.

Tuy nhiên, việc sử dụng Microsoft eMbedded Visual C++ 3.0 tương

đối rắc rối và phức tạp. Hơn nữa, đây là một công cụ phát triển phụ thuộc thiết bị và hệ điều hành, nghĩa là muốn ứng dụng chạy được trên nền nào thì ứng dụng phải được biên dịch chính xác trên nền đó. Đó là một khó khăn đối với những người bắt đầu làm quen với việc lập trình trên Pocket PC nói riêng và trên các thiết bị PDA nói chung.



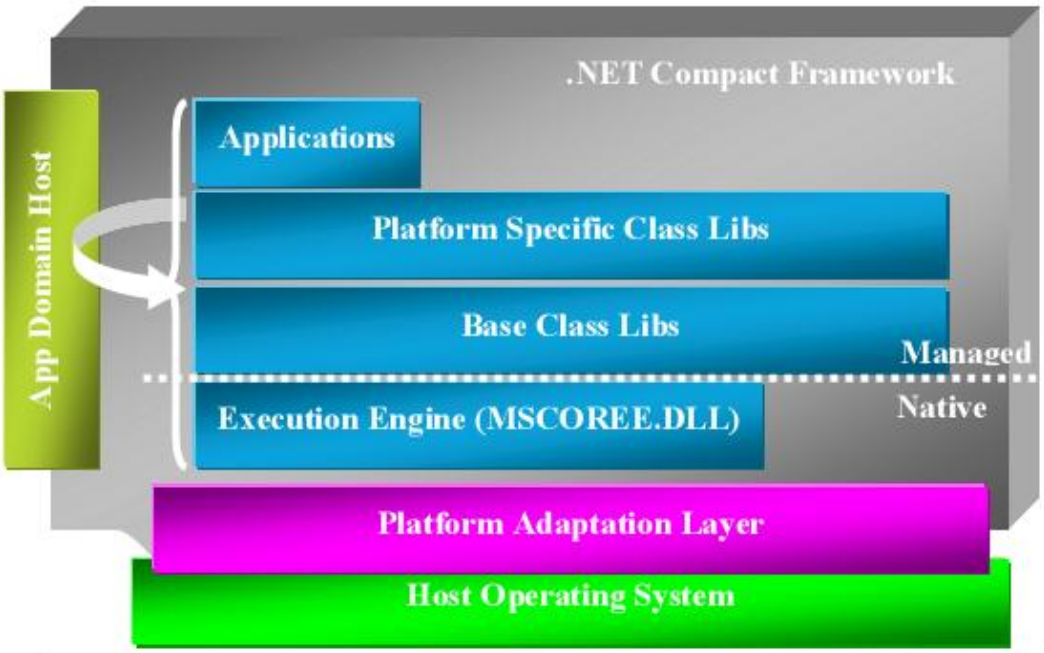
Hình 2.3: Sự phụ thuộc thiết bị khi lập trình với eVC++ 3.0

2.4 Môi trường lập trình .Net Compact Framework

2.4.1 .Net Compact Framework là gì

Net Compact Framework (.Net CF) là một giao diện lập trình, một thư viện thực thi được tạo ra như là sự kết hợp giữa hai công nghệ của Microsoft: *WindowCE* và *.Net.*

Nói cách khác, .Net Compact Framework là một tập con của .Net Framework. Nó bao gồm các thư viện lớp cơ sở và có thêm một số thư viện chuyên cho việc phát triển trên các thiết bị PDA. .Net Compact Framework được thiết kế để cho phép các ứng dụng .Net chạy được trên tất cả các thiết bị PDA mà không lệ thuộc hệ điều hành.



Hình 2.4: Kiến trúc của .Net Compact Framework

2.4.2 Một số đặc điểm của .Net Compact Framework

2.4.2.1 Độc lập với thiết bị và hệ điều hành

Hoạt động gần giống nguyên tắc với một máy ảo nhưng được thiết kế để tận dụng tối đa tài nguyên của thiết bị nhúng, .NET Compact Framework cho phát triển một ứng dụng .NET *viết một lần, chạy ở mọi nơi*.

Cần lưu ý rằng do các máy PDA sử dụng rất nhiều chủng loại CPU khác nhau như ARM4, ARM4I, Xscale… các chương trình cũng cần được biên dịch thành nhiều tập tin .EXE ứng với mỗi chủng loại CPU. Khi cài đặt, chương trình đóng gói *setup* sẽ kiểm tra và chép tập tin .EXE phù hợp với loại CPU được sử dụng trong thiết bị. Đối với ứng dụng .NET, chương trình có thể chạy trên bất kỳ loại CPU nào.

Hiện tại .NET Compact Framework chỉ mới được viết cho các máy PDA sử dụng hệ điều hành Windows CE1. Kế hoạch xây dựng .NET CF cho các hệ điều hành nhúng khác đang được xây dựng, bắt đầu từ Embedded Linux. Đây là một trong những ưu điểm nổi trội của việc chọn lựa .NET Compact Framework để phát triển ứng dụng trên Pocket PC.

2.4.2.2 Được xây dựng dựa trên những thừa hưởng từ .Net Framework

.NET Compact Framework (.NET CF) là thư viện .NET Framework đã được Microsoft thiết kế lại để chạy hiệu quả trong điều kiện tài nguyên giới hạn của các thiết bị PDA.. Nó bao gồm 18 thư viện với dung lượng khoảng 2,5MB trong khi .Net Framework phiên bản 1.1 gồm 86 thư viện chiếm khoảng 40MB.

Tất cả các thành phần chuẩn "phổ biến" của .Net Framework trên desktop đều có thể tìm thấy trong .Net Compact Framework.

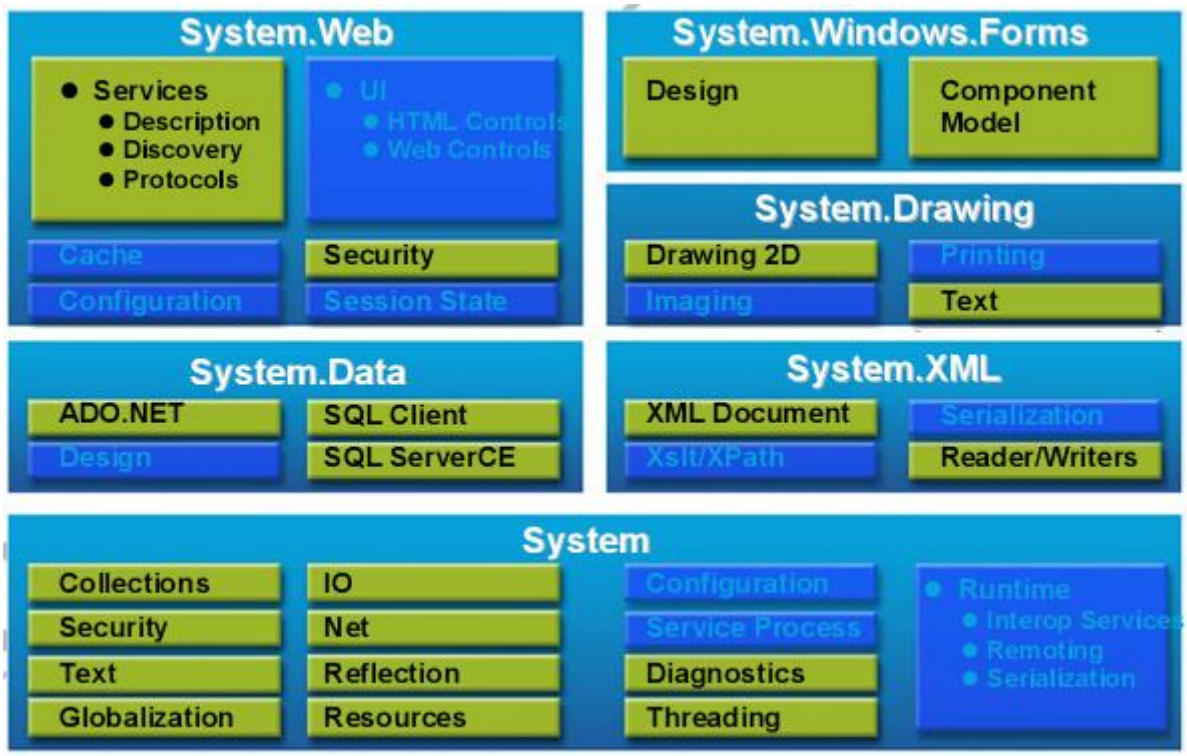
Các tập tin thực thi sử dụng tập chỉ dẫn CIL để đưa chúng vào bộ nhớ và được biên dịch JIT (Just In Time - cơ chế chỉ biên dịch các phần cần thiết để chạy chương trình) sang những chỉ dẫn ngôn ngữ máy bởi cơ chế thực thi ngôn ngữ chung (CLR - Common Language Runtime) của .Net Compact Framework.

.Net CF cũng cung cấp cơ chế quản lý bộ nhớ giống như trên desktop như: cấp phát bộ nhớ, quản lý bộ nhớ Heap, và cơ chế dọn rác tự động.

Compact Framework cũng hỗ trợ tất cả các chuẩn trong nền tảng ngôn ngữ chung (CLI – Common Language Infrastructure), gồm hệ thống kiểu chung (Common Type System) cũng như đặc tả kiểu chung (Common Language Specification). Mặc dù bộ khung này không phong phú bằng trên desktop, nhưng các thao tác trong thời gian thực thi cơ bản là chính xác.

2.4.2.3 Duy trì sự nhất quán với Desktop

Các lập trình viên đã quen với các ứng dụng .Net trên desktop sẽ cảm thấy gần gũi với .Net Compact Framework vì cả hai bộ khung chia sẻ tất cả các kiểu giá trị cơ bản, hầu hết các không gian tên (namespaces) và nhiều lớp chung.

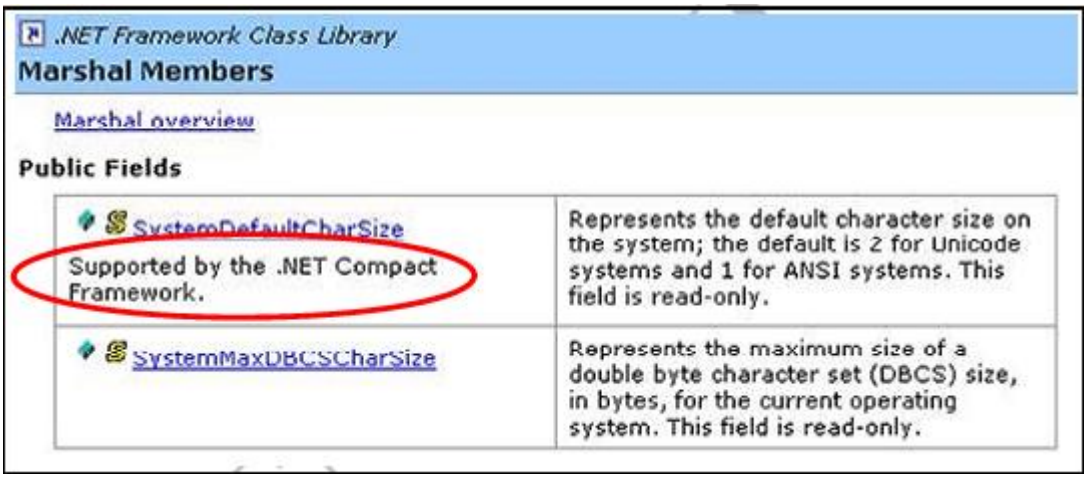


Hình 2.5: Không gian tên và các lớp của .Net Compact Framework

Tuy nhiên, hầu hết các lớp trên .Net Compact Framework có số thuộc tính, phương thức, sự kiện ít hơn. Đây cũng chính là một khó khăn cho các lập trình viên khi tiếp cận với .Net Compact Framework. Họ phải mất thời gian để thích nghi với "sự thiếu sót".

Các ứng dụng .Net CF cũng được xây dựng trên môi trường Visual Studio .Net như các ứng dụng .Net. Một chương trình viết bằng .Net CF có thể chạy được trên desktop mà không cần bất kỳ sự thay đổi nào.

Tính nhất quán cao của chúng cho phép trên cùng một tài liệu hỗ trợ cả hai bộ khung. Các thành phần nào của lớp được hỗ trợ trên .Net CF sẽ được ghi chú là "Supported by the .Net Compact Framework".



Hình 2.6: Ghi chú hỗ trợ .Net Compact Framework trong MSDN

2.4.2.4 Chạy tốt trên các thiết bị nhúng di động

Hai thách thức chính khi phát triển các ứng dụng trên các thiết bị PDA là kích thước và tốc độ.

Vấn đề đặt ra là làm sao để có thể sử dụng được bộ thư viện đồ sộ 25+ MB của desktop Framework trên các thiết bị di động mà khả năng lưu trữ rất giới hạn (từ 32MB đến 64MB). Hướng giải quyết được đưa ra là rút gọn desktop Framework từ 25+ MB xuống chỉ còn 2MB bằng nhiều cách khác nhau:

- Nếu có hai hay nhiều cách để thực hiện một tác vụ nào đó thì hầu hết chúng được loại bỏ.
- .Net CF loại bỏ các tính năng trung tâm trên desktop như kéo thả (drag- and-drop), các control phức tạp như RichTextBox (.Net CF chỉ hỗ trợ một phần các Control trên desktop. Xem Phụ lục A). Với các Control được hỗ trợ thì chỉ những tính năng cơ bản nhất được giữ lại, một tập con "PMEs" – Properties, Methods và Events.

Vấn đề còn lại là về mặt tốc độ. Thậm chí những CPU nhanh nhất trên các thiết bị nhúng di động cũng chậm hơn các CPU trên một hệ thống desktop trung bình. Để đạt được những yêu cầu về mặt tốc độ, .Net CF được đo đạc và chuyển đổi sao cho các control chủ yếu dựa trên Win32 (native control). Tuy nhiên do các control Win32 ở dạng unmanaged code mà

Compact Framework code lại chạy dưới dạng managed code nên cũng phải tốn chi phí để vượt qua ranh giới giữa managed và unmanaged code. Vì vậy để tăng tốc độ của các control thì .Net CF chỉ chấp nhận một tập con các thông điệp Win32 được kiểm soát chặt chẽ.

2.4.2.5 Thể hiện phong phú trên các Platform khác nhau

Trong mọi trường hợp, Compact Framework dựa trên các native control để thực hiện phần chính của công việc. Điều này rõ ràng là đem lại những lợi ích về mặt kích thước và tốc độ. Hơn nữa, ứng với mỗi Platform nó có còn cung cấp một chuẩn giao diện (look-and-feel) đáng tin cậy cho các ứng dụng Compact Framework trên đó.

Ví dụ, lớp MainMenu của .Net CF sẽ cung cấp các thể hiện khác nhau ứng với từng Platform. Đối với các ứng dụng trên Pocket PC, trình đơn này xuất hiện ở phía dưới của của sổ. Còn trên các thiết bị Windows CE.NET không phải là Pocket PC (non-Pocket PC), cũng với các ứng dụng đó nhưng trình đơn lại xuất hiện phía trên của cửa sổ giống như các ứng dụng trên Windows CE.NET chuẩn. Mặc dù sự cài đặt bên dưới của trình đơn Win32 là khác nhau nhưng các nhà phát triển không thể thấy được sự khác nhau đó thông qua các lớp *MainMenu* và *MenuItem* của .Net CF.

2.4.2.6 Duy trì chuẩn giao diện (look-and-feel) của từng Platform

Vấn đề nảy sinh từ những khác nhau không rõ ràng trong cách cài đặt các native control. Có nhiều sự khác nhau rất nhỏ, hầu như không thể nhận thấy được như thêm một pixel ở đây, bớt một pixel ở kia. Những khác nhau như vậy là không thể thấy được khi nhìn vào hai ứng dụng chạy trên hai thiết bị khác nhau. Nhưng khi hai ứng dụng chạy trên cùng một thiết bị thì những khác nhau đó trở nên rõ ràng. .Net CF đảm bảo rằng các chuẩn look-and-feel trên các Platform khác nhau được duy trì. Còn đối với người sử dụng thì không thể phân biệt được một cách trực quan giữa ứng dụng Compact Framework và các ứng dụng không Framework.

2.4.3 Một số hạn chế của .Net Compact Framework:

Như đã nói, .Net Compact Framework là một tập con, được đơn giản hóa từ .Net Framework trên Desktop. Do đó, việc thiếu sót một số đặc tính sẽ gây ra những khó khăn nhất định đối với các lập trình viên đã quen với môi trường .Net Framework trên Desktop. Trong một số trường hợp, người dùng có thể tìm thấy sự hỗ trợ trong các thư viện Win32 bằng cách sử dụng khai báo P/Invoke. Còn trong trường hợp xấu nhất, thì bạn phải nghĩ đến cách cài đặt lại một đặc tính thiếu sót nào đó.

.Net Compact Framework không hỗ trợ các module IL (Intermediate Language - Ngôn ngữ trung gian) tiền biên dịch, mà tất cả sự chuyển đổi từ IL sang Native code được thực hiện lúc thực thi chương trình như là JITted code (Just In Timed code – những đoạn code nào cần thiết để chạy ứng dụng mới được biên dịch).

Đối với XML Web Services, .Net Compact Framework chỉ hỗ trợ cơ chế gọi thủ tục từ xa, mà không hỗ trợ cơ chế .Net Remoting, một cơ chế mềm dẻo hơn Web Services bởi vì một đối tượng có thể thực hiện nội bộ trên cùng một máy, hoặc là thực hiện từ xa trên một máy được nối mạng. Các thư viện cơ sở trên Windows CE chủ yếu hỗ trợ tập ký tự Unicode. Bảng sau đây tóm tắt sự hỗ trợ các tập ký tự khác nhau trên các hệ điều hành họ Microsoft.

Hệ điều hành	Hỗ trợ ANSI (các ký tự một/nhiều – Byte)	Hỗ trợ Wide (ký tự Unicode)
Windows 98, Me	Có	Không**
WindowsNT, Windows2000, WindowsXP,	Có	Có
Windows CE.Net	Không	Có

**Với thư viện Microsoft Unicode (MSUL) thì có hỗ trợ sẵn Unicode.

Bảng 3.1: Hỗ trợ các tập ký tự khác nhau trên các hệ điều hành họ Microsoft

Thế giới máy tính đang chuyển dần sang sử dụng ký tự Unicode.

Windows CE chỉ hỗ trợ Unicode, điều này có nghĩa là các hàm Win32 trên Windows CE chỉ chấp nhận các ký tự Unicode. Vấn đề nảy sinh khi đọc các tập tin không phải dạng Unicode, hay truyền thông với các máy khác bằng các ký tự không phải là Unicode (non-Unicode). Chẳng hạn, hầu hết các trang Web đều không gửi các ký tự Unicode. Trong trường hợp này, bạn cần phải chuyển đổi giữa Unicode và các tập ký tự khác (được hỗ trợ bởi lớp *System.Text*).

Hệ thống registry trên Windows CE cũng quan trọng như trên Desktop.Mặc dù được nhấn mạnh là quan trọng nhưng, .Net Compact Framework không hỗ trợ các lớp truy cập registry. Các lớp *Registry* và *RegistryKey* không được hỗ trợ bởi Compact Framework. Tuy nhiên, bạn có thể dùng P/Invoke để truy cập đến registry.

Chương 3 :

Thiết kế các ứng dụng GUI bằng Windows Forms

3.1 Những điều khiển không hỗ trợ

Sau đây là danh sách các điều khiển không được .NET Compact Framework hỗ trợ.

- CheckedListBox
- ColorDialog
- ErrorProvider
- FontDialog
- GroupBox
- HelpProvider
- LinkLabel
- NotificationBubble
- NotifyIcon
- All Print controls
- RichTextBox
- Splitter

3.2 Những hàm .NET Compact Framework không hỗ trợ

Danh sách các hàm .NET Compact Framework không hỗ trợ.

- AcceptButton
- CancelButton
- AutoScroll
- Anchor
- Giao diện đa tài liệu (MDI)
- KeyPreview
- TabIndex

- TabStop
- Kéo thả
- Tất cả các khả năng in ấn
- Các điều khiển Hosting ActiveX

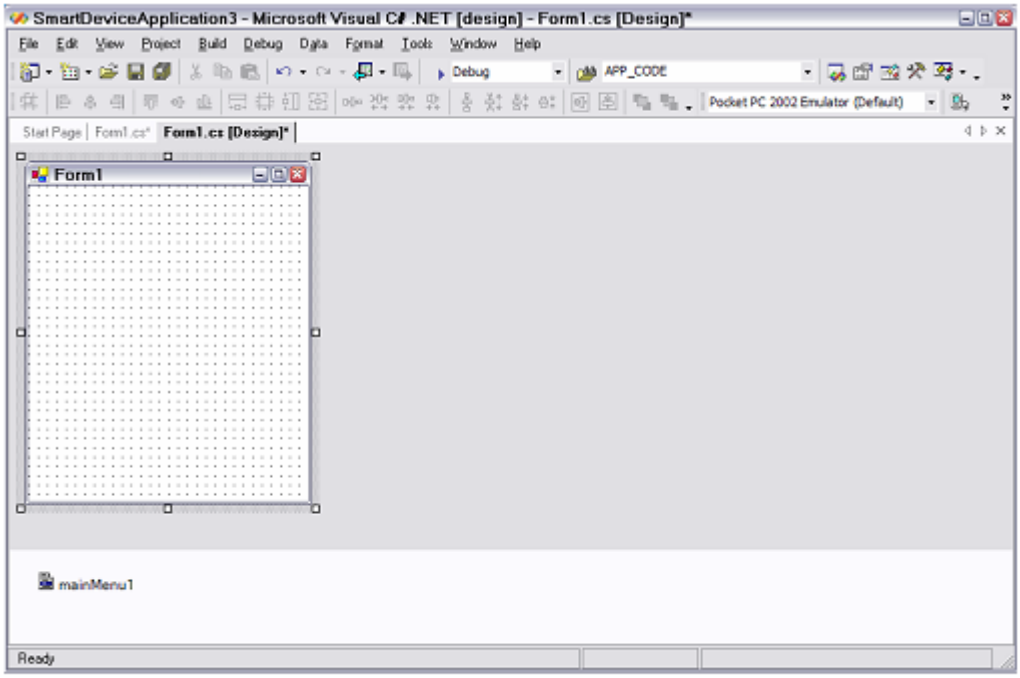
3.3 Thiết kế Form trên Visual Studio .NET

Thiết kế Form bằng Visual Studio .NET cho phép chúng ta thiết kế giao diện ứng dụng trực quan bằng cách kéo thả các điều khiển. Bạn có thể điều chỉnh vị trí các điều khiển, thiết lập các thuộc tính thông qua cửa sổ thuộc tính, và tạo các sự kiện cho các điều khiển.

3.3.1 Cửa sổ thiết kế Forms

Khi chúng ta tạo một dự án Smart Device Extension (SDE), là một ứng dụng cửa sổ, Visual Studio .NET sẽ mở dự án trong phần hiển thị thiết kế. Chúng ta có thể lựa chọn thiết kế từ menu View để đưa vào khung nhìn của dự án. Hình 2.1 đưa đến cho chúng ta Form Designer của dự án SDE Pocket PC trong khung nhìn Designer.

Chú ý rằng thành phần mainMenu1 ở phía dưới của cửa sổ thiết kế. Khu thiết kế danh riêng cho các điều khiển, những điều khiển không có sự thể hiện trực quan, giống như là điều khiển MainMenu, điều khiển ContextMenu, điều khiển Timer, và còn nhiều điều khiển khác.

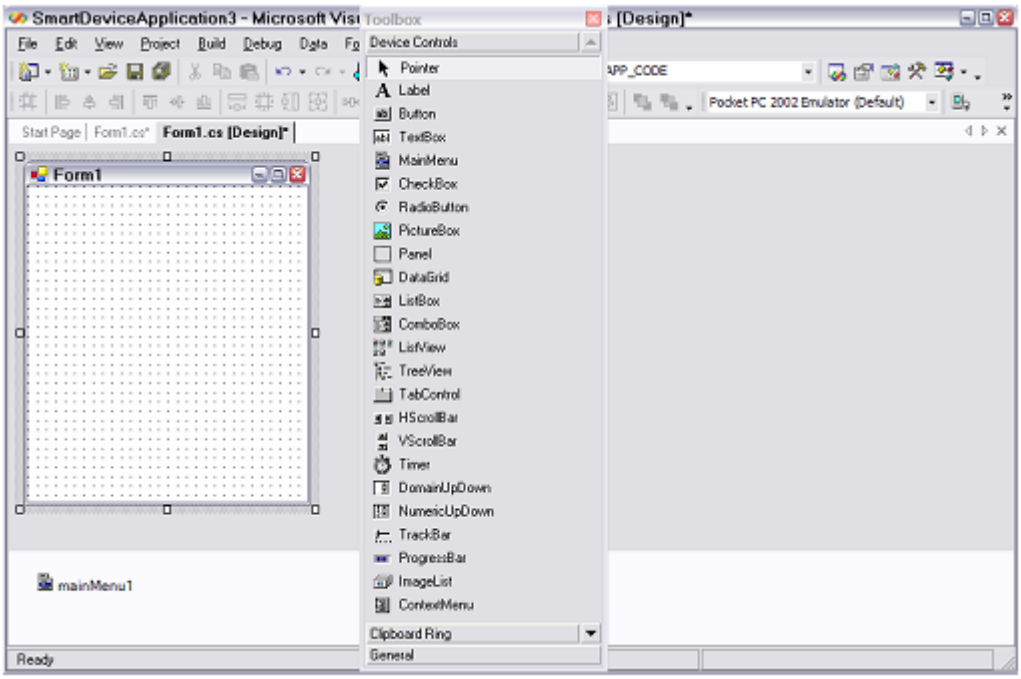


Hình 3.1. SDE Pocket PC trong màn hình Designer view

Khi Form Designer được sử dụng để xây dựng ứng dụng, phương thức InitializeComponent chứa đựng mã nguồn để xây dựng giao diện của ứng dụng. Mã nguồn này có ảnh hưởng lớn đến quá trình thực hiện nếu form của bạn chứa đựng một vài điều khiển ẩn. Trên .NET Compact Framework đề nghị các cửa sổ được tạo theo hướng từ trên xuống. Ví dụ, nếu một panel được đặt trên form và panel đó chứa một vài điều khiển, panel đó sẽ được thêm vào form, và sau đó các điều khiển mới được thêm vào panel.

3.3.2 Cửa sổ ToolBox

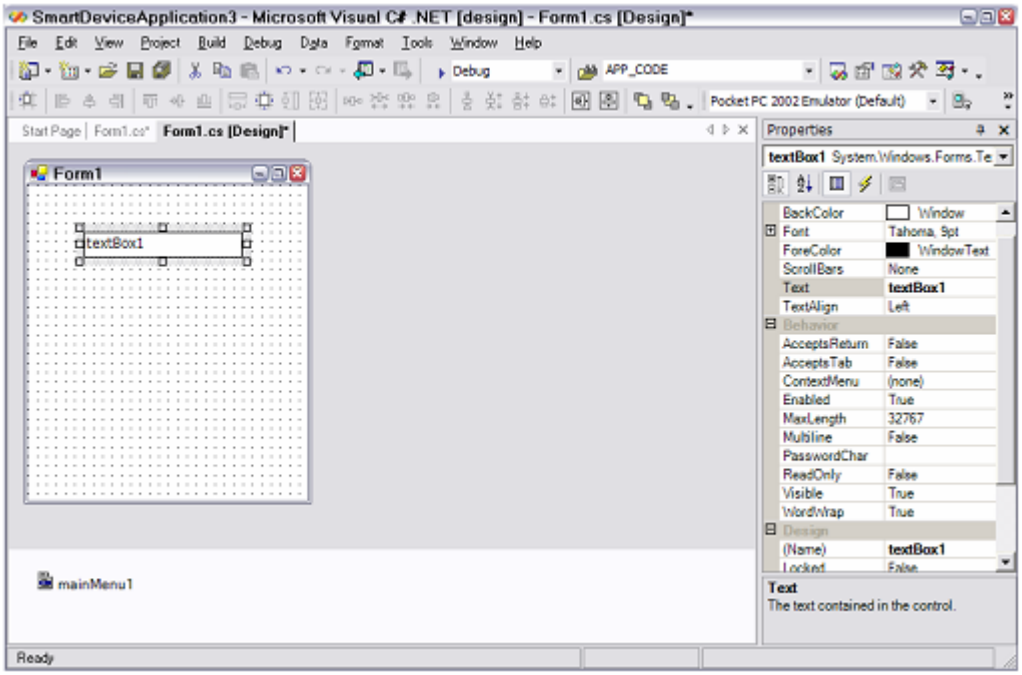
Cửa sổ ToolBox chứa đựng tất cả các điều khiển của .NET Compact Framework mà chúng ta có thể thêm vào ứng dụng. Để thêm một điều khiển vào ứng dụng vào lúc thiết kế rất dễ như là kéo một điều khiển từ ToolBox và thả vào Forms của ứng dụng trong cửa sổ Form Designer. Hình 3.2



Hình 3.2. Cửa sổ ToolBox cho dự án SDE Pocket PC.

3.3.3 Cửa sổ thuộc tính

Cửa sổ thuộc tính chứa đựng tất cả các thuộc tính public của điều khiển đang lựa chọn trong cửa sổ Form Designer. Bạn có thể thay đổi thuộc tính của các điều khiển bằng cách đưa giá trị vào điều khiển TextBox bên cạnh các tên thuộc tính. Nếu thuộc tính có giới hạn số lượng giá trị, sau đó hộp thả xuốngđược hiển thị bên cạnh tên thuộc tính đó. Nếu giá trị của thuộc tính
là một tập hợp các đối tượng hoặc một đối tượng phức tạp, có thể đặc tính đó ở bên cạnh tên thuộc tính. Chọn vào đặc tính đó sẽ hiển thị một hộp thoại cho phép chúng ta sửa đổi giá giá trị của thuộc tính. Hình 3.3 hiển thị cửa sổ thuộc tính khi một điều khiển TextBox được chọn.



Hình 3.3. Cửa sổ Properties của một điều khiển TextBox

3.4 Tìm hiểu các nền tảng Windows Form

Các dự án Smart Device Extensions (SDE) phải nhằm vào hệ điều hành Pocket PC hoặc Windows CE .NET. Hai nền tảng có các hàm giao diện người sử dụng API khác nhau. Một dự án SDE thao tác bằng cách gọi các thư viện khác nhau cho mỗi nền tảng.

3.4.1 Nền tảng Windows CE .NET

Dự án Windows CE .NET giống như các dự án ứng dụng Window .NET Framework đầy đủ. Trước tiên, nút minimize, nút maximize, và nút close xuất hiện trong hộp điều khiển của ứng dụng như chúng ta làm việc trên đối tượng Form .NET Framework đầy đủ. Các nút này có hành vi như là trên desktop. Chúng ta có thể loại bỏ các nút đó bằng cách gán thuộc tính ControlBox của Form là false. Chúng ta cũng có thể loại bỏ nút minimize và nút maximize bằng cách thiết lập các thuộc tính MinimizeBox và MaximizeBox thành false.

Khi một form ứng dụng Windows CE .NET được tạo bằng phần thiết kế Form của Visual Studio.NET, kích cỡ được thiết lập là 640 x 450. Bạn có thể thay đổi thuộc tính Size nếu nó không phù hợp. Mặc dù lớp Form được đưa ra thuộc tính FormBorderSytle, thiết lập thuộc tính Sizable sẽ không ảnh hưởng tới đường viền của cửa sổ. Những ứng dụng Windows CE .NET không thể thay đổi kích cỡ. Nó chỉ có thể thu nhỏ, phóng to hết màn hình, hoặc kích cỡ như thuộc tính Size.

3.4.2 Nền tảng Pocket PC

Các ứng dụng Pocket PC trong tương lai sẽ theo hướng các dự án ứng dụng Windows .NET Framework đầy đủ. Trước tiên, một đối tượng MainMenu luôn luôn được thêm vào một ứng dụng Pocket PC. Chúng ta có thể loại bỏ menu đó, những hành động đó sẽ là nguyên nhân phát sinh ngoại lệ khi tương tác với Soft Input Panel (SIP). SIP là một phần mềm bổ sung của bàn phím QWERTY.

Cửa sổ ToolBox của Visual Studio .NET chứa đựng một điều khiển InputPanel. Trên mỗi Pocket PC điều khiển này cho phép chúng ta tương tác với SIP. InputPanel cho phép chúng ta nâng nên và hạ xuống SIP. InputPanel sẽ gắn vào ứng dụng khi SIP có khả năng.

Trong Form phải có một điều khiển MainMenu hợp lệ cho điều khiển InputPanel được thêm vào trong Form. Nếu không có điều khiển MainMenu trên Form, sau đó một ngoại lệ sẽ được đưa ra vào lúc thực thi khi chúng ta cố gắn hiện InputPanel

3.5 Làm việc với Form

Điều khiển Form là nơi chứa các điều khiển của ứng dụng. Điều khiển Form hiện diện là một cửa sổ chứa các điều khiển của ứng dụng. Lớp Form có nhiều thuộc tính tạo ra hành vi khác nhau phụ thuộc vào nền tảng (target platform).

3.5.1 Ảnh hưởng của thuộc tính FormBorderStyle

Thuộc tính FormBorderSytle xác định kiểu đường viền của Form. Giá trị mặc định là FormBorderStyle.FixedSingle.

Trên Pocket PC, thiết lập thuộc tính FormBorderStyle.None để tạo một form cùng với đường viên và không có tiêu đề. Kiểu Form này có thể thay đổi kích thước và di chuyển trong mã nguồn nhưng không thể thay đổi bởi người sử dụng. Thiết lập thuộc tính FillBorderStyle.FixedSingle hoặc bất kỳ giá trị nào khác sẽ tạo ra một Form bao trùm toàn bộ màn hình, và Form sẽ không thể di chuyển và thay đổi kích thước.

Trên Windows CE .NET, thiết lập thuộc tính FormBorderStyle.FixedDialog hoặc FormBorderStyle.None sẽ tạo ra một form không có đường viền và tiêu đề. Form sẽ di chuyển và thay đổi kích thước chỉ thông qua mã nguồn của chương trình. Thiết lập thuộc tính FormBorderStyle.FixedSingle hoặc bất kỳ giá trị nào khác sẽ tạo Form có một kích cỡ trả về thông qua thuộc tính Size với đường viên và tiêu đề. Form chỉ có thể thay đổi kích thước thông qua thuộc tính Size với đường viên và tiêu đề. Form chỉ có thể thay đổi kích thước và di chuyển thông qua mã nguồn, và người sử dụng có thể di chuyển form.

3.5.2 Sử dụng thuộc tính ControlBox

Thuộc tính ControlBox của Form xác định hộp điều khiển của Forms có được hiển thị hay. Thiết lập thuộc tính ControlBox thành true sẽ hiển thị hộp điều khiển. Thiết lập thuộc tính này thành false sẽ ẩn hộp điều khiển.

3.5.3 Thuộc tính MinimizeBox và MaximizeBox

Trên Pocket PC hộp điều khiển chỉ chứa đựng nhiều nhất một nút, một là nút minimize, nhãn X, hoặc nút close, nhãn OK. Trên Windows CE .NET hộp điều khiển có thể chứa đựng nút minimize, nút maximize, và nút close. Để các nút này hiển thị được điều khiển bằng thuộc tính MinimizeBox và MaximizeBox. (Bảng 3.1 mô tả giá trị vị trí của MinimizeBox và ảnh hưởng của mỗi nền tảng).

Giá trị	Ứng dụng POCKET PC	Ứng dụng WINDOWS CE .NET
True	**X** (nút minimize trên menu bar)	Nút minimize giống như thông thường
False	**OK** (nút close trên menu bar)	Không có nút minimize trên thanh tiêu đề

Bảng 3.1. Giá trị thuộc tính MinimizeBox và ảnh hưởng của nó cho mỗi nền tảng

Giá trị	Ứng dụng POCKET PC	Ứng dụng WINDOWS CE .NET
Normal	Ứng dụng sẽ điền đầy vùng desktop, cái mà toàn bộ vùng màn hình trừ phần menu start và vùng thanh menu chính.	Ứng dụng có kích cỡ như thuộc tính Size
Maximize	Ứng dụng điền đầu màn hình. Nó sẽ ẩn menu start, nhưng menu chính sẽ vẫn hiển thị.	Ứng dụng phủ toàn bộ vùng desktop

Bảng 3.2. Giá trị thuộc tính MaximizeBox và ảnh hưởng của nó

3.5.4 Thuộc tính Size

Thuộc tính Size xác định kích thước của cửa sổ ứng dụng. Phụ thuộc vào giá trị của thuộc tính FormBorderStyle, ứng dụng có thể bỏ qua giá trị thuộc tính Size hoặc thiết lập giá trị kích thước đặc biệt cho ứng dụng.

3.5.5 Thiết lập vị trí của Form bằng thuộc tính Location

Thuộc tính Location xác định góc trên bên trái của Form. Trên Pocket PC thuộc tính Location không có ảnh hưởng trừ khi thuộc tính FormBorderSytle được thiết lập là FormBorderSytle.None. Trên Windows CE vị trí của cửa sổ luôn luôn bằng thuộc tính Location, trừ khi ứng dụng đưa vào trạng thái phóng to hoặc thu nhỏ hết cỡ.

3.6 Điều khiển Button

Lớp System.Windows.Forms.Button được .NET bổ sung một điều khiển button. Khi người sử dụng bấm vào nút lệnh. Chúng ta có thể thao tác sự kiện này bằng sự thực thi System.EventHandler. Đoạn mã sau đây là sự thực thi EventHandler cái đó hiển thị thời gian hiện hành.

```
Private void button_Click(object sender, System.EventArgs e)
{
  MessageBox.Show(DateTime.Now.ToShortTimeString(),
            "The Current Time Is",
            MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation,
            MessageBoxDefaultButton.Button1);
}
```

Hình 3.4. Ứng dụng GiveEmTime thực thi trên Pocket PC 2002 emulator. Nút có nhãn What is the Time đã được bấm, và thời gian hiện hành được hiển thị trong hộp thoại.

Hình 3.4. Ứng dụng GiveEmTime chạy trên Pocket PC 2002 emulator.

Giá trị KeyCode	Nút phần cứng liên quan
Keys.Up	Nút trên được bấm
Keys.Down	Nút dưới được bấm
Keys.Left	Nút bên trái được bấm
Keys.Right	Nút bên phải được bấm
Keys.Return	Nút giữa được bấm

Bảng 3.3. Mã phím được phát sinh bằng Directional Pad trên thiết bị Pocket PC

3.7 Điều khiển TextBox

Điều khiển cho phép người dùng có thể nhập dữ liệu đầu vào cho ứng. Điều khiển TextBox hỗ trợ thuộc tính BackColor và ForeColor, không giống như hầu hết các điều khiển khác trong .NET Compact Framework. Sự kiện Click không hỗ trợ, nhưng có hỗ trợ các sự kiện KeyPress, KeyUp, và KeyDown. Thuộc tính PasswordChar được hỗ trợ.

3.8 Điều khiển Label

Điều khiển nhãn cho phép chúng ta hiển thị văn bản tới người sử dụng. Thuộc tính Text của điều khiển xác định văn bắn sẽ được hiển thị tới người sử dụng. Văn bản hiển thị có thể có sự căn lề khác nhau dựa vào thuộc tính TextAlign. Thuộc tính TextAlign có thể nhận các giá trị là TopLeft, TopCenter, và TopRight.

3.9 Điều khiển RadioButton

Nút điều khiển Radio đưa tới người sử dụng một dãy các giá trị lựa chọn loại trừ nhau. Khi một nút radio trong một nhóm được chọn, các nút khác sẽ tự động bị bỏ chọn. Lớp RadioButton có hai sự kiện được đưa ra khi trang thái chọn của RadioButton thay đổi: Click và CheckedChanged. Sự kiện Click phát sinh khi người sử dụng chọn vào nút radio.

Chúng ta có thể thao tác với sự kiện này như là đối với sự kiện Click của lớp button. Sự kiện CheckedChanged được phát sinh khi trạng thái chọn của RadioButton thay đổi bằng lập trình hay giao diện đồ hoạ.

Sự kiện Click sẽ không phát sinh nếu thuộc tính Checked của radioButton được thay đổi bằng lập trình. Ứng dụng demo Arnie.exe, làm thế nào để sử dụng một nhóm các điều khiển.

Hình 3.5 cho thấy ứng dụng chạy trên Pocket PC emulator.

Hình 3.5. Ứng dụng Arnie chạy trên Pocket PC 2002 emulator

Sau đây là đoạn mã demo thao tác với sự kiện CheckedChanged.

```
private void radioButton2_CheckedChanged(object sender,
            System.EventArgs e)
{
      if(this.radioButton2.Checked)
        MessageBox.Show
          ("Wrong, The Terminator (1984) O.J Simpson almost got the role...",
               "Wrong!");
    }
```

3.10 Điều khiển CheckBox

Điều khiển CheckBox giống như điều khiển RadioButton. Điều khiển này đưa đến cho người sử dụng danh sách các lựa chọn. Điều khác là điều khiển CheckBox có thể có nhiều lựa chọn trong cùng một lúc, trong khi điều khiển RadioButton lựa chọn loại trừ.

Điều khiển CheckBox cung cấp thuộc tính CheckState, xác đinh điều khiển nào được chọn. Thuộc tính CheckState thực chất là một bảng liệt kê. Thành phần của nó là Unchecked, Checked, và Indeterminate. Trạng thái Indeterminate chỉ có thể dược sử dụng khi thuộc tính ThreeState của điều khiển CheckBox được thiết lập là true. Khi CheckState là Indeterminate và thuộc tính ThreeState là true, điều khiển được khoanh thành ô vuông. Có nghĩa là trạng thái chọn không thể kiểm soát. Điều khiển sẽ không trả kết quả tới người sử dụng khi chọn trong suốt quá trình thuộc tính AutoCheck được thiết lập là false. Khi thuộc tính AutoCheck được thiết lập true, khi đó có thể bấm chọn trên điều khiển.

Ứng dụng Apples.exe là một ví dụ khác đơn giản là xác định loại táo người sử dụng thích. Điều khiển CheckBox trên cùng có nhãn là “I like apples.”. Các điều khiển CheckBox khác có nhãn cùng với loại táo khác nhau và một trạng thái mờ mờ cho đến khi CheckBox có nhãn “I like apples” được chọn, khi đó người sử dụng lựa chọn loại táo anh ta hoặc cô ta thích. Hình 3.6 cho chúng ta thấy ứng dụng chạy trên Pocket PC emulator.

Hình 3.6. Các trạng thái của điều khiển CheckBox chạy trên Pocket PC 2002.

3.11 Điều khiển ComboBox

Điều khiển ComboBox là điều khiển thể hiện một danh sách các lựa chọn trong sự hạn chế của màn hình. ComboBox xuất hiện như là điều khiển TextBox cùng với một mũi tên bên tay phải. Một danh sách lựa chọn thả xuống dưới điều khiển khi người sử dụng chọn vào mũi tên. Khi người sử dụng lựa chọn một tùy chọn hoặc chọn lại mũi tên, danh sách các tuỳ chọn sẽ cuộn lên.

Để thêm một mục vào điều khiển ComboBox có thể hoàn thành lúc thiết kế và lúc thực thi. Để thêm một mục vào ComboBox lúc thiết kế, đơn giản là chọn ComboBox trong Form Designer. Sau đó chọn vào phần bên phải tên thuộc tính Items trong cửa sổ thuộc tính. Nó sẽ đưa đến một hộp thoại String Collection Editor (sem hình 2.7). Trong hộp thoại String Collection Editor, đưa vào danh sách các mục sẽ xuất hiện trong ComboBox. Mỗi mục phải xuất hiện trên cùng một dòng.

Hình 3.7. Hộp thoại String Collection Editor.

Các mục có thể được thêm vào điều khiển ComboBox lúc thực thi. Điều này có thể hoàn thành bằng hai cách:

Cách 1: Gọi phương thức Add trên thuộc tính tập hợp Items của điều khiển ComboBox.

Các mục có thể loại bỏ thông qua phương thứ Remove trên tập hợp Items, hoặc tất cả các mục có thể loại bỏ bằng cách gọi phương thức Clear. Đoạn mã sau thêm ba chuỗi vào điều khiển ComboBox có tên comboBox1

```
comboBox1.Items.Add("Hi");
comboBox1.Items.Add("Howdy");
comboBox1.Items.Add("Wuz Up");
```

Cách 2: Chúng ta có thể thêm vào ComboBox lúc thực thi bằng cách ràng buộc điều khiển với một đối tượng tập hợp. Điều này được hoàn thành bằng cách thiết lập DataSource với một đối tượng tập hợp. Khi ComboBox cố gắng thêm một mục vào danh sách, nó sẽ gọi phương thức ToString trên mỗi mục trong DataSource và thêm vào danh sách lựa chọn. Chuỗi có thể tuỳ biến bằng cách

thiết lập thuộc tính DisplayName của điều khiển ComboBox. ComboBox sẽ gọi thuộc tính riêng biệt trong thuộc tính DisplayName và thêm chuỗi trả về vào danh sách lựa chọn.

Đoạn mã Listing 2.1 mô tả cách ràng buộc một ComboBox với một danh sách đối tượng tuỳ biến. Lớp Customer là một lớp tuỳ biến lưu trữ tên của khách hàng. Lớp có một thuộc tính FullName, thuộc tính này lưu trữ tên đầy đủ. Khi ComboBox được giới hạn trong phương thức LoadCustomer, thuộc tính FullName được thiết lập như là DisplayName.

Listing 2.1

```
class Customer {
  string m_First;
  string m_Middle;
  string m_Last;
public Customer(string first, string middle, string last) {
  m_First = (first == null) ? string.Empty : first;
  m_Middle = (middle == null) ? string.Empty : middle;
  m_Last = (last == null) ? string.Empty : last;
}
public string FirstName {
  get { return m_First; }
}
public string MiddleName {
  get { return m_Middle; }
}
public string LastName {
  get { return m_Last; }
}
static string FullNameWithInitial = "{0} {1}. {2}";
```

```
static string FullNameNoInitial = "{0} {1}";
public string FullName {
  get {
    return (m_Middle.Length > 0) ?
string.Format(FullNameWithInitial, m_First, m_Middle[0], m_Last) :
string.Format(FullNameNoInitial, m_First, m_Last);
        }
    }
}
private void LoadCustomers() {
  if(customers != null)
    return;
customers = new Customer[6];
customers[0] = new Customer("Ronnie", "Donnell", "Yates");
customers[1] = new Customer("Moya", "Alicia", "Hines");
customers[2] = new Customer("Veronica", "Christine", "Yates");
customers[3] = new Customer("Diane", "", "Taylor");
customers[4] = new Customer("Kindell", "Elisha", "Yates");
customers[5] = new Customer("Zion", "Donnell", "Yates");
this.comboBox1.DataSource = customers;
}
```

Có hai cách để lấy mục đang được chọn trong điều khiển ComboBox. Thứ nhất, thuộc tính SelectedIndex trả về chỉ số của mục đang chọn. Chỉ số này có thể được sử dụng để truy cập mục đang chọn từ thuộc tính Items của điều khiển ComboBox. Đoạn mã sau minh hoạ thuộc tính SelectIndex:

```
string selItem = comboBox1.Items[comboBox1.SelectedIndex].ToString();
```

Điều khiển ComboBox cung cấp thuộc tính SelectedItem, thuộc tính này

trả về một tham chiếu đến mục đang chọn. Một là chúng ta có thể tham chiếu đến mục đang chọn, chúng ta không cần phải đưa chỉ số vào thuộc tính Items . Đoạn mã sau mô tả cách sử dụng thuộc tính SelectedItem:

```
string selItem = comboBox1.SelectedItem.ToString();
```

3.12 Điều khiển ListBox

ListBox sẽ được sử dụng nếu chúng ta có đủ không gian màn hình để hiển thị một vài tuỳ chọn cho người sử dụng trong một lần. ComboBox và ListBox có các thuộc tính và các phương thức giống nhau. Bao gồm thuộc tính tập hợp Items và các thương thức Add, Remove, và Clear trên thuộc tính Items . Ví dụ, đoạn mã sau thêm chuỗi vào điều khiển ListBox lúc thiết kế.

```
listBox1.Items.Add("Hi");
listBox1.Items.Add("Howdy");
listBox1.Items.Add("Wuz Up");
```

Chúng ta có thể thêm vào điều khiển ListBox lúc thực thi bằng cách gắn ListBox với một tập hợp. Trong quá trình gắn một điều khiển ListBox giống với quá trình trong điều khiển ComboBox. Trước tiên, thiết lập DataSource với một tập hợp. Sau đó, thiết lập thuộc tính DisplayMember với một mục trong nguồn dữ liệu, mục này sẽ được hiển thị như là một chuỗi.

```
private void LoadCustomers() {
DataGrid
if(customers != null)
  return;
customers = new Customer[6];
customers[0] = new Customer("Ronnie", "Donnell", "Yates");
customers[1] = new Customer("Moya", "Alicia", "Hines");
customers[2] = new Customer("Veronica", "Christine", "Yates");
customers[3] = new Customer("Diane", "", "Taylor");
customers[4] = new Customer("Kindell", "Elisha", "Yates");
```

```
customers[5] = new Customer("Zion", "Donnell", "Yates");
this.listBox1.DataSource = customers;
this.listBox1.DisplayMember = "FullName";
}
```

ListBox có hai thuộc tính SelectedIndex và SelectedItem cho phép truy cập mục đang chọn.

3.13 Các điều khiển khác

- NumericUpDown
- DomainUpDown
- ProgressBar
- StatusBar
- TrackBar
- ToolBar
- MainMenu
- ContextMenu
- Timer
- OpenFileDialog và SaveFileDialog
- Panel
- HScrollBar và VScrollBar
- ImageList
- PictureBox
- ListView
- TabControl
- TreeView
- DataGrid

Chương 4 :

Ứng dụng từ điển trên Pocket PC

4.1 Vai trò của từ điển

Ngày nay, khái niệm Từ điển đã trở nên quá quen thuộc với chúng ta. Nó là một công cụ tra cứu rất hữu ích phục cho nhu cầu học tập, nghiên cứu, cũng như giao tiếp hằng ngày.

Từ khi có sự xuất hiện của chiếc máy tính cá nhân (PC – Personal Computer), thì Từ điển lại được nâng lên một tầm cao mới. Hàng loạt các ứng dụng Từ điển được ra đời cung cấp cho người sử dụng không chỉ những chức năng tra cứu mà còn những âm thanh, hình ảnh minh họa kèm theo. Có thể nói các ứng dụng Từ điển đã giải phóng con người khỏi phải lật từng trang giấy để tra từ. Người ta chỉ đơn giản nhập vào từ muốn biết nghĩa. Công việc tìm kiếm còn lại thuộc về máy tính.

Trong thời đại mở cửa, hội nhập quốc tế hiện nay, có thể nói Từ điển đã trở thành một cẩm nang không thể thiếu.

4.2 Đặc trưng ứng dụng của một từ điển

Để xây dựng một ứng dụng Từ điển hiệu quả, ta cần quan tâm 2 đặc tính quan trọng sau đây:

- Tốc độ xử lý nhanh. Có thể nói thao tác cơ bản nhất của một ứng dụng Từ điển là tra cứu. Do đó, việc tìm kiếm phục vụ cho thao tác tra cứu xảy ra hết sức thường xuyên, yêu cầu nhanh chóng hiện kết quả cho người sử dụng.
- Dữ liệu lưu trữ lớn. Tùy theo các loại Từ điển khác nhau mà có kích thước lưu trữ khác nhau. Nhưng nhìn chung, thường thì dữ liệu lưu trữ của một Từ điển là khá lớn. Bên cạnh đó, ta cũng cần quan tâm đến số lượng các phần tử trong một Từ điển.

Ngoài ra yêu cầu dễ sử dụng cũng là một đặc tính quan trọng không chỉ riêng với ứng dụng Từ điển.

Với những tính chất cơ bản trên, thì việc khảo sát các đặc trưng của Pocket PC, đồng thời phân tích các vấn đề phát sinh khi xây dựng một ứng dụng Từ điển trên Pocket PC là rất cần thiết để từ đó đưa ra những giải pháp phù hợp.

4.3 Giới hạn về bộ xử lý

Đặc điểm chung của các bộ xử lý trên thiết bị PDA là tiết kiệm pin, nhỏ gọn, nhất là không đòi hỏi các thiết bị giải nhiệt chuyên dụng. Ngoài ra, vì lý do tối ưu giữa chi phí sản xuất và nhu cầu sử dụng chủ yếu của các thiết bị PDA, bộ xử lý cho PDA thường có tốc độ thấp và không ứng dụng hoặc ứng dụng hạn chế một số công nghệ tăng tốc xử lý dùng cho các bộ xử lý Pentium.Pocket PC sử dụng một loại CPU khác với CPU thông thường.

Máy Pocket PC 2002 sử dụng CPU StrongARM hoặc XScale của Intel. Các CPU này được gọi là các bộ xử lý RISC (Reduced Instruction Set Computer) vì nó chỉ thực hiện một lệnh trong một chu kì CPU. (Mỗi chu kì CPU trong một giây gọi là 1 Hertz hay 1 Hz). Máy PC thông thường của chúng ta sử dụng bộ xử lý CISC (Complex Instruction Set Computer), đòi hỏi nhiều chu kì CPU để xử lý một lệnh xác định. Do đó, các bộ xử lý RISC như StrongARM theo lý thuyết, có thể tính toán lên tới 206 triệu lệnh một giây nếu nó xử lý ở tốc độ 206 Mhz. Các bộ xử lý StrongARM và XScale cũng xử lý các lệnh 32bit, giống như PC thông thường. Tuy nhiên, lại có một khác biệt lớn trong việc xử lý kích thước lệnh. Bộ xử lý CISC cho phép lệnh có thể có chiều dài biến đổi, vì vậy, nó phải tính toán kích thước của dữ liệu cần đọc khi xử lý lệnh. Trong kiến trúc RISC, mỗi lệnh 32bit sẽ có 32 dữ liệu đi kèm. Vì vậy, bộ xử lý luôn biết được phải đọc bao nhiêu dữ liệu., đây là một điểm mạnh của kiến trúc này.

Độ rộng của bus cùng với tốc độ bus cũng có một ảnh hướng lớn đến tốc độ xử lý và tính toán của máy tính. Độ rộng của bus chỉ ra số lượng bit (hay byte) dữ liệu có thể đọc từ / ghi lên RAM vào bộ vi xử lý. Còn tốc độ bus chỉ ra dữ liệu có thể được đọc từ RAM vào bộ vi xử lý với độ nhanh như thế nào.

Ta có thể hình dung, độ rộng bus như số làn giao thông trên xa lộ, còn tốc độ bus như giới hạn về tốc độ khi lưu thông. Hiện nay, StrongARM và XScale sử dụng bus tốc độ 100Mhz và thiết kế hiện nay của Pocket PC cho phép dùng bus 16bit. Điều này có nghĩa là tốc độ bus tối đa là 200MB/giây.

Vấn đề đặt ra ở đây là như vậy, bộ vi xử lý có thể đạt được tốc độ xử lý bao nhiêu. Do chúng ta cần đọc 8bytes dữ liệu trước khi xử lý một lệnh, nên giả sử chúng ta có được tốc độ bus tối đa là 200MB/giây, thì thực sử, bộ xử lý chỉ thi hành được 25 triệu lệnh.

Ngoài ra, lại có một vấn đề khác, đó là cache. Cache là một loại RAM đặc biệt được chứa bên trong CPU và xử lý với cùng tốc độ của CPU. Cache trong bộ xử lý StrongARM là 16K cho mã chương trình và 8K cho dữ liệu, cache trong XScale là 32K cho mã chương trình và 32K cho dữ liệu. Nếu dữ liệu và mã chương trình cần xử lý được chứa trọn trong cache, hệ thống sẽ có thể thi hành với tốc độ thật của CPU. Như vậy, tốc độ của ứng dụng có thể được thi hành sẽ có thể tăng từ 25 lên 206 triệu lệnh một giây, tuỳ theo chương trình và dữ liệu có vừa với kích thước cache không. Tốc độ của hệ thống cũng phụ thuộc vào nguồn cung cấp năng lượng, do đó, khi sử dụng Pocket PC, ta có thể nhận thấy sự khác biệt về tốc độ khi thực thi chương trình.

Bây giờ, sử dụng các để so sánh về hiệu suất hoạt động giữa máy Desktop PC và Pocket PC. Xét các hệ thống Desktop PC sử dụng bus 100-133 Mhz. Pentium IV có thể sử dụng DDR RAM hoặc RAM BUS (RD RAM) có tốc độ từ 200-800Mhz.

Vì vậy, chỉ xét trên phương diện RAM, nếu sử dụng RAM 133Mhz, hiệu suất Desktop PC đã tăng 1/3 lần (25 lên 33 triệu lệnh một giây). Tốc độ CPU của Desktop cũng lớn hơn rất nhiều khi hiện tại đã đạt tới tốc độ trên 3Ghz dẫn đến hiệu suất toàn hệ thống cao hơn.

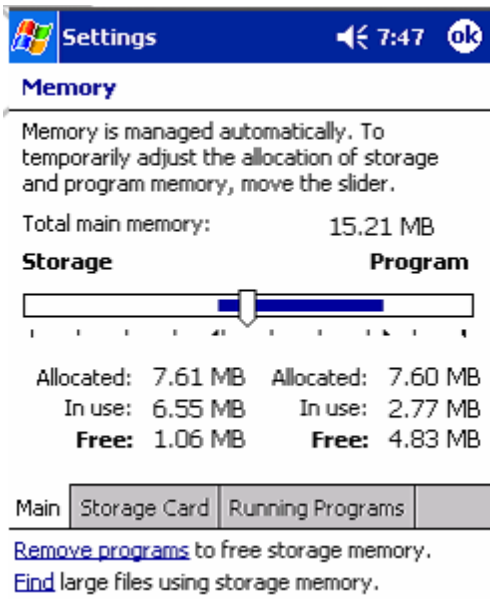
Khi xây dựng ứng dụng Từ điển trên Pocket PC, đặc điểm về bộ xử lý của Pocket PC ảnh hưởng rất lớn đến tốc độ xử lý, hiển thị các từ trong Từ

điển cũng như nghĩa của chúng. Để giải quyết vấn đề này, ta cần có những giải pháp phù hợp để tăng hiệu năng của ứng dụng, giúp người sử dụng có thể sử dụng được chương trình với tốc độ chấp nhận được.

4.4 Giới hạn về bộ nhớ và khả năng lưu trữ

Các Pocket PC thông thường có ROM ít nhất là 8MB, RAM ít nhất là 8MB. Các Pocket PC chuyên dụng có ROM ít nhất là 12MB, RAM ít nhất là 16MB. Tuy nhiên, yêu cầu về dung lượng của RAM và ROM tuỳ thuộc vào loại CPU sử dụng, những thành phần mà nhà sản xuất hỗ trợ trên Pocket PC và tuỳ thuộc vào loại Pocket PC.

Do hạn chế về nguồn năng lượng pin và kích thước của thiết bị, Pocket PC không sử dụng các thiết bị lưu trữ như đĩa cứng hay đĩa mềm mà nó sử dụng một cơ chế gọi là Bộ lưu trữ đối tượng (Object store). Object store là một vùng RAM được người dùng định nghĩa, dùng để lưu trữ các tập tin, các thông tin registry và các database (gọi là Object Store dạng Storage). Vùng RAM còn lại dùng làm bộ nhớ cho các chương trình được thực thi (gọi là Object Store dạng Program). Object store vẫn lưu giữ được thông tin của các ứng dụng và dữ liệu ngay cả khi năng lượng cung cấp chính bị mất nhờ nguồn pin dự trữ. Có thể xem object store như là đĩa cứng trên thiết bị dùng Windows CE. Vì các end-user không biết về object store nên Microsoft đã cung cấp trình WinCE Explorer để đọc nội dung của nó (tương tự như Microsoft Windows Explorer để đọc nội dung của đĩa cứng trên máy desktop PC).



Hình 4.1: Tình trạng bộ nhớ trên Pocket PC

Để cung cấp thêm khả năng lưu trữ cho Pocket PC (cũng nhu các thiết bị PDA khác), hấu hết các máy đều cho phép gắn thêm các thẻ nhớ (Flash Memory Card) đóng vai trò một bộ nhớ ngoài. Đặc điểm chung của các thẻ nhớ là có tốc độ chậm hơn nhiều so với bộ nhớ RAM nhưng giá thành rẻ và cơ động, có thể dùng chung cho các thiết bị điện tử khác. Ta có thể tận dụng khả năng lưu trữ của thẻ nhớ để giảm bớt hạn chế về khả năng lưu trữ của Pocket PC.

Hình 4.2: Một số thẻ nhớ cho PDA

Những hạn chế về bộ nhớ trên Pocket PC ảnh hưởng rất nhiều đến việc tổ chức dữ liệu Từ điển của ứng dụng. Giải pháp gắn thêm các thẻ nhớ để tăng khả năng lưu trữ cho Pocket PC là không khả thi do các thẻ nhớ có tốc độ chậm trong khi ứng dụng Từ điển lại có tần suất truy cập dữ liệu rất lớn, đòi hỏi tốc độ nhanh. Một giải pháp khác được đưa ra là *nén* dữ liệu Từ điển. Phương pháp này có vẻ khả thi hơn. Tuy nhiên, cái giá phải trả là phải truy cập dữ liệu nén. Điều này cũng ảnh hưởng nhiều đến tốc độ của ứng dụng. Vì vậy, việc dung hòa giữa tốc độ xử lý và dữ liệu lưu trữ của ứng dụng là một trong những vấn đề quan trọng nhất khi xây dựng ứng dụng Từ điển trên Pocket PC.

Ngoài ra, ta còn phải xem xét đến hình thức lưu trữ dữ liệu Từ điển. Trên Pocket PC, có 2 hình thức lưu trữ dữ liệu chính, đó là lưu trữ dưới dạng *Tập tin* và dưới dạng *Cơ sở dữ liệu*. Cũng như trên desktop, cơ sở dữ liệu thường được dùng để lưu các cấu trúc dữ liệu biến động, có nhu cầu truy xuất phức tạp,…trong khi dữ liệu Từ điển lại có cấu trúc dữ tĩnh, truy xuất tương đối đơn giản Hơn nữa, cơ sở dữ liệu trên Pocket PC (SQLCE) chưa được sử dụng rộng rãi. Hiện nay, có rất ít ứng dụng sử dụng cơ sở dữ liệu trên Pocket PC.

Do đó việc chọn tập tin là hình thức lưu trữ dữ liệu của ứng dụng Từ điển là hợp lý.

4.5 Hạn chế về khả năng tương tác giữa người dùng và thiết bị

Hạn chế về khả năng tương tác người dùng tuy không làm cản trở đến hiệu quả thi hành chương trình nhưng lại gây khó khăn cho công việc thiết kế giao diện chương trình. Việc xác định rõ các hạn chế về khả năng tương tác người dùng giúp chúng ta xây dựng các ứng dụng trên Pocket PC thân thiện, tiện dụng với người dùng hơn.

4.5.1 Màn hình hiển thị nhỏ

Do đặc tính nhỏ gọn, nên nên màn hình hiển thị của Pocket PC thường có kích thước nhỏ. Với màn hình quá nhỏ, sẽ rất khó để vừa hiển thị danh sách các từ và nghĩa của nó trong Từ điển, vừa bố trí tất cả các chức năng trên các thanh công cụ, thanh thực đơn. Giải pháp thường thấy trong các ứng dụng trên Pocket PC là sử dụng tối đa các *popup menu*, *context menu*, *dropdown toolbar button* để người dùng chọn lựa chức năng. Chỉ có các chức năng thường xuyên sử dụng mới được thiết kế nằm sẵn trên màn hình như chức năng tra nghĩa của từ, chức năng phát âm, …

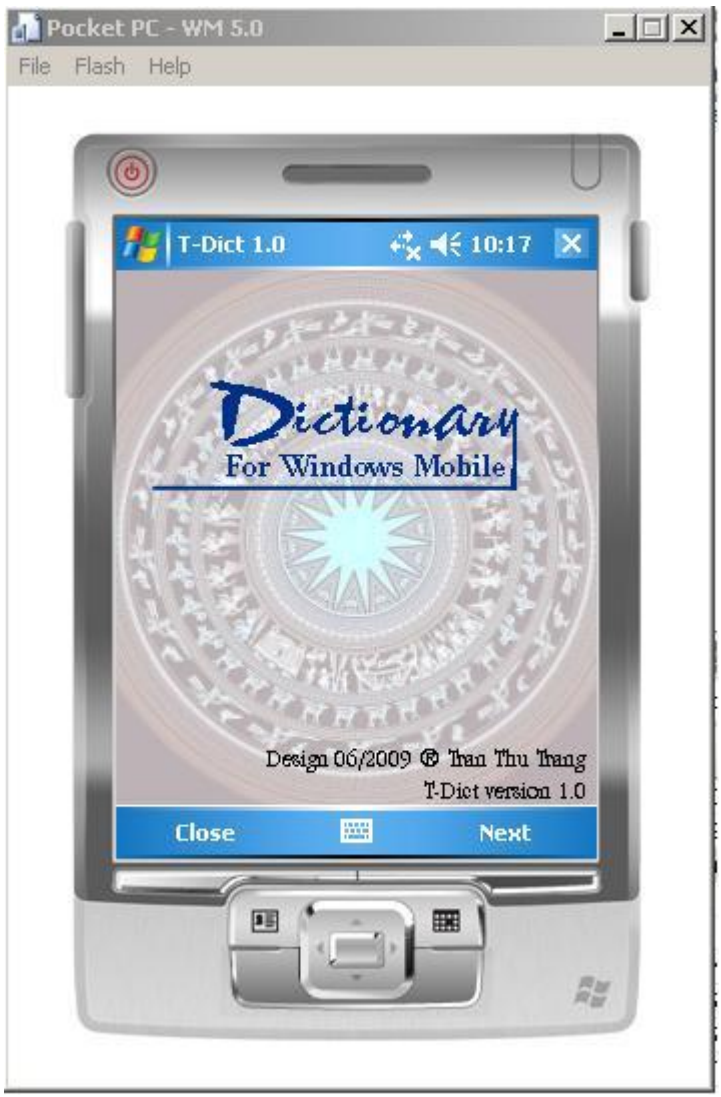
4.5.2 Bàn phím

Pocket PC không có bàn phím vật lý mà sử dụng một bàn phím ảo. Khi được kích hoạt, bàn phím ảo sẽ chiếm một phần của màn hình (khoảng $^1/_3$) và có thể sẽ che đi các nút chức năng, ô nhập liệu quan trọng.

Các ứng dụng cho Pocket PC khi thiết kế phải lưu ý đến trường hợp này để có thiết kế giao diện phù hợp. Microsoft khuyến cáo các lập trình viên thiết kế các ô nhập liệu (cần đến bàn phím) ở phía trên, tránh tình trạng khi bật bàn phím ảo để nhập liệu lại không thể nhìn thấy ô này.

4.6 Chương trình mô phỏng

4.6.1 Giao diện chính của chương trình



4.6.2 Giao diện tra từ

4.6.3 Kết quả tra từ

Tài liệu tham khảo

[1] T.S. Dương Anh Đức - Th.S. Trần Hạnh Nhi, *"Nhập môn Cấu trúc dữ liệu và thuật toán"*

[2] Lê Thụy Anh, "*Nhúng ứng dụng GIS vào thiết bị PDA*", luận án Thạc sĩ Tin học, 2003

[3] Võ Sỹ Nam - Đỗ Lệnh Hùng Sơn, *"Xây dựng một ứng dụng bản đồ trên máy pocket pc 2002 (Windows CE 3.0) cho phép hiển thị một bản đồ điện tử và cung cấp một số chức năng tìm kiếm thông tin"*, Luận văn cử nhân tin học, Đại học Khoa học Tự nhiên Tp.Hồ Chí Minh, 2003

[4] Paul Yao - David Durant, *"Programming the .NET Compact Framework in C#"*

[5] Microsoft, *Microsoft Developer Network*, 10-2003

[6] Jason P.Nottingham - Steven Makofsky - Andrew Tucker, "*Teach Yourself Windows CE Programming in 24 hours*", SAMS - 2001

[7] *"GZIP file format specification version 4.3"* (RFC1952)

[8] *"DEFLATE Compressed Data Format Specification"* (RFC1951)

[9] *Dictzip – Linux Man Page*

Website:

[10] http://www.paulyao.com/cfbook

[11] http://www.pocketpcdn.com

[12] http://www.opennetcf.org

[13] http://www.cegadgets.com/wincedevfaq

[14] http://codeproject.com/netcf