**MỤC LỤC**

[**MỞ ĐẦU** 3](#_Toc294746597)

[**I.** **Cơ sở lí thuyết Remote Execution and Distribution Computing** 4](#_Toc294746598)

[**II.** **Khảo sát thực tế với nền tảng WCF trên Web Service** 4](#_Toc294746599)

[***1.*** ***Khái niệm cơ bản*** 4](#_Toc294746600)

[*1.1.* *WCF là gì?* 4](#_Toc294746601)

[*1.2.* *Tại sao lại sử dụng WCF?* 4](#_Toc294746602)

[***2.*** ***Kiến trúc tổ chức*** 5](#_Toc294746603)

[*2.1.* *Các hiệp nghị (Contracts)* 6](#_Toc294746604)

[*2.2.* *Dịch vụ thực thi (Runtime Service)* 6](#_Toc294746605)

[*2.3.* *Thông điệp/Bản tin (Message)* 7](#_Toc294746606)

[*2.4.* *Lưu trữ và kích hoạt (Host and Activation)* 8](#_Toc294746607)

[***3.*** ***Các tính năng đặc trưng*** 8](#_Toc294746608)

[*3.1.* *Giao dịch (Transaction)* 8](#_Toc294746609)

[*3.2.* *Lưu trữ (Host)* 8](#_Toc294746610)

[*3.3.* *Bảo mật (Security)* 9](#_Toc294746611)

[***4.*** ***Dịch vụ trong WCF (Services)*** 9](#_Toc294746612)

[*4.1.* *Định kiểu (Typed Service)* 9](#_Toc294746613)

[*4.2.* *Không định kiểu (Untyped Service)* 10](#_Toc294746614)

[*4.3.* *Bản tin định kiểu (Typed Message Service)* 10](#_Toc294746615)

[***5.*** ***Kiến trúc Client trong WCF*** 10](#_Toc294746616)

[*5.1.* *Khái niệm cơ bản* 10](#_Toc294746617)

[*5.2.* *Các đối tượng phía Client* 12](#_Toc294746618)

[*5.3.* *Các phương thức liên lạc của Client* 14](#_Toc294746619)

[***6.*** ***Mô hình lập trình WCF*** 16](#_Toc294746620)

[*6.1.* *Lựa chọn phương pháp tiếp cận* 16](#_Toc294746621)

[*6.2.* *Mô hình dịch vụ* 17](#_Toc294746622)

[*6.3.* *Các phương pháp lập trình với WCF* 18](#_Toc294746623)

[***7.*** ***Bảo mật trong WCF (Security)*** 19](#_Toc294746624)

[*7.1.* *Ý nghĩa bảo mật trong WCF* 19](#_Toc294746625)

[*7.2.* *Các lĩnh vực bảo mật của WCF* 21](#_Toc294746626)

[**III.** **Ứng dụng kiểm thử** 23](#_Toc294746627)

**MỞ ĐẦU**

Ngày nay, mạng Internet phát triển ngày càng mạnh mẽ, với một tốc độ nhanh chóng và phạm vi rộng lớn. Cơ sở hạ tầng mạng được cải tiến, nâng cấp cả về số lượng và chất lượng. Đồng thời, các ứng dụng hoạt động trên mạng cũng được những nhà phát triển tập trung cải thiện, mở rộng chức năng, nâng cao hiệu suất và đặc biệt là tiêu tốn chi phí tài nguyên tối thiểu.

Việc phát triển qui mô mạng Internet nói chung cũng kéo theo một mạng lưới rất rộng lớn các thiết bị, thành phần liên kết với nhau. Tương ứng, đó là những ứng dụng trao đổi thông tin liên tục với lượng dữ liệu rất lớn. Tuy nhiên, khi lượng dữ liệu được trao đổi càng cao thì việc đảm bảo đường truyền lại là một bài toán hết sức nan giải. Đối với các nhà phát triển, việc định nghĩa ra một chuẩn giao tiếp giữa các ứng dụng trên mạng hết sức cần thiết. Điều đó giúp tạo ra một qui trình chuẩn để xây dựng các ứng dụng mạng, giảm bớt rắc rối, rườm ra, tiết kiệm chi phí trong việc tổ chức cơ sở hạ tầng cũng như tăng cường tính tương tác giữa nhiều hệ thống khác nhau. Sau khi tìm hiểu về vấn đề này, nhóm chúng em quyết định chọn đề tài: Các giao thức điều khiển, giao tiếp và tính toán trên mạng phân tán – **remote execution and distribution computing**. Toàn bộ nội dung của bài tiểu luận sẽ tập trung trình bày các phương thức trao đổi, giao tiếp giữa các ứng dụng mạng thông qua các giao diện chuẩn được xây dựng. Ngoài ra, nhóm cũng đề xuất một công nghệ mới nhằm hiện thực hóa ý tưởng xây dựng chuẩn giao tiếp trên mạng, đó là nền tảng WCF (Windows Communication Foundation) hoạt động dựa trên Web Service. Đây là mảng kiến thức hết sức thú vị và có ý nghĩa thực tế đối với sinh viên chuyên ngành truyền thông và mạng máy tính.

Trong suốt quá trình thực hiện bài tiểu luận, nhóm đã hết sức nỗ lực để tìm hiểu, trau dồi các kiến thức chuyên môn cần thiết. Tuy nhiên, báo cáo không thể tránh khỏi những thiếu xót và cần tiếp tục cải thiện. Nhóm xin chân thành gửi lời cảm ơn TS. Phạm Huy Hoàng đã tận tình hướng dẫn, giúp đỡ nhóm hoàn thành tốt bài tiểu luận này.

*Nhóm sinh viên*

1. **Cơ sở lí thuyết Remote Execution and Distribution Computing**
2. **Khảo sát thực tế với nền tảng WCF trên Web Service**
3. ***Khái niệm cơ bản***
   1. *WCF là gì?*

WCF (Windows Communication Foundation) là công nghệ nền tảng nhằm thống nhất nhiều mô hình lập trình giao tiếp được hỗ trợ trong .NET 2.0 thành một mô hình duy nhất. Vào tháng 11 năm 2005, .NET 2.0 được Microsoft phát hành trong đó có cung cấp các hàm API riêng biệt cho các liên lạc dựa trên SOAP để tối đa hoá sự làm việc giữa các nền tảng sử dụng Web Services, đồng thời .NET 2.0 còn cung cấp các API để tối ưu việc liên lạc dựa trên mã nhị phân giữa các ứng dụng chạy trên hệ thống Windows gọi là .NET Remoting, các API cho các giao dịch phân tán, và API cho liên lạc dị bộ. WCF thống nhất các API này thành một mô hình duy nhất nhằm đáp ứng mô hình lập trình hướng dịch vụ.

WCF có thể sử dụng các bản tin SOAP giữa hai tiến trình, do đó làm cho các ứng dụng dựa trên WCF có thể làm việc với các tiến trình khác thông qua việc giao tiếp sử dụng bản tin SOAP. Khi một tiến trình WCF liên lạc với một tiến trình không là WCF, các bản tin SOAP được mã hoá trên cơ sở XML, nhưng khi nó liên lạc với một tiến trình WCF khác, bản tin SOAP có thể được tối ưu hoá dựa trên mã hoá nhị phân.

* 1. *Tại sao lại sử dụng WCF?*

.NET 2.0 hỗ trợ rất nhiều phương pháp liên lạc giữa các ứng dụng khác nhau nhằm vào các mục tiêu khác nhau. Các phương pháp liên lạc này khá phức tạp và phải mất nhiều thời gian để làm chủ được công nghệ. Tuy nhiên kiến thức thu được từ việc triển khai một phương pháp ít có khả năng dùng được khi làm việc với phương pháp khác.

Với việc ra đời của WCF, mọi phương pháp liên lạc trước kia đều có thể thực hiện trên WCF. Do vậy nhà phát triển chỉ cần làm chủ được công nghệ WCF là có thể xây dựng các ứng dụng một cách nhanh chóng.

WCF là một mô hình lập trình cho phép nhà phát triển xây dựng các giải pháp dịch vụ đảm bảo tính ổn định, và bảo mật và thậm chí là đảm bảo giao dịch. Nó làm đơn giản hoá việc phát triển các ứng dụng nối kết và đưa ra cho nhà phát triển những giá trị mà có thể họ chưa nhận ra ngay, đó là cách tiếp cận phát triển hệ thống phân tán thống nhất, đơn giản, và quản lý được.

Do WCF được xây dựng trên cơ sở của .NET Framework 2.0 CLR, nó là tập các lớp cho phép các nhà phát triển xây dựng các ứng dụng hướng dịch vụ bằng môi trường lập trình quen thuộc của họ như VB.NET hay C#.

1. ***Kiến trúc tổ chức***

Hình sau mô tả các lớp chủ yếu trong kiến trúc của Windows Communication Foundation (WCF):



Hình 1 – Mô hình tổng quan về kiến trúc WCF

* 1. *Các hiệp nghị (Contracts)*

Các contract trong WCF cũng giống như các hợp đồng/hiệp định mà bạn ký trong đời sống thật. Một hợp đồng bạn ký có thể chứa các thông tin như kiểu công việc bạn sẽ làm, và những thông tin mà bạn muốn đưa ra cho các bên khác. WCF contract cũng chứa các thông tin tương tự như vậy. Contract định nghĩa các đặc tả trong hệ thống bản tin. Thông thường có các loại contract sau:

* Contract dữ liệu mô tả các tham số cho các bản tin mà một dịch vụ có thể tạo ra hay sử dụng. Các tham số bản tin được định nghĩa bằng các tài liệu sử dụng ngôn ngữ đặc tả XML Schema (XSD), điều này cho phép các hệ thống hiểu XML có thể xử lý tài liệu dễ dàng. Các dịch vụ khi liên lạc với nhau có thể không cần đồng ý với nhau về các kiểu, nhưng cần đồng ý về contract dữ liệu, nghĩa là đồng ý về các tham số và các kiểu trả về.
* Contract bản tin định nghĩa các phần có trong bản tin sử dụng các giao thức SOAP, và nó cho phép điều khiển sâu hơn tới các phần trong bản tin khi có yêu cầu sự chính xác như vậy.
* Contract dịch vụ đặc tả chi tiết các phương thức của dịch vụ, và được phân phối như là một giao diện trong các ngôn ngữ lập trình như Visual Basic hay Visual C#. Có thể hình dung về contract dịch vụ một cách gián tiếp như sau: „Đây là các kiểu dữ liệu của các bản tin của tôi, đây là nơi tôi cung cấp, và đây là các giao thức mà tôi có thể liên lạc”
* Các chính sách và các kết nối (bindings) mô tả các điều kiện cần có để giao tiếp với một dịch vụ. Các chính sách sẽ bao gồm cả các yêu cầu về bảo mật và các điều kiện khác cần phải có khi kết nối với một dịch vụ.
  1. *Dịch vụ thực thi (Runtime Service)*

Lớp dịch vụ thực thi chứa các hành xử sẽ xảy ra trong quá trình thực hiện của dịch vụ, nghĩa là các hành xử thực thi của dịch vụ. Ta sẽ thấy một số các hành xử như sau:

* **Throttling behavior:** Điều khiển luồng nhằm quy định xem có bao nhiêu bản tin được xử lý.
* **Error behavior:** Hành xử lỗi quy định những hành động khi lỗi xảy ra trong hệ thống.
* **Metadata behavior:** Hành xử với các siêu dữ liệu quy định xem làm thế nào và khi nào thì các siêu dữ liệu được đưa ra bên ngoài dịch vụ.
* **Instance behavior:** Hành xử thực thể quy định xem có bao nhiêu thực thể của dịch vụ đó được chạy.
* **Transaction behavior:** Hành xử giao dịch cho phép việc rollback các giao dịch nếu xảy ra lỗi.
* **Message inspection:** Kiểm tra bản tin đem lại cho dịch vụ khả năng kiểm tra tất cả hay một số phần của bản tin.
* **Dispatch behavior:** Khi một bản tin được xử lý bởi nền tảng WCF, dịch vụ Dispatch behavior xác định xem bản tin được xử lý như thế nào.
* **Concurrency behavior:** Hành xử đồng thời xác định xem việc xử lý thế nào với việc đa luồng của mỗi dịch vụ hay mỗi thực thể của dịch vụ. Hành xử này giúp cho việc điều khiển số lượng luồng có thể truy nhập tới một thực thể của dịch vụ.
* **Parameter filtering:** Khi một bản tin được đưa tới một dịch vụ, sẽ xảy ra một số hành động dựa trên nội dung phần đầu đề của bản tin. Phần lọc tham số sẽ thực hiện lọc các đầu đề bản tin và thực hiện các hành động đặt sẵn dựa trên việc lọc đầu đề bản tin.
  1. *Thông điệp/Bản tin (Message)*

Lớp bản tin là tập hợp các kênh. Mỗi kênh là một thành phần xử lý bản tin theo một cách nào đó. Một tập các kênh thường được gọi là ngăn xếp kênh. Các kênh làm việc trên bản tin và trên đầu đề của bản tin. Lớp này khác với lớp thực thi dịch vụ chủ yếu bởi sự khác nhau trong việc xử lý nội dung bản tin.

Có hai kênh khác nhau là kênh vận chuyển (transport channel) và kênh điều khiển (control channel):

* Kênh vận chuyển phụ trách việc đọc và ghi các bản tin từ mạng (network) hoặc từ một số điểm giao dịch bên ngoài).
* Kênh điều khiển thực hiện xử lý bản tin theo giao thức, thông thường làm việc bằng cách đọc và ghi thêm các đầu đề cho bản tin.
  1. *Lưu trữ và kích hoạt (Host and Activation)*

Nhìn một cách tổng thể thì một dịch vụ thực chất là một chương trình. Cũng giống như các chương trình khác, một dịch vụ cần phải chạy trong một tệp thực thi. Dịch vụ này thường được gọi là dịch vụ tự chứa.

Các dịch vụ còn có thể được chứa, hoặc chạy trong một tệp thực thi được quản lý bởi một agent bên ngoài như IIS hay Windows Activation Services (WAS). WAS cho phép WCF được kích hoạt một cách tự động khi phân phối tới một máy tính có chạy WAS.

1. ***Các tính năng đặc trưng***
   1. *Giao dịch (Transaction)*

Một giao dịch là một đơn vị của công việc. Một giao dịch đảm bảo chắc chắn rằng mọi thứ diễn ra trong giao dịch thành công hay thất bại đều là kết quả tổng thể. Ví dụ, nếu một giao dịch chứa ba mục công việc cần thực hiện, trong quá trình thực hiện giao dịch, một trong số các mục đó bị thất bại, khi đó cả ba mục sẽ là thất bại. Giao dịch chỉ thành công khi cả ba mục công việc đều thành công. Giao dịch thường thấy trong các thao tác với cơ sở dữ liệu.

WCF cho phép đưa vào việc xử lý giao dịch như trên với các liên lạc. Nhà phát triển có thể nhóm các liên lạc với nhau thành các giao dịch. Ở mức doanh nghiệp, tính năng này cho phép bạn thực hiện các công việc giao dịch qua các nền tảng khác nhau.

* 1. *Lưu trữ (Host)*

WCF cho phép các dịch vụ được chứa trong một số lớn các môi trường khác nhau, như Windows NT Services, Windows Forms, và ứng dụng console, cũng như ở trên IIS (Internet Information Server) và WAS (Windows Activation Services).

Chứa ứng dụng trên IIS còn có thêm các lợi điểm khác là dịch vụ có thể nhận các ưu điểm của rất nhiều tính năng có sẵn trên IIS, ví dụ IIS có thể điều khiển một cách tự động việc bắt đầu hay kết thúc một dịch vụ.

* 1. *Bảo mật (Security)*

Bảo mật là tính năng không thể thiếu trong WCF nói riêng và trong liên lạc nói chung. Trong WCF, tất cả mọi thứ từ các bản tin tới các client hay server đều phải xác thực và WCF có tính năng để đảm bảo rằng các bản tin không bị lẫn trong quá trình vận chuyển. WCF bao gồm việc đảm bảo tính toàn vẹn và bảo mật của bản tin.

WCF còn cho phép bạn tích hợp ứng dụng của bạn với cơ sở hạ tầng bảo mật sẵn có, bao gồm cả các chuẩn bên ngoài môi trường Windows bằng cách sử dụng các bản tin SOAP bảo mật.

1. ***Dịch vụ trong WCF (Services)***

Có ba kiểu dịch vụ trong WCF là: Định kiểu, Không định kiểu, và Bản tin định kiểu. Sau đây chúng ta sẽ xem xét chi tiết các kiểu dịch vụ của WCF.

* 1. *Định kiểu (Typed Service)*

Dịch vụ có định kiểu là loại dịnh vụ đơn giản nhất trong số ba kiểu dịch vụ. Tuy vậy nó cung cấp phần lớn các tính năng cần thiết để bạn phát triển các dịch vụ WCF.

Một thuật ngữ hay dùng cho dịch vụ định kiểu là “mô hình tham số”, mô hình này định nghĩa một cách chính xác kiểu dịch vụ này làm gì. Các dịch vụ định kiểu không giới hạn ở việc định kiểu của các tham số hay kết quả trả về. Các dịnh vụ kiểu này có thể chấp nhận các kiểu đơn giản cũng như phức hợp. Tuy nhiên khi truyền các tham số hay trả về giá trị kiểu phức hợp, ta cần phải định nghĩa data contract cho các kiểu đó.

Dịch vụ định kiểu trong Windows Communication Foundation xử lý tất cả các bản tin, bạn không cần phải làm việc trực tiếp ở mức bản tin. Ví dụ sau đây mô tả một dịch vụ định kiểu. Contract dịch vụ định nghĩa 2 thao tác. Thao tác thứ nhất không cần tham số, thao tác thứ 2 chấp nhận 2 tham số kiểu là int (số nguyên):

*[ServiceContract]*

*public interface IService1*

*{*

*[OperationContract]*

*string SayHello();*

*[OperationContract]*

*string GetData(int firstValue, int secondValue);*

*}*

Dịch vụ định kiểu còn hỗ trợ các kiểu tham số với từ khoá ref và out.

* 1. *Không định kiểu (Untyped Service)*

Các dịch vụ không định kiểu hơi phức tạp hơn so với dịch vụ định kiểu do chúng làm việc trực tiếp với bản tin. Trong kiểu dịch vụ này, bạn định nghĩa các bản tin và nội dung của chúng. Nghĩa là với kiểu dịch vụ này, bạn làm việc ở mức bản tin, các đối tượng bản tin được chuyển qua lại giữa client và dịch vụ, và dịch vụ có thể trả về một đối tượng bản tin nếu được yêu cầu. Hơn nữa, kiểu dịch vụ này cho bạn khả năng truy nhập tới nội dung bản tin.

* 1. *Bản tin định kiểu (Typed Message Service)*

Trong một dịch vụ bản tin có định kiểu (typed message service) còn được gọi là “mô hình contract bản tin” bạn định nghĩa các bản tin và nội dung của chúng. Ở đây, các bản tin được mô tả với thuộc tính MessageContract, là các contract bản tin thường được sử dụng để định nghĩa các lớp bản tin. Các bản tin này sẽ được sử dụng để gửi qua lại trong các thao tác của dịch vụ.

1. ***Kiến trúc Client trong WCF***
   1. *Khái niệm cơ bản*

Một client trong Windows Communication Foundation là một chương trình sử dụng các chức năng cung cấp bởi một dịch vụ WCF. Chương trình client sẽ liên lạc với dịch vụ thông qua điểm cuối dịch vụ. Để làm được việc này client cần phải biết một số thông tin về dịch vụ như địa chỉ của điểm cuối, binding mà dịch vụ sử dụng, và contract dịch vụ. Các thành phần này đã được thảo luận ở các bài trước.

Một trong những thứ bạn có thể thấy trong kiến trúc của client là kênh thông tin được xây dựng dựa trên các cấu hình binding, những cấu hình này được quy định trong tệp tin cấu hình. Những thông tin cấu hình này chính là các thông tin đã được giới thiệu trong phần nói về bindings. Những bindings này cho phép client và dịch vụ liên lạc với nhau một cách hiệu quả.

Điểm thứ hai bạn có thể thấy là cài đặt của giao diện IClientChannel. Giao diện này định nghĩa các thao tác cho phép nhà phát triển điều khiển các chức năng của kênh, như đóng phiên làm việc của client và huỷ một kênh (để thu hồi tài nguyên). Nó đưa ra các phương thức và hàm của lớp được chỉ ra sau đây: System.ServiceModel.ChannelFactory.

Cuối cùng là bạn có thể thấy contract dịch vụ được tạo ra tự động, contract dịch vụ sẽ cung cấp tính năng chuyển lời gọi hàm ở phía client thành các bản tin đi, và chuyển các bản tin tới thành thông tin mà chương trình client có thể sử dụng dưới dạng giá trị trả về hay tham số đầu ra của hàm.

Các client liên lạc với điểm cuối dịch vụ thông qua một proxy, như hình dưới. Sự liên lạc được thực hiện thông qua một kênh. Sau khi proxy và kênh được tạo ra, client có thể truy xuất các phương thức có ở điểm cuối đó.



Hình 2 – Mô hình giao tiếp giữa Client và dịch vụ của nhà cung cấp

* 1. *Các đối tượng phía Client*
     1. ICommunicationObject

Giao diện ICommunicationObject là một trong những thành phần lõi để định nghĩa chức năng liên lạc. Nhiệm vụ của đối tượng này là định nghĩa contract cho trạng thái cơ bản của tất cả các đối tượng trong hệ thống; ví dụ, trạng thái đối tượng liên lạc đóng hay mở, hoặc trong trạng thái đang mở hoặc đang đóng. Những đối tượng này bao gồm các kênh, các đầu nghe (listeners), các dispatchers, factories, và chứa dịch vụ (service host).

Một chuyển trạng thái là việc chuyển từ trạng thái này sang trạng thái khác; ví dụ kênh liên lạc chuyển từ trạng thái “đang mở” sang trạng thái “mở”.

Giao diện này định nghĩa các phương thức cho việc khởi tạo chuyển trạng thái:

* **Open**: Làm cho đối tượng liên lạc chuyển trạng thái từ “Created” sang “Opened”.
* **Close**: Làm cho đối tượng liên lạc chuyển từ trạng thái hiện tại sang trang thái “Opened”.
* **Abort**: Làm cho đối tượng liên lạc chuyển ngay lập tức từ trạng thái hiện tại sang trạng thái “Closed”.

Ngoài các phương thức khởi tạo ở trên, giao diện ICommunicationObject còn định nghĩa các sự kiện để thông báo việc thay đổi trạng thái:

* **Opening**: Sự kiện này được kích hoạt khi đối tượng chuyển trạng thái từ Created sang Opened, sự kiện này xảy ra khi hàm Open hoặc BeginOpen được gọi.
* **Closing**: Sự kiện này được kích hoạt khi đối tượng chuyển trạng thái từ Opened sang Closed, sự kiện này xảy ra khi hàm Close hoặc BeginClose được gọi.
* **Opened**: Sự kiện này được kích hoạt khi đối tượng kết thúc việc chuyển trạng thái từ Opening sang Opened
* **Closed**: Sự kiện này được kích hoạt khi đối tượng kết thúc việc chuyển trạng thái từ Closing sang Closed
* **Faulted**: Sự kiện này được kích hoạt khi đối tượng chuyển sang trạng thái Faulted.

Ngoài ra còn có một tập phương thức dị bộ cho các phương thức Open và Close:

* **BeginOpen**: Bắt đầu thao tác dị bộ để mở một đối tượng liên lạc
* **BeginClose**: Bắt đầu thao tác dị bộ để đóng một đối tượng liên lạc.
* **EndOpen**: Hoàn tất thao tác dị bộ để mở một đối tượng liên lạc
* **EndClose**: Hoàn tất thao tác dị bộ để đóng một đối tượng liên lạc

Giao diện ICommunicationObject có một thuộc tính trạng thái State, với kiểu là CommunicationState, thuộc tính này được dùng để chỉ ra trạng thái hiện tại của đối tượng.

Khi một đối tượng cài đặt ICommunicationObject được khởi tạo, trạng thái mặc định là Created chứ không phải là trạng thái Opened. Trong khi đang ở trạng thái Created, ta chỉ có thể thiết lập cấu hình cho đối tượng ICommunicationObject chứ không thể dùng nó để gửi hay nhận bản tin. Để có thể gửi hay nhận bản tin, đối tượng cần phải chuyển sang trạng thái Opened, nhưng khi ở trạng thái này, ta không thể thiết lập cấu hình cho nó nữa.

Để chuyển đối tượng sang trạng thái Opened, ta gọi hàm Open. Đối tượng sẽ giữ trạng thái này cho tới khi nó chuyển hoàn toàn sang trạng thái Closed. Hàm Close được dùng để chuyển đối tượng sang trạng thái Closed, tuy nhiên nó đợi cho các thao tác hiện tại của đối tượng hoàn tất mới thực hiện chuyển trạng thái. Hàm Abort thì không như vậy, nghĩa là những thao tác chưa hoàn tất sẽ bị huỷ bỏ, và đối tượng được chuyển ngay sang trạng thái Closed.

* + 1. IExtensibleObject

Giao diện IExtensibleObject cung cấp cách mở rộng cho client. Trong WCF, cách mở rộng đối tượng được sử dụng để thêm các tính năng mới cho các lớp thực thi đã có, qua đó mở rộng các thành phần hiện tại cũng như thêm mới các trạng thái cho một đối tượng.

Giao diện này được đưa ra duy nhất một thuộc tính để cung cấp tính năng này, Extensions, với kiểu là IExtensionCollection. Thuộc tính này được sử dụng để trả về một tập hợp các đối tượng mở rộng có thể được sử dụng để mở rộng các lớp thực thi đã có.

* 1. *Các phương thức liên lạc của Client*
     1. Một chiều (One way)

Liên lạc một chiều là liên lạc chỉ theo một hướng duy nhất. Là hướng từ phía client tới dịch vụ. Không có trả lời từ phía dịch vụ và client hoàn toàn không trông mong nhận được phản hồi. Theo cách này, client gửi đi một bản tin và tiếp tục thao tác của mình.

Do không có phản hồi từ dịch vụ trong liên lạc một chiều, client sẽ không biết được liệu có lỗi xảy ra trong quá trình liên lạc hay không, và nó cũng hoàn toàn không biết liệu yêu cầu có thành công hay không. Để tạo ra dịch vụ cho liên lạc một chiều, ta đặt tham số IsOneWay của thuộc tính mô tả OperationContract là True. Điều này sẽ thông báo cho dịch vụ biết là không cần có phản hồi. Đoạn mã nguồn sau đây biểu diễn cách thiết lập một liên lạc một chiều, thao tác AddPerson, EditPerson, và DeletePerson:

*[ServiceContract]*

*public interface IStaffInformation*

*{*

*[OperationContract]*

*bool HasPerson(int personId);*

*[OperationContract]*

*Person GetPerson(int personId);*

*[OperationContract]*

*Person[] GetAll();*

*[OperationContract(IsOneWay=true]*

*void AddPerson(Person person);*

*[OperationContract(IsOneWay=true]*

*void EditPerson(int personId, Person person);*

*[OperationContract(IsOneWay=true]*

*void DeletePerson(int personId);*

*}*

* + 1. Yêu cầu – Hồi đáp (Request – Reply)

Liên lạc theo kiểu yêu cầu-trả lời thực hiện như sau: client gửi đi một bản tin cho dịch vụ, nó sẽ chờ để nhận một phản hồi từ dịch vụ. Kiểu liên lạc này đồng thời còn có nghĩa là khi client gửi đi bản tin, nó sẽ không thực hiện thao tác khác cho tới khi nó nhận được một phản hồi từ phía dịch vụ.

Trong Windows Communication Foundation, có hai cách để quy định kiểu liên lạc yêu cầu-trả lời. Cách thứ nhất là đặt giá trị false cho tham số IsOneWay của OperationContract. Thực chất giá trị mặc định của tham số IsOneWay là false, do vậy cách thứ hai để quy định kiểu liên lạc yêu cầu-trả lời là không dùng tham số IsOneWay nữa. Do vậy các thao tác trên dịch vụ của WCF mặc định là liên lạc kiểu hỏi-trả lời.

Kiểu liên lạc hỏi-trả lời là kiểu liên lạc mặc định của WCF nên thông thường bạn cài đặt dịch vụ WCF hay sử dụng dịch vụ WCF, bạn liên lạc theo kiểu hỏi-trả lời. Các ví dụ ở các bài trước đều theo kiểu làm việc này. Hai kiểu liên lạc tiếp theo đây là liên lạc song công và liên lạc dị bộ ta ít gặp hơn.

* + 1. Song công (Duplex)

Liên lạc song công là khả năng mà cả client và dịch vụ đều có thể khởi tạo liên lạc, cũng như phản hồi các bản tin đến; nói cách khác đây là liên lạc hai chiều. Với liên lạc song công, dịch vụ không chỉ có thể trả lời các bản tin đến mà còn có thể khởi tạo liên lạc với client bằng cách gửi bản tin yêu cầu một phản hồi từ phía client.

Để cấu hình một liên lạc song công, cần có thay đổi từ cả hai phía: dịch vụ và client.

1. Phía dịch vụ

Để thực hiện liên lạc song công, phía dịch vụ cần có hai giao diện. Mục tiêu của giao diện thứ nhất là sử dụng trong liên lạc từ client tới dịch vụ, nghĩa là nó được sử dụng để nhận các bản tin từ phía client, giống như các ví dụ từ trước tới nay. Giao diện thứ hai, còn gọi là callback interface, được sử dụng trong liên lạc từ dịch vụ tới client để gửi bản tin từ phía dịch vụ tới client. Điều quan trọng là các thao tác ở trong các giao diện phải được định nghĩa là các thao tác một chiều.

1. Phía Client

Để thực hiện liên lạc song công, phía client cũng phải nhận một phần trách nhiệm trong việc cài đặt, và cũng phải cài đặt một callback contract. Client thực hiện bằng cách cài đặt callback interface của duplex contract.

* + 1. Dị bộ (Asynchronous)

Liên lạc dị bộ được thực hiện bằng cách gọi các hàm dị bộ. Việc liên lạc dị bộ cho phép chương trình tiếp tục thực hiện những việc khác trong khi hàm được gọi vẫn đang thực hiện.

Cũng giống như trong liên lạc song công, các thao tác dị bộ yêu cầu một số thay đổi về phía client và dịch vụ.

1. ***Mô hình lập trình WCF***
   1. *Lựa chọn phương pháp tiếp cận*

Nếu các bạn đã học qua môn học về các phương pháp lập trình hẳn sẽ thấy có 2 phương pháp chính là hướng thủ tục và hướng đối tượng. Và phương pháp hướng đối tượng trong thời gian gần đây được phát triển rất mạnh và được hỗ trợ ngay trong các ngôn ngữ lập trình như C# hay VB.NET. Khi làm việc với các dịch vụ web bạn đã làm quen với một phương pháp nữa là lập trình hướng dịch vụ (Service-oriented programming). Microsoft đã cung cấp nhiều công cụ trên .NET Framework để hỗ trợ phương pháp lập trình này thông qua các lớp trong không gian tên: System.Web.Services.

Như vậy khi làm việc với WCF bạn có hai lựa chọn là hướng đối tượng và hướng dịch vụ. Vậy ta nên sử dụng phương pháp nào? Câu trả lời là dùng cả hai. Nói một cách đơn giản là phương pháp hướng đối tượng được sử dụng để phát triển các ứng dụng trên desktop, còn phương pháp hướng dịch vụ được sử dụng để kết nối các ứng dụng đó với nhau. Điều quan trọng ở đây là làm sao để hiểu được sự khác nhau giữa hai phương pháp và hiểu được khi nào chúng được sử dụng và sử dụng như thế nào đồng thời cũng phải hiểu về các lợi ích chúng cung cấp.

Về hướng đối tượng có thể hiểu như sau. Các ứng dụng hướng đối tượng là hai hay nhiều lớp phụ thuộc lẫn nhau và chia sẻ chung các kiểu dữ liệu. Những lớp này liên lạc với nhau thông qua các lời gọi các hàm mà lớp đối tượng cung cấp.

Các ứng dụng hướng dịch vụ là các chương trình không biết gì về nhau. Mỗi ứng dụng liên lạc với ứng dụng khác thông qua các bản tin. Điểm đặc biệt là các bản tin này được gửi từ một ứng dụng sang ứng dụng khác mà không quan tâm tới nền tảng mà dịch vụ đang chạy.

Khi phát triển các dịch vụ WCF, điều quan trọng là cần hiểu sự liên kết giữa hướng đối tượng và hướng dịch vụ. Khi làm việc với .NET Framework bạn chắc chắn rất quen thuộc với thuật ngữ lớp (class) và giao diện (interface). Các thuật ngữ này vẫn được sử dụng khi phát triển dịch vụ WCF. Các lớp và giao diện là phần hướng đối tượng trong WCF, còn phần hướng dịch vụ trong WCF sẽ được thấy khi bạn đưa vào các thuộc tính WCF để định nghĩa các thực thể.

* 1. *Mô hình dịch vụ*

Nếu bạn đã từng làm việc với dịch vụ web, bạn sẽ thấy mô hình này quen thuộc với bạn theo một cách nào đó. Khi bạn tạo một dịch vụ web, bạn thực sự tạo ra một dịch vụ (service). Dịch vụ web chứa một tài liệu XML để mô tả tất cả mọi thứ cần biết về dịch vụ đó. Tài liệu này được mô tả bằng ngôn ngữ Web Service Description Language (ngôn ngữ mô tả dịch vụ web). Nó chứa ba phần:

* **Service (dịch vụ)**: Chứa thông tin về vị trí của dịch vụ
* **Binding**: Chứa thông tin về cách liên lạc với dịch vụ, như dịch vụ sử dụng giao thức gì, vv.
* **PortType (kiểu cổng)**: Giải thích về dịch vụ sẽ làm gì

Mô hình dịch vụ trên WCF cũng tương tự như với mô hình dịch vụ web. Điểm khác biệt là ở cách đặt tên. Trong WCF các phần không được gọi là service, binding, và portType mà được gọi tương ứng là address (địa chỉ), binding, và contract.

Mô hình dịch vụ WCF được cung cấp trong không gian tên System.ServiceModel. Không gian tên này chứa rất nhiều lớp, nhưng bạn hoàn toàn không cần biết toàn bộ chúng.

* 1. *Các phương pháp lập trình với WCF*
     1. Phương pháp khai báo (Declarative Programming)

Lập trình khai báo đạt được thông qua các thuộc tính. Những thuộc tính này được sử dụng để định nghĩa các contract và xác định hành xử của dịch vụ. Chúng được sử dụng để xác định thêm các tham số để thay đổi các chi tiết của contract và hành xử dịch vụ.

Thuộc tính ServiceContract dùng để quy định là giao diện này định nghĩa các chức năng của một dịch vụ. Thuộc tính OperationContract được sử dụng ở các hàm để quy định rằng hàm này được khai báo là một phần của dịch vụ. Đó là tất cả những gì cần để tạo ra một dịch vụ WCF.

Thêm nữa, bạn không nhất thiết phải sử dụng các giao diện (interface) khi cài đặt một dịch vụ, điều này cũng giống như việc bạn không cần phải sử dụng giao diện để định nghĩa một lớp. Tuy vậy bạn nhất thiết phải quy định phần nào thuộc về dịch vụ. Bạn có thể định nghĩa những phần khác cần cho giao diện, nhưng chỉ những hàm (phương thức) có gắn thuộc tính [OperationContract].

* + 1. Phương pháp lập trình trực tiếp (Explicit Programming)

Là phương pháp lập trình hướng đối tượng, bạn làm việc trực tiếp với các lớp và giao diện cung cấp bởi mô hình đối tượng của WCF. Làm việc trực tiếp với mô hình đối tượng cho phép nhà phát triển tính linh hoạt cao hơn và khả năng điều khiển tốt hơn thông qua mã nguồn của họ. Thêm nữa nó cho phép điều khiển sâu hơn rất nhiều so với phương pháp khai báo và phương pháp sử dụng tập tin cấu hình.

* + 1. Phương pháp sử dụng tập tin cấu hình (XML File Config)

Cũng giống như phương pháp khai báo, có rất nhiều thứ mà bạn có thể quy định liên quan đến hành xử của một dịch vụ thông qua tập tin cấu hình của dịch vụ. Điều hay trong cách tiếp cận này là những thay đổi ở tập tin cấu hình hoàn toàn không cần phải biên dịch lại dịch vụ mới sử dụng được.

1. ***Bảo mật trong WCF (Security)***
   1. *Ý nghĩa bảo mật trong WCF*

Do WCF là một nền tảng lập trình phân tán dựa trên các bản tin SOAP, với việc sử dụng WCF, các ứng dụng bạn tạo ra có thể tạo và xử lý các bản tin từ một số không hạn chế các dịch vụ và client khác. Trong các ứng dụng này, các bản tin có thể được truyền từ điểm này sang điểm khác, thông qua các thiết bị như tường lửa, router, switch, hay qua Internet, và qua một loạt các điểm trung chuyển SOAP. Điều này tạo ra không ít mối đe doạ tới an ninh của bản tin. Những ví dụ sau đây cho ta thấy một số các mối đe doạ thường thấy khi trao đổi bản tin giữa các ứng dụng, những mối đe doạ này hoàn toàn có thể loại bỏ được nhờ vào sử dụng tính năng bảo mật trong WCF:

* Quan sát các bản tin trên mạng để lấy ra các thông tin nhạy cảm. Ví dụ ở máy client thực hiện login vào một hệ thống sử dụng chế độ gửi tên tài khoản và mật khẩu dạng text không mã hoá. Hacker hoàn toàn có thể bắt được bản tin đó và trích ra thông tin về tài khoản cùng với mật khẩu.
* Đóng giả một dịch vụ mà client không hề biết. Việc này cũng tương tự như web phishing, nghĩa là làm giả một trang web giống như trang web mà người dùng quen thuộc (như trang web yahoo hay trang web ngân hàng). Người dùng sẽ nhập thông tin về tài khoản cùng với mật khẩu đề đăng nhập vào trang giả đó. Khi đó hacker sẽ có được các thông tin này.
* Thay đổi nội dung bản tin. Hacker hoàn toàn có thể thay đổi nội dung của một bản tin mà client lẫn dịch vụ không biết.

Việc đảm bảo bảo mật cho các bản tin trao đổi giữa client với dịch vụ WCF cần phải chú trọng ở những điểm sau:

* Xác thực điểm cuối dịch vụ
* Xác thực client
* Tính nhất quán của bản tin
* Tính bảo mật của bản tin
* Phát hiện replay (hiện tượng lặp lại yêu cầu của client hoặc dịch vụ mà thực chất client/dịch vụ không đưa ra)
  + 1. Tích hợp với các kiến trúc bảo mật có sẵn

WCF hoàn toàn có thể làm việc với các giải pháp bảo mật có sẵn như Secure Sockets Layer (SSL) hoặc giao thức Kerbeos. Ngoài ra nó cũng có thể làm việc với kiến trúc bảo mật đang sử dụng như domain trên Windows sử dụng Active Directory. Ngoài việc hỗ trợ các giải pháp bảo mật thường thấy, WCF còn tích hợp với các mô hình bảo mật sẵn có ở tầng vận chuyển và có thể chuyển hạ tầng sẵn có sang các mô hình mới hơn dựa trên bảo mật các bản tin SOAP.

* + 1. Tích hợp với mô hình xác thực có sẵn

Một phần quan trọng trong bất kỳ mô hình bảo mật truyền tin nào là khả năng xác định và xác thực các thực thể trong quá trình trao đổi dữ liệu. Các thực thể trong quá trình này sử dụng các “định danh điện tử”, hay còn gọi là credentials, để xác thực chúng với đối tượng đang trao đổi. Với sự phát triển của các nền tảng phân tán, có rất nhiều cách để xác thực các credentials được tạo ra. Lấy ví dụ, trên Internet cách thông dụng nhất là sử dụng tên tài khoản cùng với mật khẩu để xác thực người dùng. Trong mạng intranet, mô hình thông dụng là sử dụng Kerberos domain controller để xác thực người dùng cùng với dịch vụ. Do vậy với cùng một dịch vụ, ta có thể có các cách xác thực khác nhau cho dịch vụ đó tuỳ thuộc vào việc dịch vụ đó được sử dụng ở môi trường nào. WCF hỗ trợ rất nhiều các mô hình xác thực khác nhau:

* Anonymous caller (Người gọi vô danh)
* Username client credential
* Certificate client credential
* Windows (Kerberos và NT LanMan – NTML)
  + 1. Làm việc liên môi trường

Trong thế giới với nhiều nền tảng như hiện nay, các nền tảng tính toán/truyền thông phân tán cần phải làm việc liên thông với các công nghệ khác nhau mà các nhà cung cấp hỗ trợ. Cũng vì vậy mà bảo mật cũng cần phải làm việc liên môi trường (interoperable).

Để thực hiện các hệ thống bảo mật interoperable, các công ty sử dụng dịch vụ Web với một loạt các chuẩn khác nhau. Về bảo mật thì có thể kể ra một số chuẩn như sau: WS-Security, SOAP Message Security, WS-Trust, WS-SecureConversation, và WS-SecurityPolicy.

Với các dịch vụ WCF ta có thể sử dụng WSHttpBinding để hỗ trợ WS-Security 1.1 và WS-SecureConversation.

* 1. *Các lĩnh vực bảo mật của WCF*

Bảo mật trong WCF chia ra thành ba vùng chức năng: transfer security (bảo mật truyền thông), (access control) điều khiển truy nhập, và auditing (ghi vết).

* + 1. Transfer Security

Bảo mật truyền thông bao gồm ba chức năng chính: sự nhất quán, sự bảo mật, và sự xác thực. Sự nhất quán là khả năng phát hiện liệu bản tin có bị thay đổi hay không. Sự bảo mật là khả năng giữ cho bản tin không đọc được bởi những người không đủ thẩm quyền; điều này có được nhờ vào cryptography (mã mật – mật mã). Sự xác thực là khả năng xác minh được một định danh có thực đúng hay không. Kết hợp ba chức năng này cho ta đảm bảo rằng các bản tin được gửi đi một cách an toàn, đến đúng nơi cần đến.

* + 1. Các chế độ bảo mật vận chuyển và bản tin

Có hai phương pháp chính dùng để thực hiện bảo mật truyền thông trong WCF là chế độ bảo mật ở tầng vận chuyển (transport security mode) và chế độ bảo mật ở bản tin (message security mode):

* *Transport security mode* sử dụng các giao thức ở tầng vận chuyển như HTTPS để đảm bảo bảo mật. Chế độ này có ưu điểm là được sử dụng rất nhiều ở các nền tảng khác nhau, và độ phức tạp tính toán ít hơn. Tuy vậy nhược điểm là chỉ đảm bảo bảo mật các bản tin từ điểm-tới-điểm.
* *Message security mode* sử dụng chuẩn WS-Security để đảm bảo bảo mật. Do bảo mật bản tin được áp dụng trực tiếp lên các bản tin SOAP và được chứa trong các vỏ (envelope) SOAP, cùng với dữ liệu của chương trình, nó có ưu điểm là không phụ thuộc vào giao thức vận chuyển, dễ mở rộng, đảm bảo bảo mật từ đầu cuối-tới-đầu cuối (thay vì điểm-tới-điểm). Nhược điểm của nó là chậm hơn so với chế độ transport security do nó phải làm việc với XML trong bản tin SOAP.
  + 1. Điều khiển truy nhập

Điều khiển truy nhập còn được biết tới như là authorization (nhận thực). Authorization cho phép những người dùng khác nhau có các quyền khác nhau để xem dữ liệu. Trong WCF, các tính năng điều khiển truy nhập được cung cấp dựa vào sự tích hợp với CLR (common language runtime) thông qua lớp thuộc tính PrincipalPermissionAttribute và qua một loạt các hàm API.

* + 1. Auditing

Auditing là quá trình ghi lại các sự kiện bảo mật vào hệ thống log của hệ điều hành Windows (Windows event log). Bạn có thể ghi lại các sự kiện có liên quan tới bảo mật như là xác thực lỗi hay thành công. Bạn có thể xem thêm trong bài How to: Audit Windows Communication Foundation Security Events

1. **Ứng dụng kiểm thử**