

Naruliev S.G.
Kiến trúc “1C:DOANH NGHIỆP”
như là một sản phẩm của tư duy công nghiệp

Trong bài viết của người phụ trách khối phát triển giải pháp kinh tế của hãng “1C”, Sergey Nuraliev, có mô tả chi tiết các đổi mới về kiến trúc và công nghệ được thực thi trong hệ thống “1C:DOANH NGHIỆP” mà tổng thể đã tạo ra một loạt các công nghệ mới để phát triển giải pháp kinh tế ứng dụng với chất lượng mới.

Bản dịch tiếng Việt do Công ty cổ phần Hệ thống 1-V (1VS JSC) thực hiện với sự đồng ý của hãng “1C”

© ZAO “1C”, 2004

© 1VS JSC, 2007

Kiến trúc “1C:DOANH NGHIỆP” như là một sản phẩm của tư duy công nghiệp

Mục lục

Kiến trúc “1C:DOANH NGHIỆP” như là một sản phẩm của tư duy công nghiệp.....	2
Mục lục.....	2
Kiến trúc “1C:DOANH NGHIỆP” như là một sản phẩm của tư duy công nghiệp.....	3
Nhìn từ góc độ khác.....	3
Chúng ta phấn đấu vì điều gì.....	3
Nền tảng công nghệ và các giải pháp ứng dụng.....	4
Metadata, phương pháp mô tả giải pháp ứng dụng.....	6
Xây dựng ứng dụng trên cơ sở mô hình.....	7
Quản lý dữ liệu.....	7
Nguyên mẫu chuẩn của các đối tượng ứng dụng.....	11
Các đối tượng và cơ chế ứng dụng.....	13
Mô hình cao cấp của giao diện.....	16
Các cơ chế thông minh để lập báo cáo.....	19
Xây dựng hệ thống thông tin phân tán và tích hợp.....	22
Cài đặt và cập nhật các giải pháp ứng dụng.....	23
Và hơn nữa.....	24
Tổng kết.....	25

Kiến trúc “1C:DOANH NGHIỆP” như là một sản phẩm của tư duy công nghiệp

Nuraliev S.G

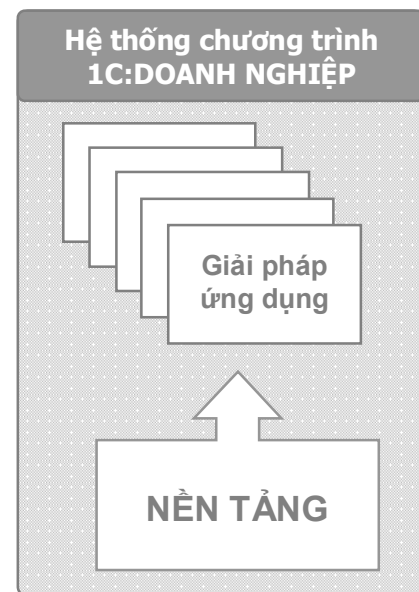
Nhìn từ góc độ khác

Ngày nay cứ mỗi sản phẩm thứ hai do con người tạo ra để đáp ứng tiêu dùng đều có liên quan đến kỹ thuật công nghệ cao. Có nghĩa là sản phẩm không chỉ phải đáp ứng một số nhu cầu cụ thể của con người, mà còn phải là hiện thân của ý tưởng khoa học và kỹ thuật. Hiển nhiên, trong một số trường hợp, những ý tưởng đã được đưa vào sản phẩm và sự thành công xác định hoàn toàn bản chất và giá trị sử dụng của nó.

Trong bài này chúng tôi không tập trung vào việc mô tả các khả năng của sản phẩm «1C:DOANH NGHIỆP» mà chính là những phát triển của công nghệ mới, những yếu tố quyết định thành công cho kỹ thuật lập trình giải pháp ứng dụng và chất lượng của chính các ứng dụng mới đó.

Trước hết, chúng tôi muốn có một vài nhận xét nhỏ về sử dụng thuật ngữ. Bài viết này không chỉ dành riêng cho những người hiện đang sử dụng chương trình của hệ thống «1C:DOANH NGHIỆP». Chúng tôi nghĩ, có thể nó sẽ gây được sự chú ý của nhiều chuyên gia, những người đang quan tâm đến sự phát triển công nghệ của các ứng dụng kinh tế và quản lý mà không phụ thuộc vào việc họ đang sử dụng phần mềm quản lý nào. Cho nên, trong bài này chúng tôi sẽ hạn chế tối đa việc sử dụng các thuật ngữ có liên quan đến «1C:DOANH NGHIỆP». Trong một số trường hợp, chúng tôi sẽ sử dụng một số thuật ngữ khác nhưng gần với các khái niệm tương đương và có kèm các giải thích cần thiết.

Cần nói qua về tên gọi của sản phẩm. «1C:DOANH NGHIỆP», đây là hệ thống các chương trình bao gồm trong đó nền tảng công nghệ và bộ các giải pháp ứng dụng được xây dựng dựa trên nền tảng đó, được sử dụng cho nhiều lĩnh vực và các doanh nghiệp có quy mô khác nhau. Trong bài này chúng tôi sẽ nói chủ yếu về nền tảng công nghệ, bởi vì các giải pháp ứng dụng được xây dựng dựa trên nền tảng đó.



Chúng ta phấn đấu vì điều gì

Trước hết cần phải xác định, đâu là mục tiêu chính của nền tảng công nghệ «1C:DOANH NGHIỆP». Nếu điếm qua lịch sử phát triển máy vi tính và kỹ thuật lập trình thì có thể thấy rằng, ngoài việc nâng cao hiệu suất, tăng khối lượng thông tin cần xử lý, nâng cao tính thân thiện của giao diện..., còn có sự mong muốn chính đáng là nâng cao mức độ trừu tượng hóa các hệ thống chương trình. Ở một mức độ nào đó, khuynh hướng này là một thuộc tính độc nhất vô nhị vốn có của họ máy tính; trong khi ở các lĩnh vực hoạt động khác của con người, các mục tiêu phát triển chiến lược thường mang tính thực dụng hơn. Có thể dễ dàng quan sát thấy quá trình phát triển mức độ trừu tượng hóa một cách đơn giản như sau: bắt đầu việc lập trình ở mức độ kết nối các đầu mối, tiến đến ngôn ngữ máy, hợp ngữ, sau nữa là ngôn ngữ lập trình có cấu trúc ... Mỗi bước đều đánh dấu sự nâng cao mức độ trừu tượng hóa tương tác của con người (là lập trình viên và người sử dụng) với máy vi tính.

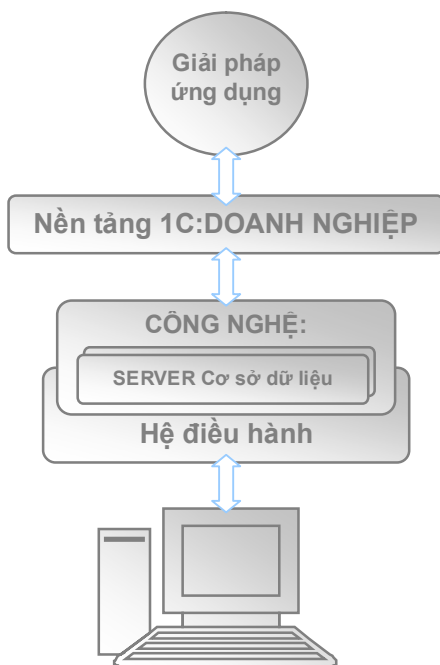
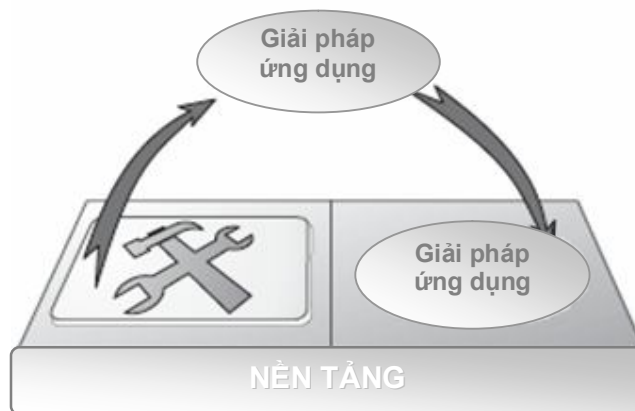
Như vậy, nhiệm vụ chính của nền tảng công nghệ «1C:DOANH NGHIỆP», trước hết là thực hiện các quan điểm trên khi lập trình và cả khi sử dụng các giải pháp ứng dụng. Dĩ nhiên, để làm được

việc đó thì cần phải giải quyết một loạt các nhiệm vụ truyền thống có liên quan tới hiệu suất, giao diện và các tính năng khác... Nhưng chính việc nâng cao mức độ trừu tượng hóa lại cho phép chuyển từ khái niệm kỹ thuật ở mức thấp đến nội dung ở mức cao hơn, tức càng gần với ngôn ngữ được các chuyên gia và người sử dụng lựa chọn. Rất cuộc, tất cả những điều đó cho phép đẩy nhanh tốc độ và thống nhất của chính việc tạo dựng việc bảo trì/bảo hành hệ thống.

Nền tảng công nghệ và các giải pháp ứng dụng

Có thể nói nền tảng công nghệ và các giải pháp ứng dụng là khái niệm nổi bật trong kiến trúc của hệ thống «1C:DOANH NGHIỆP».

Đĩ nhiên, bất kỳ hãng phát triển phần mềm đều sở hữu riêng các phương tiện công nghệ và công cụ đủ mạnh để xây dựng cho mình các giải pháp. Thế nhưng, khi xây dựng «1C:DOANH NGHIỆP», ngay từ đầu chúng tôi đã định hướng tạo ra nền tảng công nghệ có đầy đủ các chức năng hoàn chỉnh để sử dụng làm cơ sở phát triển các giải pháp ứng dụng đa dạng nhất không chỉ cho riêng hãng “1C” mà còn cho nhiều công ty lập trình khác có quan tâm và am hiểu với đặc thù của hệ thống «1C:DOANH NGHIỆP». Đó cũng là câu trả lời vì sao mà ngay từ đầu nền tảng công nghệ đã được tạo dựng như một sản phẩm đóng gói bao gồm trong đó tất cả các công nghệ cần thiết để phục vụ việc khai thác các giải pháp ứng dụng cùng với các công cụ phục vụ việc lập trình và chỉnh sửa chúng.



«1C:DOANH NGHIỆP» được chia thành hai phần rõ rệt: nền tảng và các giải pháp ứng dụng. Nền tảng có thể gọi là “khung làm việc” (“framework”) mà trong đó giải pháp ứng dụng hoạt động. Một mặt, chúng ta có thể coi framework là nền móng để xây dựng trên đó các ứng dụng, mặt khác nó có thể được coi là môi trường thực thi các ứng dụng đó. Ngoài ra dĩ nhiên, nền tảng còn bao gồm trong đó cả các công cụ cần thiết để lập trình, quản trị hệ thống và hỗ trợ các giải pháp ứng dụng. Ứng dụng này là một thực thể độc lập và có thể xem như một sản phẩm chương trình riêng biệt, nhưng lại hoàn toàn dựa trên công nghệ của nền tảng.

Có thể nói, khái niệm nền tảng và nền tảng để xây dựng các ứng dụng hiện tại được xem là khái niệm chung và được luận giải rộng hơn nhiều so với khả năng làm việc bình thường của nó trong một hệ điều hành. Có thể hiểu nền tảng là môi trường thực thi và tập hợp (bộ) các công nghệ được sử dụng để làm cơ sở xây dựng các ứng

dụng. Trên thực tế, các ứng dụng được thiết kế dựa trên cơ sở một số nền tảng. Cho nên, điều quan trọng là nền tảng đáp ứng được cho người lập trình mô hình thực thi xác định, thông thường, dùng để tách người lập trình khỏi các khái niệm và những chi tiết công nghệ thấp hơn và các nền tảng. Như vậy, nền tảng của «1C:DOANH NGHIỆP» cho phép sử dụng các công nghệ khác nhau nhất ở mức thấp hơn mà không cần phải thay đổi mã nguồn của các giải pháp ứng dụng.

Ví dụ, nền tảng đáp ứng cho người lập trình mô hình làm việc riêng với dữ liệu và tách người lập trình khỏi các đặc tính của một kho chứa dữ liệu cụ thể, điều này cho phép, khi không cần thay đổi giải pháp ứng dụng, cũng có thể sử dụng trong đó các kho lưu dữ liệu khác nhau. Một ví dụ khác,

với tính chất là cơ sở dữ liệu, khi giải các bài toán có quy mô nhỏ ta có thể sử dụng động cơ quản lý cơ sở dữ liệu dạng tệp, còn để làm việc ở quy mô doanh nghiệp - sử dụng MS SQL Server.

Có lẽ việc đáp ứng đầy đủ các phương tiện để giải quyết các vấn đề đặt ra trước các giải pháp ứng dụng là tính chất căn bản của nền tảng «1C:DOANH NGHIỆP». Nó cho phép đáp ứng đầy đủ và tốt nhất các công nghệ và phương tiện cho người lập trình. Dẫu sao thì việc thường xuyên xuất hiện các “mối hàn” giữa các công nghệ lắp ghép khác nhau cũng sẽ là nguyên nhân của các vấn đề cần được quan tâm nghiêm túc, mà ví dụ đơn giản nhất là hệ thống các kiểu dữ liệu. Trong nền tảng của «1C:DOANH NGHIỆP» người lập trình sử dụng một hệ thống các kiểu dữ liệu để tương tác với cơ sở dữ liệu và cũng để triển khai lô-gíc nghiệp vụ, kể cả việc tạo dựng các giải pháp giao diện:



Chính vì vậy, người lập trình không bị nảy sinh các vấn đề có liên quan tới việc biến đổi các kiểu dữ liệu khi chuyển giữa các lớp khác nhau của hệ thống ứng dụng. Một yếu tố rất quan trọng khác, đó là việc chuẩn hóa. Việc tồn tại một nền tảng duy nhất đối với một số lượng lớn các giải pháp ứng dụng đã góp phần tạo nên một “lớp văn hóa” chung, bao gồm con người (lập trình viên, các nhà phân tích, người sử dụng) và phương pháp luận (các cấu trúc dữ liệu mẫu, thuật toán và các giao diện sử dụng). Dựa trên “lớp văn hóa” này, người lập trình không cần phải tốn nhiều công sức để tìm kiếm giải pháp cần thiết, tức quá trình từ khi người lập trình bắt đầu nhận dự án cho đến khi kết thúc bằng việc triển khai một giải pháp ứng dụng cụ thể của một phân hệ nào đó theo phương pháp luận mẫu.

Mức độ tùy ứng cao của các giải pháp ứng dụng theo yêu cầu của khách hàng là một trong số những lợi thế quan trọng được phân định rõ ràng giữa nền tảng và các giải pháp ứng dụng.

Cần hiểu rằng, đối với các bài toán kinh tế thì điều đặc biệt quan trọng là người lập trình phải biết cách đưa những thay đổi hợp lý vào giải pháp đóng gói (mà trong quá trình tạo dựng họ không được tham gia) để tùy chỉnh giải pháp theo yêu cầu cụ thể của doanh nghiệp. Khác với nhiều lĩnh vực khác, trong công nghệ lập trình luôn có những lập trình viên, họ không phải là những người viết chương trình “từ trang giấy trắng” mà chỉ chuyên triển khai và phát triển các giải pháp mẫu.

Những người lập trình này phải dựa vào sự hiểu biết của mình đối với giải pháp mà họ đang triển khai cho doanh nghiệp để xác định những yêu cầu đặc biệt và tiến hành ngay các tùy chỉnh hợp lý, công việc này cần được phát huy tối đa trong tất cả mọi cơ chế của nền tảng.

Các phương tiện của «1C:DOANH NGHIỆP» không phải là một “bộ công cụ” (“toolkit”) bổ sung nào đó mà là một phần không tách rời của nền tảng. Chúng được sử dụng như nhau trong việc tạo các giải pháp và để tùy ứng cho giải pháp của từng doanh nghiệp cụ thể. Các công cụ này đã được cài đặt sẵn trong bộ sản phẩm đóng gói của «1C:DOANH NGHIỆP» và được sử dụng để bổ sung những thay đổi nhỏ vào bộ sản phẩm đóng gói (ví dụ, bổ sung vào khuôn của biểu mẫu in), cũng như để thực hiện các chỉnh sửa quan trọng của giải pháp ứng dụng (bao gồm cấu trúc dữ liệu và lô-gíc nghiệp vụ). Các tính năng được sử dụng bổ sung hiệu quả các thay đổi vào ứng dụng khi triển khai nó đã được cài sẵn trong chính các công cụ đó, ngoài ra, còn được bổ sung thêm cả kiểu thiết kế để tạo giải pháp ứng dụng. Trong các phần tiếp theo của bài này chúng tôi sẽ cố gắng trình bày luận điểm này.

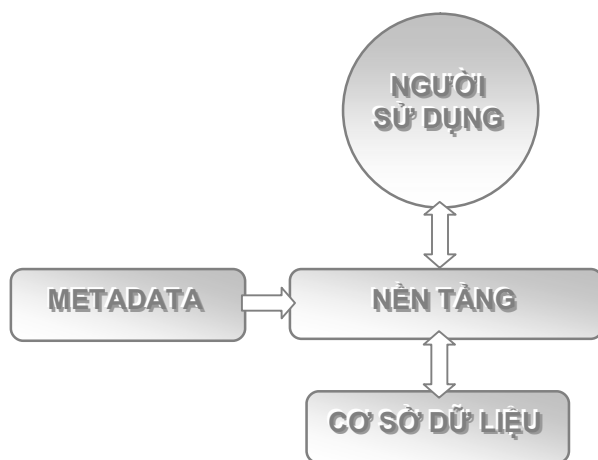
Việc tách giải pháp ứng dụng ra như một phần tử độc lập, thực tế, là đã giúp chúng tôi tổ chức hoàn chỉnh công nghệ xây dựng giải pháp, phổ biến và hỗ trợ các hệ thống ứng dụng đa dạng, để tập

trung nỗ lực của mình vào đặc thù của lớp bài toán này mà không yêu cầu có kiến thức sâu về công nghệ. Ví dụ, đối với một công ty mà có đội ngũ chuyên gia lành nghề trong một lĩnh vực cụ thể thì chỉ cần một khoảng thời gian ngắn cũng có thể đưa ra thị trường sản phẩm riêng của mình mà không phải mất nhiều năm để xây dựng cơ sở công nghệ.

Metadata, phương pháp mô tả giải pháp ứng dụng

Trong «1C:DOANH NGHIỆP», giải pháp ứng dụng không được viết trực tiếp bằng ngôn ngữ lập trình. Ngôn ngữ lập trình chỉ được sử dụng ở những nơi cần thiết nhất.

Trong nhân của giải pháp ứng dụng có metadata. Metadata là bản mô tả cấu trúc của giải pháp. Metadata tạo ra thứ bậc các đối tượng, từ các đối tượng này để tạo ra các thành phần của hệ thống ứng dụng, và chính các thành phần này xác định tất cả các tính chất của hệ thống. Thực tế, khi giải pháp ứng dụng làm việc thì nền tảng “tái tạo lại” (diễn lại) metadata để đáp ứng mọi chức năng cần thiết.



Tất cả các cấu trúc dữ liệu, thành phần các kiểu dữ liệu, những mối liên hệ giữa các đối tượng, các đặc điểm về thuộc tính của chúng, cách trình bày trực quan, hệ thống phân quyền truy cập, giao diện sử dụng... đều được mô tả bằng metadata. Trên thực tế, trong metadata không chỉ là nơi tập trung các thông tin về “cái gì chứa trong cơ sở dữ liệu” mà về cả “vì sao” cần phải lưu dạng thông tin này hay dạng thông tin khác, thông tin được lưu có vai trò gì trong hệ thống, và các mảng thông tin được liên hệ với nhau như thế nào. Việc sử dụng ngôn ngữ lập trình nói chung chỉ đem lại kết quả hạn chế, khi giải quyết các vấn đề có đòi hỏi mô tả thuật toán, ví dụ, tính thuế, kiểm tra độ chính xác khi nhập dữ liệu...

Phương pháp xây dựng giải pháp ứng dụng này sẽ cho ta những gì? Thứ nhất, khi mô tả metadata, ta có thể sử dụng rộng rãi cách soạn thảo trực quan. Điều này cho phép ráp nối phần lập trình quan trọng với phần được thiết kế bằng trực quan mà không phải viết tỉ mỉ các dòng mã lệnh. Thế nhưng, trong phương pháp này còn có các lợi thế khác cũng không kém phần quan trọng. Khi mô tả giải pháp ứng dụng trong các thuật ngữ metadata, người lập trình “thông báo” cho nền tảng rất nhiều thông tin bổ ích để sử dụng hiệu quả trong các mục đích khác nhau. Trên cơ sở metadata, hệ thống sẽ tự động “xây dựng” phần lớn các cơ chế và các đối tượng để đảm bảo hoạt động cho giải pháp ứng dụng. Ví dụ, mô tả metadata cho nền tảng để tự động tạo ra giao diện sử dụng của hệ thống, đảm bảo việc nhập và soạn thảo các thông tin có mối ràng buộc với nhau. Ví dụ, cung cấp tính năng giúp cho người sử dụng không có kỹ năng lập trình cũng có thể lập các báo cáo phức tạp.

Ý tưởng sử dụng metadata có thể được khái quát bằng luận điểm đơn giản nhất: “Hãy đừng lập trình tất cả các chức năng của giải pháp cần được khai thác. Hãy nói cho nền tảng biết về thành phần, cấu trúc, các đặc điểm và các mối ràng buộc giữa các phần khác nhau của nó, rồi để nền tảng tự giải quyết lấy”.

Ý tưởng này (Metadata Driven) ngày nay được ứng dụng rộng rãi trong công nghệ lập trình đầy triển vọng.

Xây dựng ứng dụng trên cơ sở mô hình

Ngay từ đầu trong «1C:DOANH NGHIỆP» đã có đặt định hướng xây dựng giải pháp ứng dụng trên cơ sở mô hình xác định.

Cách làm này hoàn toàn hợp lý, và theo đánh giá của chúng tôi, đó sẽ là phương pháp chiếm ưu thế trong tương lai của các công cụ lập trình hiện đại. Ý tưởng xây dựng giải pháp ứng dụng trên cơ sở mô hình, ví dụ, có thể tìm thấy hiện thân trong cấu trúc MDA (Model Driven Architecture) của cộng đồng OMG.

Có thể hiểu mô hình như là toàn bộ ý tưởng xây dựng giải pháp ứng dụng. Đó là các phương pháp xây dựng cấu trúc dữ liệu, các dạng liên hệ giữa dữ liệu, các nguyên tắc quản lý và sử dụng dữ liệu, các dạng mô tả lô-gíc nghiệp vụ, các phương pháp liên kết dữ liệu với các đối tượng giao diện, sự phân định chức năng theo các lớp của hệ thống và các vấn đề khác.

Điều quan trọng là các giải pháp ứng dụng phải tuân thủ theo mô hình đã được chọn và chúng phải được đáp ứng đồng nhất và phải tiên lượng được các tính chất của chúng. Trên thực tế, người lập trình mong muốn thể hiện trong giải pháp ứng dụng đặc thù của lĩnh vực ứng dụng cần thiết và có đủ các phương pháp để giải quyết vấn đề này bằng các công cụ có trong nền tảng. Một mặt, phương pháp này hạn chế (hoàn toàn có chủ ý) tự do của người lập trình, nhưng mặt khác, nó giúp họ tránh được nhiều lỗi khi lập trình, và cho phép họ chỉ trong thời gian ngắn có thể thu được giải pháp cần thiết, để sau đó giải pháp có thể được phát triển và hỗ trợ bằng chính mình hay nhờ sự trợ giúp của các chuyên gia khác, nếu thấy cần thiết.

Kết quả rõ ràng của phương pháp này là việc cách ly người lập trình giải pháp ứng dụng khỏi các chi tiết công nghệ lưu thông tin, tổ chức cấu trúc ba lớp... Ví dụ, như đã nói trên, tất cả các giải pháp ứng dụng được tạo dựa trên nền tảng của «1C:DOANH NGHIỆP» đều không cần phải bổ sung bất kỳ mọi thay đổi cũng có thể làm việc ngay với tệp dữ liệu riêng và Server dữ liệu. Lúc đó, các cấu trúc dữ liệu cần thiết đều được tạo ra và được thay đổi bằng hệ thống tự động trên cơ sở mô tả của metadata, và người lập trình không cần phải tìm hiểu chi tiết các định dạng lưu hệ thống quản lý cơ sở dữ liệu cụ thể. Việc xử lý dữ liệu trong ứng dụng cũng được mô tả ở mô hình bậc cao và được thực hiện tự động, có tính đến các đặc tính của kho chứa thông tin được sử dụng.

Theo đánh giá của chúng tôi, việc đảm bảo cho người lập trình mô hình xác định đã được trừu tượng hóa từ các công cụ sử dụng cụ thể, trong tương lai sẽ chiếm ưu thế trong các công cụ lập trình hiện đại.

Sự tồn tại một mô hình thống nhất đã tạo điều kiện tốt cho người lập trình dễ dàng khai thác hệ thống. Toàn bộ việc lập trình được tiến hành trong khuôn khổ một hệ thống khái niệm xuyên suốt và trong một không gian thống nhất của các kiểu dữ liệu. Người lập trình không cần phải hiểu thấu đáo cách trình bày của một số mô hình và không phải mất nhiều công sức để chuyển đổi chúng giữa các lớp khác nhau.

Ví dụ, bản mô tả trong các đối tượng (bản chất, thực thể) metadata này hay khác xác định được ngay cả các kiểu tương ứng của ngôn ngữ lập trình hệ thống và cả các cấu trúc dữ liệu cần thiết để lưu chúng. Tất cả các thủ thuật tiếp theo để xử lý các dữ liệu này trong bộ nhớ cũng như trong cơ sở dữ liệu đều được thực hiện như nhau mà không cần phải vượt bất kỳ một “rào cản” nào giữa các ký hiệu khác nhau, đã được chấp nhận khi làm việc với hệ thống quản lý cơ sở dữ liệu và với các ngôn ngữ lập trình tổng hợp.

Quản lý dữ liệu

Một trong số các khác biệt cơ bản của giải pháp ứng dụng so với các chương trình có chức năng khác là hệ thống biến hóa được sử dụng để xử lý dữ liệu trong chúng.

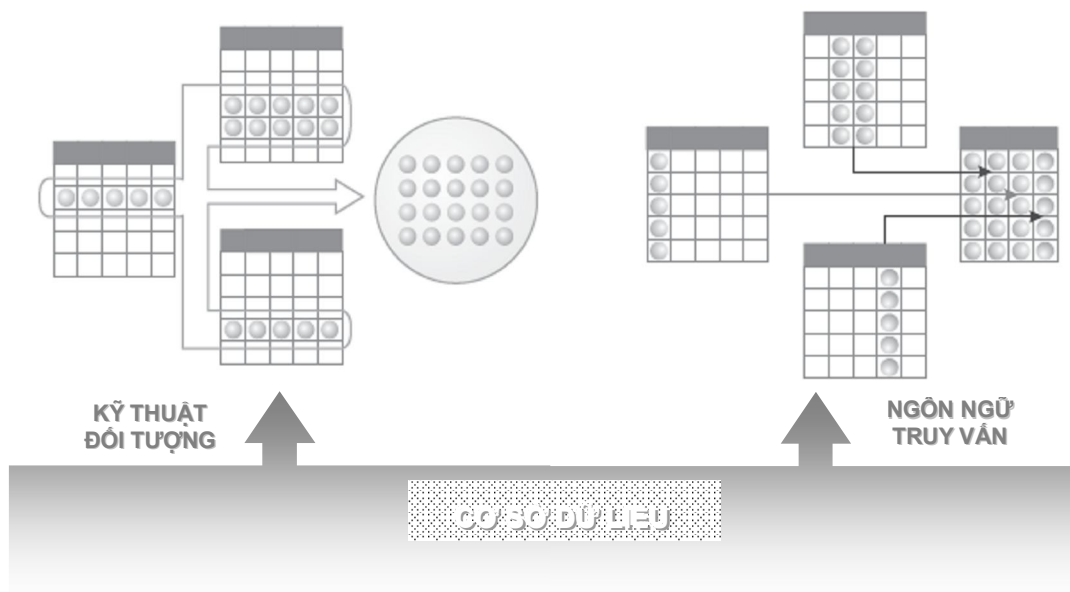
Như chúng ta được biết, đối với các bài toán quản lý thì làm việc với dữ liệu là phần nội dung chính, nhưng quan trọng hơn là vấn đề nghiên cứu để giải bài toán đó. Điều có thể lý giải, nếu trong nhiều lĩnh vực tạo dựng các hệ thống chương trình đã đạt được kết quả đến mức “gần như hoàn thiện” (ví dụ, trong bộ công cụ soạn thảo văn bản) thì trong lĩnh vực phần mềm quản lý kinh tế còn lâu mà đạt được như vậy.

Hệ thống các chương trình ứng dụng quản lý kinh tế thường xuyên bị các yêu cầu trái ngược “áp đảo”. Đó là do phải xử lý các khối lượng dữ liệu lớn nhưng đồng thời phải đáp ứng chức năng rộng và hiệu quả lao động cao. Khối lượng dữ liệu cần được xử lý tăng sẽ dẫn đến việc tăng nhu cầu giải các bài toán đa dạng, và những đòi hỏi về nâng cao và mở rộng quy mô. Đồng thời cũng không được quên về việc bảo đảm tiện lợi cho công tác lập trình, về quản lý hệ thống, về các khả năng cập nhật và chỉnh sửa chương trình... Trong thực tế của thế giới hiện đại, còn tồn tại nhiều hình thức quản lý và sử dụng dữ liệu, chúng được sử dụng trong các phương tiện tạo dựng các giải pháp ứng dụng trong các tổ hợp khác nhau. Và lại, còn xa mới đạt được lý tưởng - tất cả các công nghệ hiện có và đang được tạo ra đều là các giải pháp dung hòa, nhằm nâng cao chỉ tiêu của các hệ thống theo một số các tiêu chí đã được quy định.

Dĩ nhiên, các mâu thuẫn này có liên quan đến tất cả các khía cạnh của hệ thống, nhưng trong các phương pháp quản lý và xử lý dữ liệu thì chúng được thể hiện rõ ràng hơn.

Trong «1C:DOANH NGHIỆP» có sử dụng phương pháp hỗn hợp, phương pháp này có nhiều cách tiếp cận chung đã được đưa vào trong công tác lập trình của các hãng khác, nhưng vẫn giữ được các nét khác biệt căn bản của giải pháp.

Để thực hiện các thao tác chỉnh sửa dữ liệu (như tạo mới, sửa đổi, hủy bỏ), trong «1C:DOANH NGHIỆP» có sử dụng kỹ thuật đối tượng đặc biệt. Có nghĩa là người lập trình tương tác với cơ sở dữ liệu không phải ở lớp các nội dung mà là nhờ các đối tượng có bản chất tương ứng đang được lưu trong cơ sở dữ liệu. Để xử lý các dữ liệu này, người lập trình không cần phải viết các truy vấn phức tạp và biến đổi các kết quả xử lý dữ liệu thành các đối tượng của ngôn ngữ lập trình. Ta cần lấy ra đối tượng từ cơ sở dữ liệu, thay đổi các thuộc tính của nó và lưu trở lại. Khi đó người lập trình có khả năng tạo các bộ xử lý sự kiện có liên quan đến việc thay đổi dữ liệu, nhờ chúng để thực hiện các kiểm tra khác nhau và thay đổi dữ liệu khác khi thấy cần thiết. Hệ thống đảm bảo hỗ trợ hiệu quả công nghệ phương pháp đối tượng, ví dụ, như sắp xếp các đối tượng, kiểm tra tính toàn vẹn và các tham chiếu đối tượng... Để đọc dữ liệu, ta có thể sử dụng kỹ thuật đối tượng hay ngôn ngữ truy vấn hình thức – đây là ngôn ngữ được dựa trên ngôn ngữ lập trình cũ SQL, nhưng có một loạt các mở rộng quan trọng. Các mở rộng này, một mặt, nhằm hỗ trợ làm việc với các đối tượng được lưu trong cơ sở dữ liệu, mặt khác, để giải quyết thành công các nhiệm vụ quản lý kinh tế. Sau đây chúng ta sẽ xem xét chi tiết hơn về ngôn ngữ truy vấn:



Một trong số các lợi thế căn bản của luận thuyết quản lý dữ liệu theo đối tượng quan hệ là tính rành mạch và đơn giản khi xây dựng ứng dụng. Kỹ thuật đối tượng được sử dụng chính là để xử lý dữ liệu, để đảm bảo đọc ra các thuật toán lô-gíc nghiệp vụ, để giảm lỗi cơ bản khi lập trình, cả để đáp ứng an toàn ở mức độ cao cho tính toàn vẹn dữ liệu. Ngoài ra, còn một đặc tính quan trọng của kỹ thuật đối tượng là đơn giản hóa việc chuyển đổi sang hệ thống phân tán và tích hợp.

Về phía mình, việc mô tả hình thức các truy vấn cho phép lựa chọn tùy ý dữ liệu và đảm bảo đủ khả năng trừu tượng hóa cần thiết để giải quyết các nhiệm vụ phân tích.

Theo chúng tôi, kỹ thuật hỗn hợp này dù ở dạng nào cũng sẽ là sự khởi đầu cho việc sử dụng rộng rãi và đầy triển vọng của công tác lập trình cho hệ thống quản lý kinh tế.

Xin được nói sớm một chút là trong mô hình «1C:DOANH NGHIỆP» có thực thi mô hình ý tưởng hiện đại để làm việc với thông tin, trong đó kết hợp ba phương pháp trình bày dữ liệu. Đó là lưu các thực thể trong cơ sở dữ liệu, trình bày chúng trong ngôn ngữ lập trình ở dạng đối tượng, và thể hiện trong định dạng XML. Thực tế, bất kỳ mọi thông tin đều có thể được trình bày bằng một trong ba phương pháp trên, tùy thuộc vào chế độ làm việc hiện tại.

Việc lưu lâu dài các bản chất hay thực thể (persistence) được thực hiện trong cơ sở dữ liệu, đảm bảo được độ tin cậy và xử lý hiệu quả khối lượng lớn thông tin. Để đưa vào các thay đổi, trước hết dữ liệu phải được chuyển thành các đối tượng trong ngôn ngữ hệ thống. Khi trao đổi dữ liệu trong cơ sở dữ liệu, chúng ta chỉ tương tác với các đối tượng thông tin khác, dữ liệu được lưu trữ vào thành định dạng XML.



Cần lưu ý rằng, phương pháp trình bày dữ liệu trong cơ sở dữ liệu được dựa trên các đối tượng, và các đối tượng này có thể được chuyển đổi dữ liệu thành định dạng XML. Các nhiệm vụ phân tích và lập trình không cần phải sử dụng các đối tượng để chuyển đổi dữ liệu thành định dạng XML, và các nhiệm vụ phân tích này sẽ được thực hiện bằng các phương pháp khác.

Cách thức này cho phép «1C:DOANH NGHIỆP» hòa nhập cùng với các giải pháp không đồng nhất trong tương lai, theo đánh giá của chúng tôi (cũng giống với đánh giá của các chuyên gia cao cấp) thì ngày càng được áp dụng rộng rãi.

Một trong số những đặc tính khác biệt của mô hình «1C:DOANH NGHIỆP» mà không có các điểm tương tự trong các hệ thống tương đương khác, đó là việc phân chia tất cả các dữ liệu sử dụng ra thành thực thể đối tượng và những đối tượng không có các bản chất đó. Cần nhớ rằng, để quản lý bất kỳ kiểu dữ liệu này hay kiểu dữ liệu khác ta cần sử dụng kỹ thuật đối tượng.

Việc phân chia này phù hợp với bản chất thực của dữ liệu. Trong lĩnh vực ứng dụng luôn luôn tồn tại các thực thể có tính chất đối tượng, ví dụ, “Khách hàng”, “Pháp nhân”, “Hàng hóa”. Ở đây bản thân đối tượng được xác định ý nghĩa đầy đủ mà không phụ thuộc vào dữ liệu mô tả nó. Ví dụ, ở một người có thể thay đổi họ, tên, số chứng minh nhân dân, nhưng điều quan trọng mà chúng ta biết thì đây chính là cá nhân đó (đối tượng duy nhất), mặc dù họ tên đã được thay đổi. Mặt khác, cũng có những thực thể không có tính chất đối tượng. Ví dụ, khi ghi nhận việc tiếp nhận một loại hàng hóa nào đó vào kho, đây hoàn toàn là thông tin về sự luân chuyển hàng hóa, ngoài việc nó được ghi nhận vào nhật ký (biểu ghi) thì không có nội dung nào khác. Nhưng nếu ta thay thế loại hàng hóa trong biểu ghi này bằng một loại hàng hóa khác thì ý nghĩa biểu ghi luân chuyển hàng hóa sẽ hoàn toàn thay đổi. Nói cách khác, đối với một bản chất mà nội dung không chỉ ra giá trị cụ thể cho các trường thì không có ý nghĩa.

Nhờ sự phân định bản chất, nền tảng «1C:DOANH NGHIỆP» đã đáp ứng sẵn cho người lập trình phương pháp luận để dễ dàng giải quyết các bài toán xử lý dữ liệu phù hợp với thuộc tính của các bản chất này.

Làm việc với các thực thể đối tượng được hỗ trợ bằng cách trình bày (thể hiện) bản chất của cơ sở dữ liệu trong kiểu các đối tượng ngôn ngữ lập trình hệ thống, và bởi các kiểu dữ liệu chuyên dùng

để trình bày các tham chiếu đối tượng (các tham chiếu tới đối tượng cơ sở dữ liệu). Lúc này, khi đã biết đối tượng thì dễ dàng nhận lại được tham chiếu của nó, còn khi biết được tham chiếu thì ta có thể lấy ra được đối tượng từ cơ sở dữ liệu. Kỹ thuật này đảm bảo cho ta có phương pháp đơn giản mô tả các thuật toán lô-gíc nghiệp vụ trong mã nguồn, để quản lý các đối tượng; ngoài ra, nó còn đảm bảo sự toàn vẹn lô-gíc cho dữ liệu, khi thực hiện bất kỳ mọi thao tác. Kỹ thuật này nhắc lại cách viết ứng dụng của các cơ sở dữ liệu đối tượng, chỉ có khác là, việc lưu dữ liệu được tiến hành trong các bảng có liên quan tới hệ thống quản lý cơ sở dữ liệu. Lúc đó, trong các mô-đun đã được viết trên ngôn ngữ hệ thống có thể cùng tồn tại một số đối tượng để trình bày một thực thể của cơ sở dữ liệu.

Các cơ chế của nền tảng đảm bảo việc hỗ trợ tên đơn trị (ID) của các đối tượng (tham chiếu), đảm bảo việc kiểm tra các phiên bản đối tượng, đảm bảo duy trì cả việc phong tỏa tạm thời hệ thống trong trường hợp cần thiết. Việc phong tỏa tối ưu nhằm đảm bảo cho sự toàn vẹn lô-gíc khi thay đổi đối tượng, còn việc phong tỏa không tối ưu cho phép người sử dụng cùng đồng thời tiến hành việc soạn thảo một loại đối tượng trong giao diện của «1C:DOANH NGHIỆP». Nền tảng thực hiện tối ưu hóa các thao tác đọc đối tượng nhờ sử dụng cơ chế lưu giữ chúng bên trong cũng như bên ngoài chúng. Khi xử lý việc chỉnh sửa các đối tượng, trong hệ thống có thực thi kỹ thuật “ghi chép thông minh”: hệ thống sẽ kiểm soát các thay đổi của chúng và ghi vào đĩa cứng chỉ những dữ liệu đã được xử lý mà vẫn bảo đảm được sự toàn vẹn của thao tác này.

Làm việc với các thực thể không mang tính chất đối tượng được thực hiện nhờ bộ bản ghi, về thực chất thì chúng là tập hợp bản ghi. Các bộ bản ghi này được hỗ trợ việc đọc và xử lý dữ liệu theo yêu cầu chi tiết đối với lô-gíc ứng dụng.

Điều quan trọng là cả hai phương pháp nêu trên đều đáp ứng thành công việc xây dựng thuật toán lô-gíc nghiệp vụ. Khi đọc thuật toán, người lập trình cần xử lý lô-gíc nghiệp vụ hiện có trong ứng dụng có thể hiểu nhanh bản chất vấn đề và tiến hành gỡ rối, chỉnh sửa mã nguồn. Bây giờ, để tương tác với cơ sở dữ liệu, người lập trình không cần phải lập phức tạp trình biến đổi thông tin từ các cấu trúc đối tượng thành các thông tin quan hệ, cho phép nâng cao hiệu quả tạo và gỡ rối giải pháp phần mềm.

Có một đặc tính quan trọng của kỹ thuật đối tượng nữa trong nền tảng của «1C:DOANH NGHIỆP», đó là những đối tượng có mặt trong các mô-đun của ngôn ngữ hệ thống (đối với các thực thể là đối tượng và cả các thực thể không mang tính đối tượng), chúng được sử dụng cả để phản ánh dữ liệu trong giao diện. Các phần tử điều khiển hộp thoại được liên hệ trực tiếp với các đối tượng cần thiết, người sử dụng có thể cho hiển thị và soạn thảo chúng mà không cần đến sự hỗ trợ của người lập trình.

Đối với các thực thể mang tính đối tượng, nền tảng «1C:DOANH NGHIỆP» có hỗ trợ cơ chế trình bày. Cơ chế này đáp ứng việc hiển thị trong giao diện các giá trị nhận được bằng cách tham chiếu tới các thực thể của cơ sở dữ liệu. Khi cần hiển thị giá trị tham chiếu, hệ thống sẽ tự động tạo cách trình bày trên cơ sở các thuộc tính của metadata, và ta có thể nhận được thông tin từ cơ sở dữ liệu nhờ kho chứa hay các cơ chế khác. Trong quá trình xử lý các truy vấn và xây dựng các báo cáo cũng sử dụng cách trình bày. Bằng phương pháp tổng hợp cho phép ta nhận được các cách trình bày các trường tham chiếu, nếu truy vấn được tạo để hiển thị dữ liệu trong giao diện sử dụng, và đưa chế độ tự động hiển thị cách trình bày vào trong các báo cáo đối với những trường chứa các giá trị tham chiếu. Điều quan trọng là cơ chế trình bày đảm bảo khả năng cho người lập trình dễ dàng xử lý và sử dụng các tham chiếu đối tượng, nếu ta đồng thời sử dụng một số lệnh gọi đến cơ sở dữ liệu.

Cùng với các phương pháp quản lý dữ liệu và lập truy vấn đã được mô tả ở phần trên, hệ thống còn đáp ứng một phương pháp truy cập dữ liệu khác, đó là phương pháp chọn động. Cơ chế này cho phép truy cập và chọn được khối lượng thông tin rất lớn mà vẫn bảo đảm cho người sử dụng đọc được các mục thông tin đó. Khi đó người lập trình chỉ việc chỉ ra những dữ liệu nào họ cần, còn hệ thống sẽ tự động thực hiện truy cập các thông tin cần thiết. Để làm việc này người sử dụng không cần sử dụng bất kỳ mọi phương tiện chuyên dụng đọc động cụ thể, mà hệ thống sẽ tự động định dạng các truy vấn để chọn lần lượt các khối bản ghi.

Còn một giải pháp quan trọng nữa trong phần làm việc với dữ liệu trong «1C:DOANH NGHIỆP», đó là sự hỗ trợ bằng các kiểu dữ liệu phức hợp trong các trường. Khả năng này, như chúng ta biết, không giống như trong các hệ thống khác. Khi mô tả kiểu trường của một đối tượng nào đó, ta

không chỉ có thể chọn một trong số các kiểu dữ liệu hợp lý mà trên thực tế có thể chọn bất kỳ mọi tổ hợp của chúng. Ví dụ, trong trường “Người trả tiền” của chúng từ để phản ánh các giao dịch với ngân hàng được phép lưu các tham chiếu tới đơn vị pháp nhân và cá nhân, tùy thuộc vào từng giao dịch cụ thể. Mặc dù đây là một ví dụ đơn giản, khả năng làm việc với các kiểu phức hợp cho phép giải các bài toán này, ví dụ như lưu lại các đặc trưng tùy ý của hàng hóa, tiến hành kế toán phân tích trên các tài khoản kế toán theo thành phần bất kỳ của phạm vi kế toán phân tích đã được thiết lập bởi người sử dụng... Điều quan trọng là hệ thống không chỉ đáp ứng được khả năng lưu các giá trị khác loại trong một trường mà còn giúp người lập trình hiểu rõ phương pháp hơn. Trước hết, cần đánh dấu sự hỗ trợ công việc toàn phần của “nhân” cơ sở dữ liệu và của ngôn ngữ truy vấn với các trường có kiểu phức hợp. Các trường này đều được hỗ trợ bộ các thao tác chuẩn (so sánh, mô phỏng...). Một vấn đề quan trọng khác là sự hỗ trợ của các kiểu phức hợp trong các cơ chế giao diện của hệ thống. Ví dụ, trường nhập liệu có liên quan với dữ liệu của kiểu phức hợp loại này có thể bảo đảm các khả năng soạn thảo (chọn kiểu; soạn các giá trị của tất cả các kiểu để đưa vào trường đã nói tới trên; hạn chế các kiểu cần chọn).

Cũng cần lưu ý đến các công cụ lập truy vấn của «1C:DOANH NGHIỆP». Như chúng tôi đã nói ở trên, công cụ này được tạo ra trên cơ sở của các cấu trúc ngôn ngữ chuẩn của QSL, thế nhưng cũng có một loạt các mở rộng quan trọng.

Trước hết lưu ý là trong ngôn ngữ truy vấn có hỗ trợ công việc với các đối tượng đã được lưu trong cơ sở dữ liệu. Tất cả các toán tử của ngôn ngữ truy vấn bảo đảm công việc với các kiểu tham chiếu (với các trường lưu các tham chiếu lên các đối tượng của cơ sở dữ liệu). Ví dụ, hỗ trợ việc gọi các trường theo ký hiệu “dấu chấm” mà không bị hạn chế số các lớp. Có thể chỉ ra trường được chọn trong truy vấn, ví dụ, “HàngHóa.NhàSảnXuất.Tên nước”. Đối với các đối tượng của cơ sở dữ liệu, cho phép gọi ra các bảng lồng trong, giống như là cách gọi các bảng riêng hoặc giống như là khi gọi các trường thông thường của đối tượng có chứa tập hợp bản ghi.

Một đặc trưng quan trọng nữa là trong nền tảng có thực thi tính năng tạo lập các tổng kết nhiều chiều với trình tự quét các chiều đo tùy ý. Khi đó, có thể sử dụng khả năng như kết hợp quét nhiều chiều và nhiều lớp theo phân nhánh giá trị của từng chiều đo (ví dụ, cấu trúc nhiều lớp của các bộ phận, hoặc nhiều lớp gom nhóm hàng hóa). Ngoài ra còn hỗ trợ một loạt các chế độ đặc biệt để xử lý các kết quả tổng kết.

Cơ chế truy vấn có khả năng tự động sắp xếp các tập mẫu dữ liệu trên cơ sở các thuộc tính của các đối tượng ứng dụng đã được xác định trong metadata. Như vậy sẽ cho phép người lập trình trong khi tạo truy vấn có thể nhận báo cáo hoặc để đọc dữ liệu, không cần chỉ ra theo những trường nào cần thực hiện việc sắp xếp mà chỉ cần bật chế độ tự động để chấp nhận việc sắp xếp chuẩn đối với các dữ liệu đã được chọn.

Còn một công cụ mạnh của cơ chế truy vấn nữa, đó là các bảng ảo. Các bảng ảo bảo đảm truy cập dễ dàng tới dữ liệu bất kỳ hiện có trong các phân hệ ứng dụng khác nhau mà không cần phải lập các truy vấn phức tạp. Ví dụ, có thể gọi bảng ảo để nhận lại số dư và phát sinh hàng hóa theo các kho và hàng hóa theo từng kỳ. Cần nhớ rằng, làm việc với các bảng ảo không giống như khi lưu bình thường các truy vấn mẫu (view). Khi sử dụng các bảng ảo người lập trình phải chỉ ra bộ các tham số để mô tả lựa chọn cần thiết, đó không chỉ là các giá trị cụ thể mà còn có thể là các điều kiện phức tạp. Người tạo giải pháp ứng dụng làm việc với bảng ảo, thực tế, cũng như với các bảng bình thường, nhưng hệ thống sẽ giúp định dạng truy vấn cho cơ sở dữ liệu để đạt được hiệu quả cao nhất. Chẳng hạn, khi gọi dữ liệu cần được xử lý bằng các cơ chế kế toán ta có thể sử dụng các kết quả trung gian đã được lưu bởi chúng.

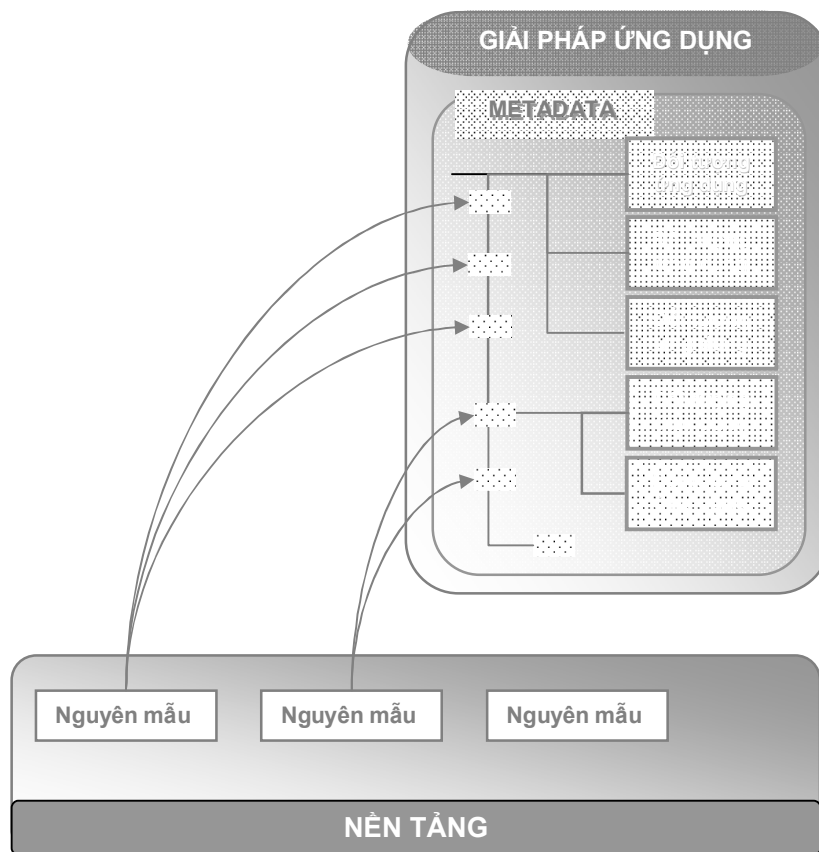
Nguyên mẫu chuẩn của các đối tượng ứng dụng

Nếu nói về sự khác nhau của các mô hình của giải pháp ứng dụng và của các công cụ tạo dựng chúng thì có lẽ, điều quan trọng hơn hết là phải biết, ứng dụng được mô tả trong những khái niệm nào (thậm chí có thể nói “trong luận thuyết nào”). Dĩ nhiên, trong mỗi công cụ tạo dựng đều có thể sử dụng nhiều phương pháp mô tả, nhưng chỉ một trong số các bộ khái niệm được đóng vai trò chủ đạo.

Để làm ví dụ, có thể đưa ra việc mô tả trong các thuật ngữ của các bảng quan hệ, các lớp trong ngôn ngữ lập trình hướng đối tượng, các thực thể được lưu (Entity Persistent)...

Trong mô hình của «1C:DOANH NGHIỆP» có sử dụng phương pháp không giống như trong các hệ thống khác. Ở đây, toàn bộ giải pháp ứng dụng được mô tả bằng metadata dưới dạng tổng hợp của các đối tượng ứng dụng chọn ra từ bộ các nguyên mẫu (các lớp) được xác định chặt chẽ. Có thể đặt tên các đối tượng đã được tạo ra là các cấu phần ứng dụng, còn các nguyên mẫu của chúng là các khuôn mẫu (patterns). Mỗi nguyên mẫu này đảm nhiệm việc phản ánh một tổ hợp các đối tượng hoặc quy trình trong lĩnh vực ứng dụng mà có các đặc tính và có vai trò giống nhau trong khung giải pháp.

Những ví dụ của các nguyên mẫu trong «1C:DOANH NGHIỆP» là: “Danh mục”, “Chứng từ”, “Biểu ghi tích lũy”... Mỗi nguyên mẫu đều có cơ sở thực thi nào đó để mà xác định đặc điểm tính năng của các đối tượng được tạo ra trên cơ sở của nguyên mẫu này: cấu trúc của thực thể cần lưu cùng với một số trường định trước, tập hợp kiểu dữ liệu của ngôn ngữ lập trình, phương thức, thuộc tính và sự kiện, kể cả các thủ thuật chuẩn cho bài toán cần giải quyết, cách hiển thị và soạn thảo, các phương pháp tùy chỉnh phân quyền truy cập của người sử dụng...



Nếu trong quá trình lập trình mà cần tạo đối tượng để thể hiện các đặc tính của lĩnh vực ứng dụng thì người lập trình sẽ chọn nguyên mẫu phù hợp, và trên cơ sở nguyên mẫu đó tạo ra đối tượng metadata. Tiếp đến, người lập trình có thể đưa thêm các thuộc tính khác nhau để xác định các đặc tính của đối tượng đã được tạo trên cơ sở nguyên mẫu này, có thể bổ sung cấu trúc dữ liệu của đối tượng khi thấy cần thiết, triển khai bộ các phương thức cần thiết, xác định các hàm xử lý sự kiện, làm chi tiết phương pháp hiển thị đối tượng trong giao diện...

Đối với các đối tượng đơn giản thì hoàn toàn có thể không cần phải bổ sung thêm như vậy, và toàn bộ thủ tục tạo dựng đối tượng sẽ chỉ còn là xác định tên cho nó trong các thuộc tính của metadata. Trong những trường hợp phức tạp hơn, người lập trình cũng có khả năng xác định chi tiết cho cấu trúc dữ liệu, các thuật toán lô-gíc nghiệp vụ, nhưng khi đó chức năng chuẩn sẽ được thực thi tự động mà không cần đến sự trợ giúp từ bên ngoài.

Điều quan trọng là số lượng của các nguyên mẫu này trong nền tảng không nhiều, khoảng vài chục và có thể dễ dàng tìm hiểu để sau đó áp dụng hiệu quả cho việc giải quyết các nhiệm vụ của lĩnh vực ứng dụng.

Như vậy, cấu trúc metadata của «1C:DOANH NGHIỆP» không chỉ đơn giản là bộ mô tả của các đối tượng trong các thuật ngữ thống nhất. Toàn bộ giải pháp ứng dụng, trên thực tế, bao gồm các đối tượng được phân chia rõ ràng theo các vai trò trong giải pháp ứng dụng. Phương pháp này làm tăng đáng kể hiệu quả làm việc: trong việc mô tả hệ thống trong các giới hạn của metadata, và trong cả việc xây dựng ứng dụng trên cơ sở mô hình.

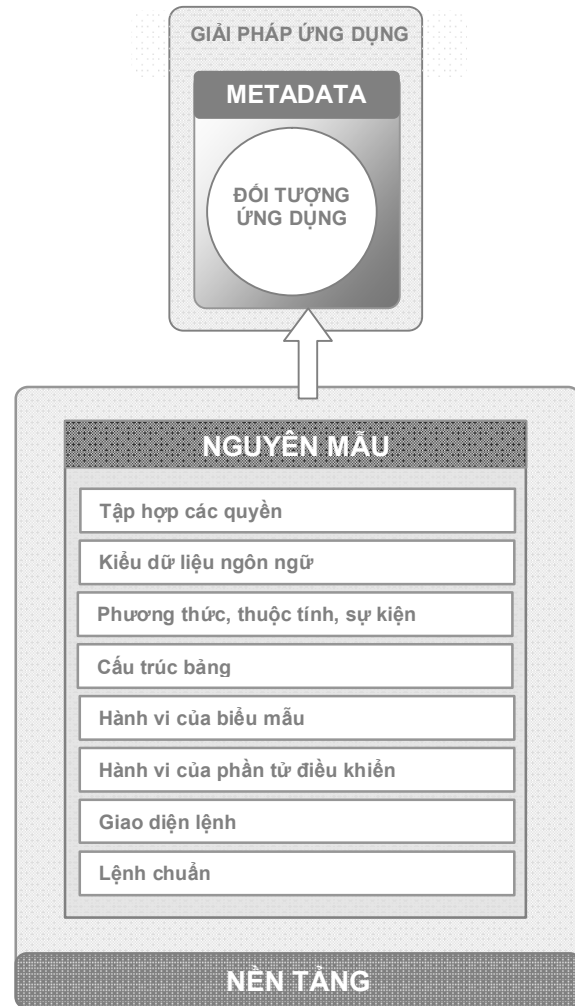
Nhờ các đối tượng đã được mô tả bởi metadata và nhờ chức năng chuẩn và vai trò định trước mà hệ thống có thể tự động giải được nhiều lớp nhiệm vụ giành trực tiếp cho các đối tượng này cũng như đối với các cơ chế chung để làm việc với chúng. Thực tế, khi biết được chức năng (vai trò trong giải pháp ứng dụng) của đối tượng này hay đối tượng khác thì không cần hỗ trợ của người lập trình, bản thân nền tảng có thể đáp ứng “phương pháp riêng” phù hợp đối với nó trong mọi tình huống. Ví dụ, cơ chế các hộp thoại “biết” cần phải làm thế nào để soạn thảo tốt hơn đối với dữ liệu này hay dữ liệu khác, còn cơ chế chuẩn xây dựng các báo cáo có thể tự “xây dựng” để tiến hành phân tích thông tin một cách hiệu quả nhất.

Việc sử dụng các nguyên mẫu định trước đảm bảo cho việc chuẩn hóa mô hình tạo dựng giải pháp ứng dụng. Việc chuẩn hóa này có thể không thực sự quan trọng đối với một ứng dụng được làm để sử dụng riêng, nhưng nó có ý nghĩa to lớn khi tạo công nghiệp lập trình và hỗ trợ cho nhiều lớp giải pháp ứng dụng. Việc chuẩn hóa cho phép giảm nhẹ đáng kể bài toán thiết kế, giảm độ phức tạp được nhiều lần và giảm chi phí cho việc bảo trì/bảo hành sản phẩm. Ví dụ, nếu cần bổ sung thêm một chuyên gia vào đề án hay cần phải chuyển một cộng tác viên sang làm công việc hỗ trợ hay phát triển giải pháp thì người lập trình mới sau khi đã xem qua cấu trúc của metadata, họ có thể trình bày tổng quát cách xây dựng về cơ chế hoạt động chính của giải pháp ứng dụng, bởi vì trong cấu trúc của metadata tất cả các đối tượng ứng dụng đều được nhóm theo các vai trò chức năng trong mô hình của «1C:DOANH NGHIỆP». Sau đó người lập trình sẽ có thể nhanh chóng hiểu rõ cấu trúc chung và lô-gíc nghiệp vụ của từng phân hệ. Theo kinh nghiệm thì các chỉ tiêu này hoàn toàn khác biệt so với những gì đã đạt được khi sử dụng các phương pháp khác để mô tả ứng dụng.

Các đối tượng và cơ chế ứng dụng

Ở phần trên, chúng ta chỉ mới mô tả nguyên tắc xây dựng giải pháp ứng dụng dựa trên cơ sở sử dụng các nguyên mẫu chuẩn của các đối tượng ứng dụng.

Đĩ nhiên, bài viết này không phải là nơi để mô tả chi tiết các nguyên mẫu hiện có. Nhưng để minh họa các ý tưởng đã được đưa vào mô hình hiện có phục vụ xây dựng các giải pháp ứng dụng, sau đây chúng tôi sẽ trình bày tổng quan về các nguyên mẫu và các khả năng được đáp ứng bởi chính chúng.



Trước hết chúng ta hãy xem một số các ví dụ nguyên mẫu đơn giản, đó là các danh mục và chứng từ. Các danh mục dùng để mô tả các thư mục mà có nội dung thường ít thay đổi. Đó có thể là danh sách các mặt hàng xuất xưởng, danh sách khách hàng của công ty, danh sách ngoại tệ... Danh mục đảm bảo việc hỗ trợ cấu trúc phân nhánh, cho phép cung cấp dữ liệu theo từng đối tượng riêng biệt và theo các nhóm, đồng thời cung cấp một loạt các tính năng tiện ích khác.

Chứng từ dùng để phản ánh trong hệ thống các sự kiện xảy ra trong đời sống của doanh nghiệp: nhập hàng, chi trả tiền gửi, tiếp nhận người lao động vào làm việc... Nguyên mẫu này đảm bảo phản ánh chúng trong các cơ chế kế toán khác nhau, duy trì việc kiểm tra trình tự xử lý các sự kiện, thực hiện việc đánh số thứ tự xuyên suốt các đối tượng của các dạng khác nhau...

Chúng ta sẽ dành thời gian nhiều hơn để xem xét một số khả năng của hai nguyên mẫu này (danh mục và chứng từ).

Như ta đã biết, các mô hình quan hệ cổ điển không có sẵn các phương tiện để hỗ trợ việc phân nhánh dữ liệu và việc thực thi luôn luôn đòi hỏi nhiều công sức. Danh mục (và nhiều nguyên mẫu khác của các đối tượng ứng dụng của «1C:DOANH NGHIỆP») có hỗ trợ phân nhánh nhiều lớp ngay từ đầu. Cơ chế này được bật lên một cách đơn giản bằng cách kích hoạt thuộc tính tương ứng trong metadata. Khi đó, việc hỗ trợ phân nhánh được mở rộng ngay đến tất cả mọi phương diện sử dụng của đối tượng ứng dụng. Ví dụ, nguyên mẫu đảm bảo hỗ trợ các thuộc tính cần thiết và các phương pháp trong mô hình đối tượng quản lý dữ liệu (xác định lớp đối tượng, kiểm tra vòng lặp phân nhánh...). Trong các cơ chế giao diện có thực thi việc trình bày dữ liệu dưới dạng danh sách phân nhánh hoặc hình cây có khả năng di chuyển theo các nhánh cũng như thay đổi cấu trúc phân nhánh một cách trực tác. Trong các cơ chế báo cáo có đáp ứng việc lập ra các chứng từ phân nhánh và nhận lại các kết quả tổng kết phân nhánh nhiều tầng mà trong đó đối tượng kiểu này được dùng để làm tham số gom nhóm.

Một ví dụ khác về khả năng của các nguyên mẫu, đó là cơ chế kết chuyển chứng từ. Cơ chế này đảm bảo cho người lập trình một mô hình chuẩn để tổ chức mối liên hệ giữa thông tin về các sự kiện xảy ra trong doanh nghiệp với các cơ chế kế toán khác nhau. Bất kỳ mọi thông tin do người sử dụng nhập vào dưới dạng chứng từ đều được phản ánh trong mọi cơ chế kế toán (như việc lập kế hoạch, kế toán quản trị, kế toán doanh nghiệp...). Người lập trình chỉ cần chỉ ra trong các thuộc tính của metadata mối liên hệ giữa các chứng từ và các cơ chế kế toán, kể cả mô tả thuật toán kết chuyển chứng từ. Tất cả mọi thao tác cần thiết để kết chuyển và hủy kết chuyển đều do hệ thống tự động thực hiện. Khi đó, hệ thống sẽ đáp ứng các tính năng bổ sung như: hỗ trợ phản ánh các sự kiện trong thời gian thực, hỗ trợ việc khôi phục trình tự phản ánh các sự kiện xảy ra trong doanh nghiệp, khi thay đổi hồi tổ chứng từ... Kết quả là đưa một mô hình thống nhất, liên hệ với dữ liệu gốc và các cơ chế kế toán; mô hình này không chỉ đơn giản là làm giảm nhẹ lao động cho người lập trình, mà còn đảm bảo tính chất thống nhất cho mọi giải pháp ứng dụng, và giảm nhẹ đáng kể cho việc khai thác và bảo trì hệ thống.

Điều thú vị là các nguyên mẫu đối tượng của «1C:DOANH NGHIỆP» đã được áp dụng vào cơ chế tính toán định kỳ phức tạp. Cơ chế này có thể xem như một công cụ tổng hợp để giải quyết vấn đề tính toán có độ phức tạp bất kỳ. Thực tế thì cơ chế này thường được áp dụng cho việc tính lương, nhưng nhiệm vụ của cơ chế này không phải để tính lương mà chính là để giải quyết các bài toán khác đòi hỏi phải mô tả tính toán định kỳ phức tạp. Ví dụ, tính cổ tức, tính giá dịch vụ công cộng... Cơ chế có chứa bộ các chiến lược mẫu, ví dụ, đó là cơ chế loại trừ, được dùng để thay thế một số dạng tính toán bằng một số các dạng tính toán khác khi thời gian sự kiện của chúng giao nhau. Khi áp dụng tính lương, tính năng này giải quyết được mọi vấn đề có liên quan với việc tính lương và khấu trừ trong đó có các quy tắc loại trừ lẫn nhau phức tạp. Một ví dụ khác, đó là khả năng thiết lập mối ràng buộc giữa các dạng tính lương và khấu trừ khác nhau trong kỳ hoạt động. Như vậy, cơ chế là một mô hình toán học để xác định mối liên hệ giữa các dạng tính toán khác nhau và đảm bảo cho người lập trình các cơ chế thực sự để xử lý thông tin trên cơ sở của mô hình này.

Chúng ta sẽ không đi sâu chi tiết về cơ chế này nữa mà có thể nhấn mạnh rằng: cơ chế hoàn toàn cho phép mô tả sơ đồ tính toán trên cơ sở nguyên mẫu của các đối tượng sử dụng và các tính năng sẵn có của chúng. Trong ví dụ của cơ chế này, có thể minh họa hiệu quả sử dụng phương pháp đã được đề xuất. Cho đến thời điểm này, chúng tôi không có bất kỳ thông tin nào về sự tồn tại một công cụ lập trình phần mềm quản lý kinh tế tương tự như cơ chế này. Như chúng ta biết, mô-đun Tiền lương (Payroll) có giá khá cao cho dịch vụ thiết kế và triển khai. Ngoài ra mô-đun này về căn

bản phải viết lại hoặc một phần, hoặc phải viết lại hoàn toàn để phù hợp với từng quốc gia, bởi vì các cơ chế bao giờ cũng có liên quan đến luật pháp của nước sở tại. Việc áp dụng các cơ chế tính toán định kỳ phức tạp của «1C:DOANH NGHIỆP» cho phép tạo ra mô-đun này, bắt đầu từ khâu thiết kế từ “trang giấy trắng” cho đến khi khai thác theo sơ đồ chuẩn. Theo đánh giá của chúng tôi, cơ chế đã giúp giảm nhẹ chi phí và thời gian thiết kế, nhưng cái chính là làm giảm đáng kể việc bổ sung các thay đổi trong quá trình hỗ trợ hệ thống.

Có một ví dụ đặc trưng nữa, đó là việc sử dụng cơ chế của các dạng đặc tính để giải các bài toán mà rất khó hòa hợp được với mô hình cơ sở dữ liệu cổ điển. Một trong số các bài toán đó là tạo ra các cấu trúc để phản ánh các tính chất (đặc tính) khác nhau của hàng hóa hay sản phẩm. Nếu tạo ra giải pháp dành cho một doanh nghiệp có định hướng rõ ràng về nhóm hàng hóa xác định thì tất cả các thông tin cần thiết có thể trực tiếp đưa vào trong cấu trúc của danh mục hàng hóa. Ví dụ, đối với một công ty có tên tuổi chuyên bán ô tô thì các thông tin chi tiết là màu sắc, dung tích động cơ... Thế nhưng, phương án đó sẽ không phù hợp nếu chúng ta phát triển một sản phẩm đóng gói hay một giải pháp dành cho một doanh nghiệp sản xuất hay bán các loại hàng hóa khác nhau, bởi vì rõ ràng là thành phần các đặc tính luôn phụ thuộc vào nhóm hàng hóa cụ thể, và việc bổ sung một nhóm mới cũng đồng nghĩa với việc chỉnh sửa giải pháp ứng dụng. Rõ ràng rằng, việc triển khai giải pháp như vậy rất khó trên cơ sở cấu trúc quan hệ truyền thống và cần phải có một cách tiếp cận đặc biệt. Cơ chế các đặc tính được thực thi trong «1C:DOANH NGHIỆP» cung cấp cho người lập trình một công cụ mẫu để giải quyết các bài toán tương tự. Nguyên mẫu chuyên dụng được sử dụng để tổ chức lưu lại danh sách các dạng đặc tính. Khi đó, mỗi dạng này được có kiểu dữ liệu của riêng mình. Ví dụ, dung tích động cơ sẽ được quy định bằng số, còn màu sắc được chọn từ danh mục. Điều quan trọng là các dạng đặc tính cụ thể không chỉ được xác định trong chính ứng dụng, mà nó còn được bổ sung bởi người sử dụng trong quá trình khai thác hệ thống mà không cần thay đổi giải pháp ứng dụng. Lúc này, người lập trình có thể sử dụng cơ chế này trong cả các tình huống khác nhau nhất, ví dụ, để lưu các tổ hợp của các đặc tính có liên hệ lô-gíc của hàng hóa và để tính giá thành theo tổ hợp đó.

Một giải pháp thú vị nữa, đó là cơ chế của các quy trình nghiệp vụ (work-flow), cho phép người lập trình tổ chức cho nhiều người sử dụng cùng đồng thời làm việc khi thực hiện các thao tác mẫu theo trình tự công việc. Trong nhiều hệ thống thông tin hiện có, để giải các bài toán của các quy trình nghiệp vụ, người ta sử dụng các sản phẩm phần mềm chuyên dụng, những sản phẩm này cần được tích hợp với các ứng dụng để giải quyết các bài toán quản lý kinh tế. Trong nền tảng «1C:DOANH NGHIỆP», cơ chế quy trình nghiệp vụ được tích hợp hoàn toàn vào một hệ thống đến mức người lập trình và người sử dụng đều không thể phát hiện được “lớp đệm” phân cách cơ chế này với các tính năng khác. Cơ chế này bao gồm các công cụ dùng để mô tả các sơ đồ quy trình nghiệp vụ trong giải pháp ứng dụng và để mô tả vai trò lộ trình của chúng, để định dạng các nhiệm vụ cần thực hiện trong mỗi điểm của lộ trình, để quản lý quy trình nghiệp vụ và để tổ chức liên kết với các chức năng khác của giải pháp ứng dụng. Cơ chế này bảo đảm cung cấp cho người lập trình có khả năng linh hoạt quản lý việc phân nhánh quy trình và lập ra các nhiệm vụ. Ví dụ, ngoài lệnh có điều kiện, người lập trình có thể mô tả trực quan việc tiến hành song song một số nhánh của lộ trình, và chỉ ra điểm hội tụ của chúng. Được phép định hướng nhiệm vụ cho một nhóm những người sử dụng tiềm năng, ví dụ, nếu mở tài khoản thì người đứng tên chủ tài khoản phải là một trong số những người quản lý bộ phận. Và ngược lại, tại một điểm lộ trình, ta có thể kích hoạt một số các nhiệm vụ, ví dụ, nếu là báo cáo tài chính thì tất cả các lãnh đạo của các phòng cần phải đệ trình. Vai trò lộ trình cho phép không chỉ lập ra nhiệm vụ cho một nhân viên cụ thể mà còn cho phép phân chia nhiệm vụ theo vai trò, theo các bộ phận hay theo các tiêu chí khác mà người lập trình giải pháp ứng dụng có thể mô tả chúng. Khi thực hiện vai trò lộ trình, được phép chỉ ra việc phân chia trách nhiệm của nhân viên, có tính đến yếu tố thay thế tạm thời hay tổng hợp một số các trách nhiệm... Có thể nhấn mạnh, cơ chế này sẵn sàng đáp ứng được việc tự động hóa hoạt động liên kết nhân viên của doanh nghiệp. Để mô tả các quy trình nghiệp vụ đơn giản, bằng phương pháp trực quan, người lập trình có thể đưa ra sơ đồ lộ trình, và chỉ ra các điều kiện chuyển tại các nút của chúng. Tất cả các thao tác còn lại đều do hệ thống thực hiện tự động. Khi triển khai các quy trình nghiệp vụ phức tạp, người lập trình cần phải chú ý đến việc thực hiện chặt chẽ các thao tác có liên quan tới các tính năng của giải pháp ứng dụng.

Trong các ví dụ minh họa nêu trên, mỗi trong số các nguyên mẫu của các đối tượng ứng dụng và các cơ chế hợp nhất đều bao trùm lên một vùng xác định các bài toán của lĩnh vực ứng dụng và

giảm đáng kể chi phí cho việc lập trình, nhưng điều quan trọng hơn là bảo đảm được độ chuẩn hóa cho các giải pháp ứng dụng.

Để có thể hình dung về số lượng các công cụ đáp ứng cho người lập trình, sau đây chúng tôi sẽ mô tả ngắn gọn thêm một số các cơ chế.

Như vậy, để giải quyết nhiệm vụ có liên quan kế toán biến động tài sản (tài chính và vật chất), trong hệ thống có sử dụng cơ chế biểu ghi tích lũy.

Biểu ghi tích lũy hỗ trợ hệ thống kế toán đa chiều có thành phần tùy ý về chiều đo và số đo, bảo đảm việc tối ưu hóa khi nhận lại các kết quả tổng kết tại mọi thời điểm khác nhau, và như vậy, cho phép giải quyết hiệu quả các nhiệm vụ kế toán nguyên vật liệu, lập kế hoạch sản xuất và tài chính, thanh toán với khách hàng...

Cơ chế kế toán doanh nghiệp là “động cơ” tổng hợp để tự động giải quyết các nhiệm vụ tự động hóa công tác kế toán trong các mô hình kế toán khác nhau.

Cơ chế này có thể được sử dụng ở nhiều nước khác nhau và có độ linh hoạt cao đối với việc tùy chỉnh bổ sung. Trên nền tảng cơ chế là nguyên tắc ghi kép, hơn nữa, nó cho phép làm việc với các giao dịch tài chính, bao gồm các đối ứng tài khoản đơn giản (một Có và một Nợ) được áp dụng trong kế toán ở Nga, và cả trong các mô hình phức tạp như đang được áp dụng rộng rãi tại nhiều nước thuộc phương Tây (nhiều Có và nhiều Nợ).

Cơ chế hỗ trợ hệ thống kế toán đa chiều có thành phần các chiều đo và số đo bất kỳ. Tính chất đặc thù của nó là khả năng tùy chỉnh thành phần các chiều đo bổ sung mà không lệ thuộc vào từng tài khoản. Người lập trình có thể lập ra thành phần các chiều đo khi xây dựng giải pháp ứng dụng, còn người sử dụng có thể thực hiện việc tùy chỉnh trong quá trình làm việc với hệ thống. Các kết quả tổng kết nhiều chiều và nhiều lớp thu được trong khuôn khổ của các kỳ tính toán, của các đối ứng tài khoản... được khởi tạo một cách tự động.

Cơ chế thông tin được sử dụng để giải quyết các nhiệm vụ có liên quan tới việc lưu giữ các thông tin khác nhau về đối tượng theo các bình diện khác nhau.

Ví dụ, các dữ liệu về giá hàng hóa có thể được lưu trong khuôn khổ mặt hàng, loại khách hàng... Khi cần thiết có thể đảm bảo việc lưu lại lịch sử của các thay đổi đã được đưa vào, để khi cần thiết, có thể lấy ra các dữ liệu đã được lưu đó. Cơ chế này bảo đảm toàn bộ tính lô-gíc để làm việc với các thông tin được lưu, bắt đầu từ việc điều khiển cho đến việc tự động hiển thị trong giao diện sử dụng.

Một lần nữa xin lưu ý, trong mô hình của «1C:DOANH NGHIỆP», các cơ chế ứng dụng và các nguyên mẫu không chỉ đơn thuần là bộ các khuôn mẫu mà người lập trình có thể sử dụng chúng “khi cần thiết”. Toàn bộ giải pháp ứng dụng nhất thiết phải được dựa trên việc sử dụng các nguyên mẫu được sẵn có.

Bộ nguyên mẫu này (cũng rất gọn nhẹ) cũng khá đầy đủ để đáp ứng được mọi nhu cầu của lĩnh vực ứng dụng. Thực tế, nhờ các nguyên mẫu này, hệ thống cung cấp cho người lập trình một mô hình thiết kế chuẩn và cho phép họ giảm được chi phí đáng kể trong việc xây dựng và hỗ trợ các giải pháp ứng dụng.

Mô hình cao cấp của giao diện

Rõ ràng, đối với một giải pháp ứng dụng thì vấn đề tổ chức giao diện người sử dụng có một ý nghĩa đặc biệt. Trước hết, vì đối với đại đa số người sử dụng thì các ứng dụng này là công cụ chính mà hàng ngày họ làm việc với nó, trong khi họ lại không phải là người sử dụng máy tính chuyên nghiệp. Thứ đến, vì giao diện của các giải pháp ứng dụng thường rất “lớn”. Khác với nhiều loại hệ thống của giải pháp ứng dụng, thường trong giao diện chứa hàng trăm, có khi hàng ngàn các biểu mẫu. Ngoài ra, các biểu mẫu này có thể được thay đổi theo định kỳ, ví dụ, trong quá trình phát triển hệ thống hay khi cần phải thay đổi lô-gíc nghiệp vụ. Cho nên, chi phí để xây dựng giao diện của giải pháp ứng dụng là rất lớn. Chính vì vậy, trong nền tảng đã được đưa vào một loạt các cơ chế để giúp tạo nhanh giao diện người sử dụng. Trong các cơ chế này có thực thi riêng mô hình của sổ, hệ thống các biểu mẫu, bộ các phần tử điều khiển...

Ý tưởng chính về việc xây dựng giao diện trong hệ thống «1C:DOANH NGHIỆP» là việc sử dụng tối đa thông tin từ metadata, kể cả các đối tượng điều khiển và sử dụng dữ liệu, để sao không cần phải tùy chỉnh chi tiết toàn bộ cấu trúc từ phía người lập trình, và để hệ thống có thể làm việc phần lớn ở chế độ tự động. Như chúng tôi đã nói ở trên, các đối tượng của ngôn ngữ hệ thống để điều khiển dữ liệu đồng thời cũng được sử dụng cả để hiển thị trong giao diện. Người sử dụng chỉ cần liên kết đối tượng này với phần tử điều khiển, còn cơ chế giao diện đảm trách hoàn toàn việc tổ chức xem và xử lý dữ liệu. Thế nhưng, hiệu quả lớn nhất mà nó đạt được là việc thiết lập mối liên hệ giữa đối tượng điều khiển dữ liệu và chính biểu mẫu (hộp thoại), chứ không phải với các phần tử điều khiển riêng biệt. Trong trường hợp này, toàn bộ chức năng của biểu mẫu có liên quan đến việc xem và soạn thảo dữ liệu, kể cả mỗi ràng buộc của nó với các hộp thoại khác được tự động đảm bảo bởi hệ thống. Để làm việc đó, trong hệ thống «1C:DOANH NGHIỆP» có thực thi cơ chế mở rộng biểu mẫu và mở rộng phần tử điều khiển. Các mở rộng được tự động “bật lên” và có tính đến kiểu dữ liệu mà phần tử điều khiển hay hộp thoại có liên quan tới chúng.

Ví dụ, hộp nhập liệu bảo đảm toàn bộ các thao tác cần thiết để phục vụ việc soạn thảo các giá trị tham chiếu (chọn từ các hộp thoại riêng, chọn kết hợp, xóa, tự động đánh dấu giá trị chưa điền, chuyển đến biểu mẫu soạn của đối tượng được chọn...). Sử dụng cơ chế này để nhập theo dòng rất hiệu quả, nó cho phép đẩy nhanh tốc độ nhập thông tin lên nhiều lần, ngoài trừ phải lựa chọn các giá trị từ các hộp thoại của các danh sách. Để thực thi nó, trong các thuộc tính của metadata ta cần tạo danh sách các trường, theo đó người sử dụng có thể nhanh chóng tìm được đối tượng cần thiết, ví dụ, mã hàng, tên hàng. Khi chọn các ký tự để đưa vào hộp nhập liệu, hệ thống sẽ tự động điền giá trị, một trong số các trường đã được nói đến trong metadata của giá trị đó sẽ được bắt đầu từ chính các ký tự đó. Nếu số lượng các giá trị tìm được lớn hơn một thì người sử dụng cần phải tiến hành chọn từ danh sách. Lúc này không đòi hỏi gì thêm từ người lập trình, bởi vì cơ chế giao diện được dựa trên các thuộc tính của metadata.

Một ví dụ lý thú khác, đó là hộp bảng. Phần tử điều khiển này là một cơ chế mạnh được dùng để xem và soạn thảo dữ liệu, đáp ứng cho người sử dụng khả năng xem và soạn thảo các danh sách lớn nhờ cơ chế đọc động. Ở phần trên, chúng ta đã nói về cơ chế đọc động như một trong số các công cụ xử lý dữ liệu. Trong cơ chế giao diện của «1C:DOANH NGHIỆP», nó cho phép xem được các mảng dữ liệu trên thực tế ta không thể xem hết được trong bộ nhớ do hạn chế về kích thước. Khi đó người sử dụng có thể xem dữ liệu theo danh sách cũng không khác so với việc xem trong bộ nhớ của máy tính. Cơ chế này hỗ trợ khả năng linh hoạt để tìm và lọc, xem các cấu trúc phân nhánh, soạn thảo dữ liệu trong danh sách và trong các hộp thoại khác nhau, tùy chỉnh thành phần các cột và sắp xếp, in các danh sách và đưa chúng ra theo các định dạng khác nhau...

Đối với các hộp thoại, các danh sách và các phần tử khác, hệ thống tạo tự động các thanh lệnh (nút lệnh, menu và các thanh lệnh hoàn chỉnh) để bảo đảm mọi thao tác, giúp ta có thể xem, soạn thảo, kể cả để liên kết với các hộp thoại khác và các chức năng tiện ích khác.

Sau khi mô tả đối tượng metadata trong giải pháp ứng dụng, người lập trình, nói chung, có thể không cần tạo ra biểu mẫu, vì tất cả các hộp thoại cần thiết sẽ được tạo tự động trong quá trình làm việc của hệ thống. Nếu người lập trình muốn tạo ra biểu mẫu để lập cho nó các thuộc tính đặc thù thì bộ thiết kế sẽ cung cấp hộp thoại chuẩn mà ngay từ đầu không chứa bất kỳ một mã lệnh nào cả, bởi vì phần mở rộng của các phần tử điều khiển và biểu mẫu đã đảm bảo toàn bộ chức năng cần thiết. Dĩ nhiên, người lập trình, không cần đến bộ dựng, cũng có thể chèn các điều khiển vào biểu mẫu, liên kết chúng với dữ liệu và nhận được cấu trúc làm việc hoàn chỉnh mà không cần viết thêm bất kỳ một dòng lệnh nào. Người lập trình có thể viết bổ sung một cái gì đó chỉ khi họ muốn xác định trước hoặc mở rộng các tính năng chuẩn.

Điều quan trọng là giao diện của «1C:DOANH NGHIỆP» không chỉ đơn giản đáp ứng công việc của các biểu mẫu và các phần tử điều khiển riêng biệt. Cơ cấu điều khiển các hộp thoại của các đối tượng ứng dụng bảo đảm các phương án liên hệ khác nhau giữa biểu mẫu được sử dụng khi chọn các giá trị, khi lựa chọn theo tổ hợp, khi diễn giải thông tin... Thực tế, nền tảng của «1C:DOANH NGHIỆP» sẵn sàng đáp ứng cho người lập trình một chiến lược tổ chức toàn bộ giao diện của giải pháp ứng dụng, chiến lược này có các phương pháp triển khai tất cả các kịch bản làm việc cần thiết của người sử dụng.

Nhờ sử dụng trong metadata các kiến thức về các mối liên hệ giữa các đối tượng khác nhau, cho nên hệ thống hỗ trợ việc tự động ra các lệnh chuyển giữa các biểu mẫu có mối ràng buộc lô-gic. Do

vậy, khi làm việc với chúng tôi, người sử dụng không cần sự trợ giúp của người lập trình cũng có thể truy cập được các nội dung khác mà chúng tôi này phản ánh trong các cơ chế kế toán khác nhau. Ngoài các ràng buộc được xác định bằng lô-gíc nghiệp vụ của giải pháp ứng dụng trong metadata, người lập trình còn có thể xác định (dưới dạng các tiêu chí lọc), và cả các ràng buộc bổ sung để chuyển giữa các đối tượng, những đối tượng này cũng sẽ tự động xuất hiện trong thanh giao diện của biểu mẫu.

Có một giải pháp giao diện chuẩn khác, đó là cơ chế “nhập trên cơ sở”. Cơ chế này khi nhập dữ liệu theo một đối tượng, cho phép điền hoàn chỉnh một loạt các trường nhờ chọn các giá trị của chúng từ nguồn dữ liệu của các đối tượng khác. Ví dụ, sau lập hóa đơn thanh toán, người sử dụng có thể ấn nút để tạo ra hóa đơn bán hàng trên cơ sở hóa đơn thanh toán này; để làm việc này, ta chỉ cần thiết lập trong metadata mối liên hệ giữa các đối tượng đã được nói đến và mô tả các quy tắc điền các trường.

Người lập trình giải pháp ứng dụng được cung cấp các công cụ chuẩn để triển khai việc “thay đổi” giao diện lệnh. Trong các giải pháp ứng dụng đều có khối lượng lớn các chức năng, điều đó cho phép người sử dụng có thể chuyển sang chế độ làm việc khác bằng cách thay đổi một phần hay toàn bộ giao diện lệnh mà không cần phải thoát ra khỏi hệ thống.

Cơ chế biểu mẫu đảm bảo cho việc xác định chiến lược nhận diện (ID) biểu mẫu, cho phép không mở lại các biểu mẫu đã mở mà chỉ cần kích hoạt các biểu mẫu được mở trước đó.

Như vậy, nhờ cơ chế mở rộng của biểu mẫu và các cơ chế khác, trong hệ thống được hỗ trợ đầy đủ chiến lược giao diện người sử dụng, bao gồm các phương án tương tác giữa các biểu mẫu khác nhau, tự động tạo giao diện lệnh để chuyển theo các chế độ và các chức năng khác nhau của hệ thống, kể cả các chế độ tiện ích khác nhau nhằm bảo đảm cho người sử dụng làm việc thuận tiện nhất.

Như chúng tôi đã nói, trong giao diện của «1C:DOANH NGHIỆP» đã được thực thi nhiều phần tử chuyên dụng, nhưng không với ý định để sử dụng các công cụ mẫu của hệ điều hành. Việc làm này là để tạo được giao diện đáp ứng tối đa tính đặc thù của giải pháp ứng dụng.

Chẳng hạn, trong hệ thống «1C:DOANH NGHIỆP» có thực thi mô hình riêng về hệ thống cửa sổ để làm việc với lượng lớn các thông tin khác loại. Ví dụ, người sử dụng có thể mở cực đại cho riêng từng cửa sổ chứ không phải dùng chế độ thống nhất đối với toàn bộ ứng dụng (như đã được thực hiện trong các ứng dụng chuẩn MDI). Ngoài các dạng cửa sổ truyền thống trong hệ thống cửa sổ riêng, còn sử dụng cửa sổ *đính kèm* và cửa sổ *ẩn*. Trong giao diện của «1C:DOANH NGHIỆP», từ cửa sổ tình thái (modal) có thể gọi vào các cửa sổ phi tình thái, nhờ vậy mà có thể tạo được nhiều lớp giao diện đa cửa sổ. Người lập trình có thể mở bất kỳ mọi hộp thoại tình thái mà không cần thay đổi lô-gíc làm việc của chúng.

Có hỗ trợ chế độ tự động điều chỉnh kích thước các phần tử điều khiển khi thay đổi kích thước biểu mẫu, hay kéo chuyên đường phân tách trong đó. Như vậy, người sử dụng có thể thiết lập kích thước mà họ muốn cho từng biểu mẫu, hay cho từng phần riêng biệt của nó, còn các biểu mẫu có thể sử dụng toàn bộ không gian mà nó có để hiển thị thông tin hiệu quả nhất. Điều quan trọng là cơ chế không đòi hỏi phải tùy chỉnh từ phía người lập trình.

Các biểu mẫu và phần tử điều khiển được thiết kế riêng, một mặt để phản ánh được tối đa lượng thông tin, mặt khác để giảm nhẹ cường độ lao động cho người sử dụng khi họ phải làm việc kéo dài với máy vi tính. Để làm việc đó trong hệ thống đã sử dụng bộ thiết kế giao diện “phẳng” gần tương tự với bộ thiết kế Web. Để điều khiển bộ thiết kế giao diện, trong hệ thống đã được chuẩn bị sẵn cơ chế kiểu cách, cho phép thay đổi tập trung hình dạng ngoài của giải pháp ứng dụng. Thực tế thì mỗi phần tử điều khiển đều “được trang bị” chức năng phù hợp với đặc thù của các bài toán kinh tế. Ngoài ra, cũng đã đưa vào một loạt các phần tử điều khiển để phục vụ trực tiếp cho các bài toán phân tích, ví dụ, đó là biểu đồ, biểu đồ Gantt...

Trong các cơ chế giao diện của «1C:DOANH NGHIỆP» còn có rất nhiều phần tử điều khiển đáng được quan tâm, trên đây ta chỉ mới liệt kê một số các giải pháp. Điều quan trọng là tất cả các cơ chế giao diện được cung cấp cho người lập trình ở mức độ khái niệm khá cao. Như đã nói, đặc điểm quan trọng của việc lập trình trong hệ thống «1C:DOANH NGHIỆP» là tách rời giải pháp ứng dụng ra khỏi các chi tiết kỹ thuật khác nhau. Ví dụ, người lập trình không có quyền điều khiển chuột hay vẽ chi tiết các phần tử điều khiển. Ngay cả những cơ chế khá phức tạp cũng được cung cấp

cho người lập trình dưới dạng để sao cho, một mặt, họ có thể hoạt động mà không cần thêm bất kỳ tùy chỉnh nào, mặt khác họ cũng được trang bị khả năng tùy chỉnh ở mức độ cao mà không cần có hiểu biết đặc biệt. Nhờ đó, trong các giải pháp ứng dụng được trang bị giao diện có chức năng hoàn toàn hiện đại, đồng thời, chỉ một phần nhỏ các mã nguồn được dành cho phần hiển thị trên màn hình, còn phần lớn mã lệnh là dùng để triển khai các lô-gíc nghiệp vụ của giao diện giải pháp ứng dụng.

Cũng cần nhắc đến cơ chế Web mở rộng. Cơ chế này cho phép triển khai giao diện Web cho các giải pháp ứng dụng của «1C:DOANH NGHIỆP». Chúng ta sẽ không nói chi tiết về thiết bị công nghệ của cơ chế này. Điều đáng quan tâm hơn trong việc triển khai ở đây là nó cũng như giao diện chính (rich) đều sử dụng tối đa thông tin từ metadata để tạo tự động các biểu mẫu Web, cũng như sự hiểu biết về chức năng và cơ cấu của các nguyên mẫu của các đối tượng ứng dụng. Biểu mẫu để soạn thảo đối tượng bất kỳ hoặc để xem danh sách các đối tượng được tạo tự động bằng cấu phần Web mở rộng và cũng có thể được xác định thủ công trong thiết kế bởi người sử dụng. Nếu người lập trình tự tạo lấy hộp thoại thì họ có thể sử dụng các phần tử điều khiển chuyên dụng, các phần tử này không chỉ cho phép họ tổ chức mối liên kết với dữ liệu cần thiết mà còn được đảm bảo chức năng cần thiết để xem và soạn thảo. Ví dụ, trong các hộp nhập liệu có hỗ trợ việc chọn từ các hộp thoại danh sách và lựa chọn theo các chữ cái đầu, trong các danh sách được hỗ trợ xem theo phân nhánh, thiết lập chế độ lọc và sắp xếp (do người sử dụng thực hiện). Hệ thống đáp ứng bộ các thao tác chuẩn để “bảo trì” dữ liệu, và tự tổ chức liên kết giữa các biểu mẫu ứng dụng. Nhờ đó mà đảm bảo được sự đồng nhất của hai phương án giao diện sử dụng (dĩ nhiên là cần sự điều chỉnh để hòa hợp với môi trường Web), kể cả tự động hóa tối đa việc lập trình của chúng trên cơ sở sử dụng thông tin từ metadata.

Một trong số các kết quả quan trọng trong việc xây dựng giao diện sử dụng trên cơ sở các công cụ chuẩn được đáp ứng bởi nền tảng, đó là giao diện của tất cả các giải pháp ứng dụng đều được xây dựng thống nhất. Điều này cho phép người sử dụng áp dụng tối đa kinh nghiệm của những người đi trước. Sau khi đã biết cách làm việc với giao diện của một giải pháp ứng dụng, người sử dụng sẽ có thể dễ dàng bắt đầu làm việc với các giải pháp khác của «1C:DOANH NGHIỆP».

Các cơ chế thông minh để lập báo cáo

Trong bất kỳ giải pháp nào đều có hai cơ chế chính để đáp ứng việc tương tác giữa người sử dụng và hệ thống, đó là: giao diện dùng để nhập thông tin và các công cụ để lập báo cáo. Bởi vì các cơ chế nhận báo cáo được dùng, trước hết, là để trình bày thông tin cần thiết cho người sử dụng để nhận lại các kết quả quản lý, cho nên trong «1C:DOANH NGHIỆP» nó được đặc biệt lưu ý.

Công nghệ lập báo cáo được áp dụng trong «1C:DOANH NGHIỆP» có chứa một loạt các giải pháp độc nhất vô nhị, không có điểm tương tự trong các sản phẩm khác. Trước hết, trong nền tảng «1C:DOANH NGHIỆP» không có các công cụ truyền thống để làm “động cơ báo cáo” riêng như đối với phần lớn các hệ thống khác. Để lập báo cáo, người lập trình được cung cấp đầy đủ các cơ chế, chúng có thể được sử dụng trong các tổ hợp khác nhau hay kết hợp với các cơ chế khác. Nhờ đó mà trong «1C:DOANH NGHIỆP», các báo cáo được hòa hợp với giao diện chung của ứng dụng. Thực tế thì trong quá trình làm việc, người sử dụng không nhìn thấy ranh giới giữa giao diện chung và cơ chế báo cáo. Theo chúng tôi, trong hệ thống ứng dụng quản lý kinh tế hiện đại không thể khác hơn: thế nhưng, thông tin phân tích có thể được sử dụng trong tất cả các chế độ và trong các hộp thoại, còn các báo cáo thì được tạo theo khối lượng lớn thông tin mà chúng sẽ không được in ra, mà để phân tích và lập ra báo cáo. Cho nên trong «1C:DOANH NGHIỆP», các công cụ dùng để chuẩn bị lập báo cáo có liên quan chặt chẽ với các cơ chế khác và có các tính năng mạnh để làm việc trực tiếp.

Một trong số các cơ chế nổi bật của loại này đã được đáp ứng bởi nền tảng, đó là bộ lập báo cáo, dùng để đảm bảo khả năng nhận lại báo cáo nhờ chức năng mở rộng với chi phí tối thiểu.

Chúng tôi muốn người lập trình hoặc người sử dụng chỉ thực hiện các mô tả ngắn gọn để đưa ra những thông tin cần phân tích, và những gì còn lại để tạo ra báo cáo thì do bộ lập báo cáo tự làm lấy.

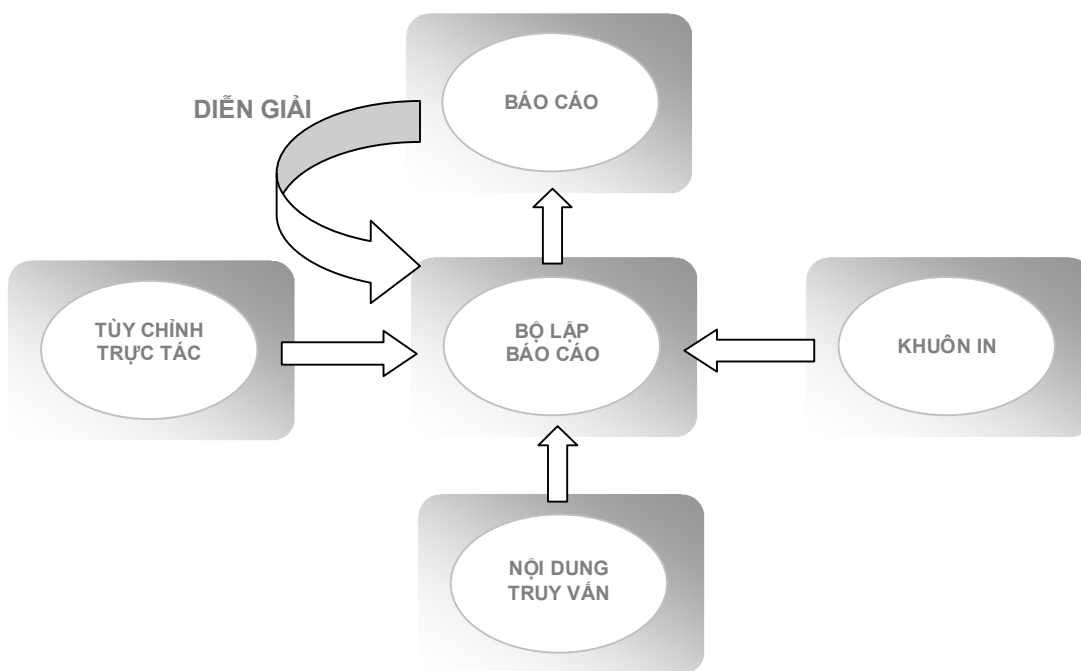
Việc lựa chọn thông tin gốc được tiến hành trên cơ sở các mô tả đã được thực hiện nhờ ngôn ngữ truy vấn của «1C:DOANH NGHIỆP». Do được sử dụng các bảng ảo của các cơ chế kế toán, nội

dung truy vấn của một báo cáo có độ phức tạp vừa phải sẽ chiếm một số dòng lệnh. Cho nên nó không cần mô tả bằng tay. Trong nhiều trường hợp truy vấn được tạo bởi các công cụ của bộ thiết kế trực quan, và thường thì toàn bộ chu trình rút ngắn được thời gian chọn nguồn dữ liệu và chỉ ra các trường kết quả. Công cụ này cho phép tạo trực quan các cơ cấu lô-gic có độ phức tạp bất kỳ, bao gồm việc liên kết các truy vấn và lồng các truy vấn, kể cả khả năng “tăng cường” nội dung truy vấn để soạn thảo (nội dung này có bất kỳ mọi nơi trong mô-đun).

Trên cơ sở phân tích nội dung truy vấn một cách có hệ thống, bộ thiết kế tự tạo ra toàn bộ cơ sở cần thiết không chỉ để tạo báo cáo mà còn cho phép người sử dụng thực hiện một số các tùy chỉnh chi tiết, kể cả để di chuyển giữa các báo cáo. Kết quả là tạo ra hộp thoại, trong đó chứa báo cáo riêng cũng như các công cụ khác nhau để điều khiển chúng, đó là chọn các trường cần đưa vào báo cáo, chọn phễu lọc thông tin theo các tiêu chí phức tạp, gom nhóm theo dòng và theo cột, thiết lập chế độ sắp xếp dữ liệu... Báo cáo có thể được phản ánh trong dạng bảng với nhiều thứ bậc các cột và các dòng, các biểu đồ, biểu đồ tổng hợp, hay các bảng tổng hợp.

Cần biết rằng, việc thiết lập trực tiếp chế độ lọc cũng đảm bảo các tính năng này, như khi thiết lập bộ lọc theo các trường bằng có liên quan (nhận “qua dấu chấm”), chỉ ra tiêu chí của bộ lọc tổ hợp các giá trị hay nhóm các đối tượng mà trong thứ bậc của chúng cần phải chọn thông tin. Ví dụ, người sử dụng có thể lập báo cáo mua hàng bằng cách chỉ đưa vào các hóa đơn chứa nội dung hàng hóa và nhà sản xuất thuộc các nhóm lệnh “Trong nước” và “Nước ngoài”. Bộ dựng báo cáo cũng cho phép lưu tùy chỉnh giao diện báo cáo hiện tại để sau đó sử dụng lại.

Ngoài ra, bộ dựng báo cáo còn đưa toàn bộ thông tin cần thiết vào báo cáo đã được lập để diễn giải cho từng ô riêng biệt của báo cáo (Drill-down). Việc diễn giải các ô được thực trong chế độ trực tác, tức là lần lượt nhấp đúp chuột trực tiếp lên các ô đã được chọn của báo cáo để mở ra báo cáo chi tiết trên cơ sở dữ liệu của các ô và của thông tin chứa trong báo cáo ban đầu:



Điểm quan trọng là người lập trình được đáp ứng tất cả các tính năng này ở dạng sẵn có và chúng có thể được sử dụng theo các tổ hợp khác nhau. Vì vậy đối với từng báo cáo cần tạo, nó có thể linh hoạt xác định “mức tự do” cần phải đảm bảo cho người lập trình, (tất cả những điều đó đều được ghi trong truy vấn). Ví dụ, có thể chỉ ra theo trường nào người sử dụng có thể thiết lập chế độ lọc, sắp xếp, gom nhóm... Chính các phần tử điều khiển báo cáo được đưa vào hộp thoại cũng là do ý của người lập trình. Vì tất cả các thành phần của cơ chế này có thể được sử dụng độc lập, nên người lập trình có thể áp dụng bộ thiết kế báo cáo không chỉ để chuẩn bị lập báo cáo mà còn để giải quyết nhiều nhiệm vụ khác. Ví dụ, người lập trình có thể tạo thủ tục thay đổi dữ liệu theo nhóm; trong chu trình đó để điều khiển việc lựa chọn dữ liệu ta cần cho khởi động tất cả các cơ chế của bộ

thiết kế, nhưng sau đó lựa chọn đã thu được sẽ không được phản ánh trong báo cáo mà sẽ được chuyển qua bộ xử lý, ví dụ, đối với nhóm hàng hóa đã được chọn sẽ được điều chỉnh lại giá.

Kết quả quan trọng của việc sử dụng cơ chế này là trên cơ sở của nó trong các giải pháp ứng dụng của «1C:DOANH NGHIỆP» đã triển khai các công cụ mà trong đó những người sử dụng “có kinh nghiệm” có thể tự lập được những báo cáo có đủ độ phức tạp. Trên cơ sở của cơ chế này đã xây dựng được bộ các báo cáo sử dụng cho nhiều giải pháp ứng dụng và để bảo đảm cho người sử dụng khả năng thiết kế các báo cáo riêng, có nghĩa là họ có thể lập các báo cáo, thiết kế, thậm chí họ có thể tạo được hệ thống các báo cáo chi tiết có mối ràng buộc lẫn nhau.

Chúng ta cũng không thể bỏ qua hiệu quả sử dụng của các công cụ hiển thị. Trong «1C:DOANH NGHIỆP» có sử dụng văn bản dạng bảng (không kèm các công cụ tính toán) để xem và để in các báo cáo. Ý tưởng chính của giải pháp này là sử dụng mô hình bảng điện tử (Spreadsheet) để làm công cụ tạo các mẫu báo cáo mà không cần phải sử dụng nó như là một công cụ tính toán.

Khác với hầu hết các bảng tính khác, văn bản dạng bảng của «1C:DOANH NGHIỆP» có thể chứa các bảng có kích thước lớn theo chiều rộng cũng như chiều cao. Trong văn bản dạng bảng loại này có đầy đủ mọi tính năng hiển thị, cho phép một mặt là để giúp tạo các báo cáo phân tích, mặt khác là để lập ra các bản mẫu sát với các đối tượng chứa trong chúng, phù hợp với các tọa độ khu vực.

Trong nhiều trường hợp, việc điền báo cáo được thực hiện trên cơ sở khuôn mẫu báo cáo. Để làm việc đó, trong văn bản dạng bảng đã thực thi hệ thống linh hoạt tên miền và mô tả các tham số. Người lập trình thiết kế khuôn in bằng cách trực quan, để xác định hình thức bên ngoài của báo cáo bằng cách tách trong nó thành các vùng cần điền dữ liệu khác nhau và mô tả các tham số của chúng. Khi tạo báo cáo là lúc tiến hành hợp nhất các miền với dữ liệu, bằng cách kết hợp phần chứa nội dung báo cáo với hình thức bên ngoài của mẫu báo cáo đã được chọn. Cần biết rằng, khuôn báo cáo có thể được thiết kế tự động bởi bộ thiết kế báo cáo hoặc do chính người lập trình thiết lập. Được phép sử dụng các kiểu cách sẵn có để tạo báo cáo và thiết kế các báo cáo riêng.

Điều quan trọng là các thiết bị tạo các văn bản dạng bảng không chỉ để nhận lại các biểu mẫu in tĩnh. Trong thiết bị này có cơ chế mạnh để trực tiếp điều khiển gom nhóm nhiều lớp, để diễn giải các ô của báo cáo (Drill-down), tự động thiết lập chiều rộng cột... Được phép lưu báo cáo trong các định dạng khác nhau (HTML, XLS, TXT). Ngoài ra, để phân tích trực tiếp các thông tin nhiều chiều trong văn bản dạng bảng, có thể sử dụng bảng tổng hợp, bảng này cho phép điều khiển trực tiếp việc gom nhóm thông tin và điều khiển thành phần của dữ liệu cần được hiển thị mà không cần phải cập nhật báo cáo.

Một tính chất cực kỳ quan trọng của văn bản dạng bảng, đó là khả năng sử dụng chính nó để làm cơ chế nhập và soạn thông tin. Văn bản này có thể chứa bất kỳ các điều khiển được sử dụng trong các hộp thoại của «1C:DOANH NGHIỆP», trong chính các ô hay hình vẽ, cũng như bên trên các ô. Việc này đặc biệt có lợi trong những trường hợp khi chính hộp thoại sẽ vừa là báo cáo, vừa là công cụ để soạn thảo thông tin. Báo cáo về chỉ tiêu kế hoạch mà được phép điều chỉnh là một ví dụ.

Để trình bày trực quan thông tin phân tích, người lập trình được cung cấp đầy đủ các loại biểu đồ, kể cả biểu đồ Gantt, biểu đồ tổng hợp... Hơn nữa các phân tử điều khiển này cũng được đưa vào các hộp thoại như các điều khiển khác và cũng có thể đưa vào các văn bản bảng để nhận được các báo cáo phân tích tổng hợp.

Cơ chế phân tích dữ liệu (Data mining) thông minh cũng rất độc đáo, nhờ cơ chế này, với chi phí tối thiểu, giải pháp ứng dụng có thể được trang bị thêm các công cụ phân tích, ví dụ như cây giải pháp...

Đặc tính quan trọng của nền tảng «1C:DOANH NGHIỆP» là để lập các báo cáo có mức độ phức tạp khác nhau, khi người lập trình không cần kết nối với các cơ chế bổ sung từ bên ngoài. Lúc đó tất cả các công cụ nội tại được kết nối với nhau, dựa trên một hệ thống khái niệm thống nhất và sử dụng tối đa các khả năng của nhau. Ví dụ, bảng tổng hợp được xếp trong văn bản dạng bảng để tự động xử lý truy vấn đến cơ sở dữ liệu theo mức độ sử dụng bằng trực quan để chỉ ra các lớp mới của nhóm, gọi sử dụng các chức năng của bộ thiết kế báo cáo. Dĩ nhiên trong hệ thống có cả bộ thiết kế báo cáo, nó cho phép thiết kế chúng trực tiếp, thế nhưng điều quan trọng là bộ thiết kế chỉ “tập hợp” báo cáo từ các cơ chế đang được áp dụng cả để giải nhiều bài toán khác. Thực tế cho thấy, với phương pháp lập trình này, để tạo được một báo cáo có độ phức tạp vừa phải, người lập trình không cần mất nhiều thời gian.

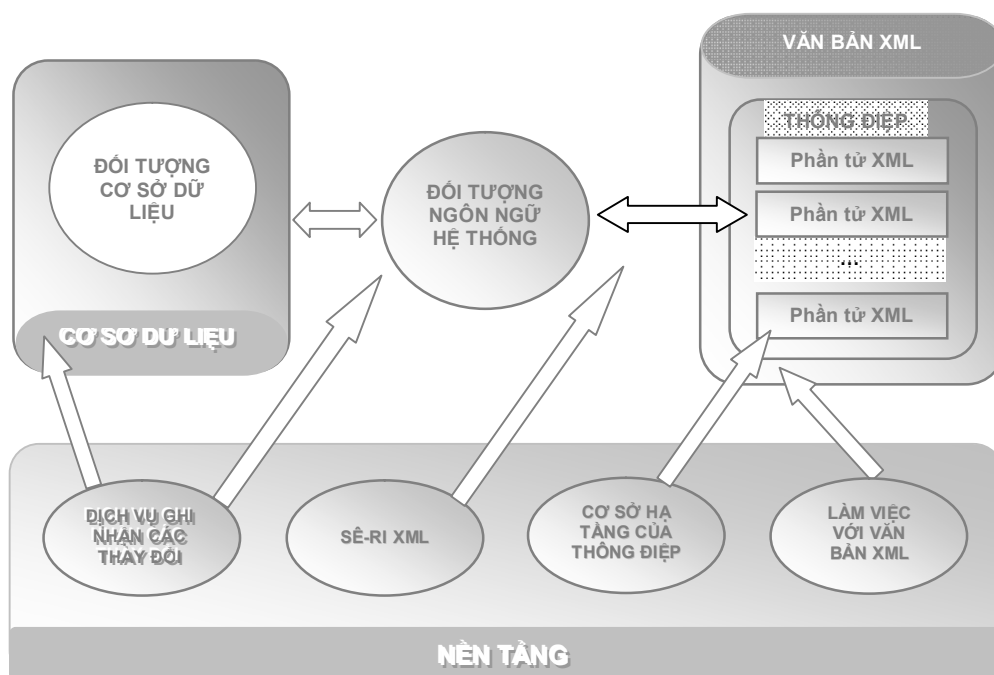
Xây dựng hệ thống thông tin phân tán và tích hợp

Còn có một “Át chủ” của nền tảng «1C:DOANH NGHIỆP», đó là cơ chế trao đổi dữ liệu.

Ban đầu, việc đưa ra quyết định xây dựng cơ chế trao đổi dữ liệu cho các cơ sở thông tin phân tán theo khu vực mà không đòi hỏi phải kết nối thường xuyên, là do tại phần lớn các nhà máy trong nước còn thiếu các kênh dữ liệu bằng thông rộng. Thế nhưng sau đó không lâu, như chúng ta đã biết, việc tạo các cơ chế này đã hội nhập vào dòng chảy của một trong số các hướng viễn cảnh nhất của một ý tưởng công nghệ thế giới.

Ngày nay, trong tay của người lập trình giải pháp có bộ cơ chế trao đổi mạnh, có khả năng giải quyết nhiều nhiệm vụ khác nhau nhất. Bên cạnh việc hỗ trợ hệ thống thông tin phân tán theo phương pháp truyền thống dùng để tích hợp với các chương trình ứng dụng khác, còn có thể xây dựng hệ thống thông tin không đồng nhất mà có sự góp mặt của các giải pháp dựa trên nền tảng «1C:DOANH NGHIỆP» cùng với các ứng dụng ngoài.

Việc ứng dụng các công nghệ trao đổi dữ liệu để giải quyết thành công rất nhiều các bài toán đã cho thấy, kết quả thu được không phụ thuộc duy nhất vào người lập trình mà là bộ cơ chế, chúng có thể được ứng dụng riêng cũng như được kết hợp trong tổ hợp bất kỳ. Các cơ chế này hỗ trợ làm việc với các tài liệu XML, XML-Serialization, hạ tầng thông báo, dịch vụ ghi nhận thay đổi, sơ đồ trao đổi, quản lý cơ sở thông tin phân tán:



Theo chúng tôi, tính chất của bộ các cơ chế này là độc nhất vô nhị vì nó đáp ứng được mức độ sẵn sàng làm việc cao của hệ thống trong môi trường xác định - thực tế việc tổ chức trao đổi này không đòi hỏi các chi phí bổ sung cho việc lập trình. Chỉ cần quy định trực tiếp thành phần dữ liệu cần tham gia trao đổi, còn cơ chế sẽ đảm bảo tạo các thông điệp và kết nhập chúng. Khi đó hệ thống sẽ tự động thực hiện thủ tục trao đổi chỉ đối với những thông tin cần thay đổi, theo dõi để nhận các thông điệp, xác định mức độ cần chuyển lại dữ liệu, cho phép xử lý và kiểm tra sự toàn vẹn của thông tin đã được kết nhập. Khả năng tùy ứng linh hoạt cho phép tạo ra bất kỳ sơ đồ topo của các đầu mối trao đổi (hoa thị, bông tuyết, sơ đồ không có nút trung tâm). Thành phần dữ liệu tham gia trao đổi cũng như các quy chế xử lý số liệu có thể được quy định theo ý người lập trình. Khi đó các cơ chế trao đổi, một mặt hạn chế được tối đa khối lượng thông tin cần chuyển (chỉ chuyển những dữ liệu thay đổi), mặt khác là để đảm bảo không để mất các thông điệp. Nói cách khác, hệ thống có khả năng hoạt động trong điều kiện được cung cấp đầy đủ các thông điệp, cũng như khi không có.

Sự tồn tại trong nền tảng các cơ chế trao đổi hiệu quả nhưng không đòi hỏi các tùy ứng phức tạp là rất cần thiết, trước hết, là để đáp ứng cấu trúc chung trong việc xây dựng các giải pháp ứng dụng

đã được triển khai trong nền tảng của «1C:DOANH NGHIỆP». Chúng ta cũng đã nói, việc chuyển sang hệ thống phân tán và hệ thống tích hợp được đảm bảo bằng kỹ thuật đối tượng quản lý dữ liệu đang được sử dụng trong «1C:DOANH NGHIỆP». Hệ thống của các đối tượng tương ứng đảm bảo các khả năng lưu trữ bình thường (persistence) bất kỳ mọi dữ liệu ứng dụng theo định dạng XML. Ngoài ra, nhờ sự tồn tại của các nguyên mẫu chuẩn của các đối tượng ứng dụng mà nền tảng có khả năng tự quy định cho từng đối tượng những thông tin thay đổi cần chuyển, và chiến thuật trao đổi phù hợp với các chức năng của nó. Vấn đề quan trọng của nhiều hệ thống nổi tiếng được dùng để trao đổi và tiếp nhận là: trong cơ sở ngữ cảnh mô tả dữ liệu không được đưa vào các thông tin để chỉ mục tin nào cần phải trao đổi, các quy định xử lý thông tin, và làm thế nào để bảo đảm được toàn vẹn lô-gíc và sự đồng nhất của dữ liệu. Vì vậy người lập trình khi sử dụng các cơ chế tương tự thường phải mô tả chi tiết các chu trình cho từng mảng thông tin.

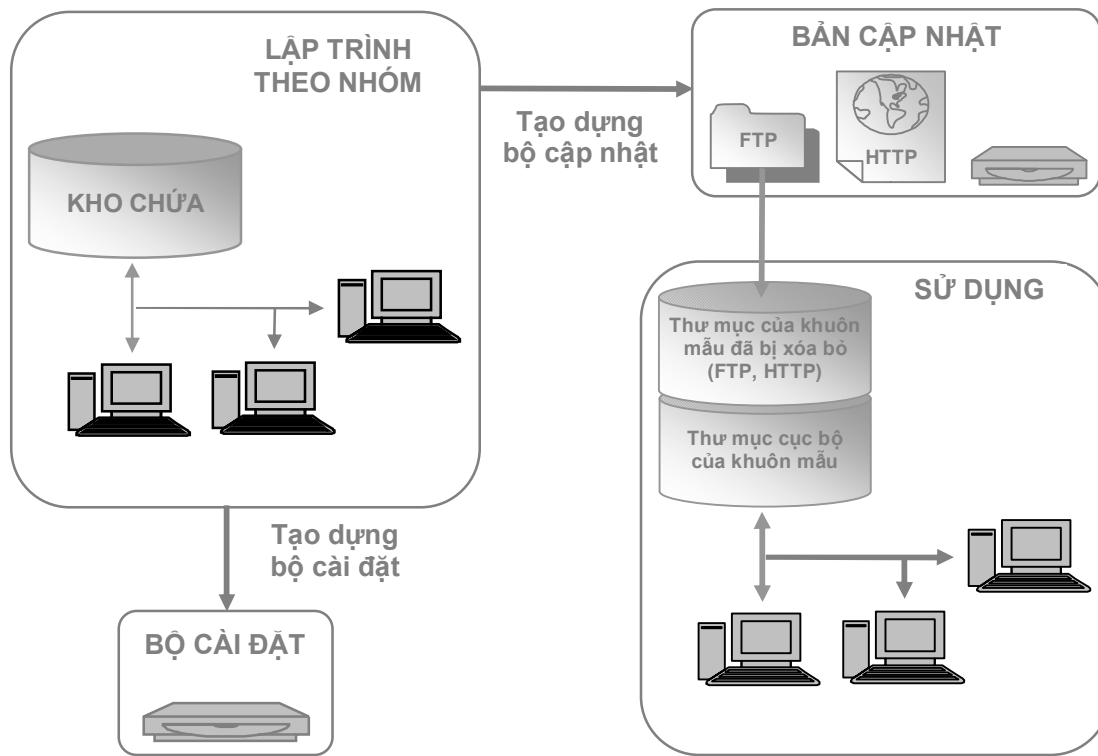
Các cơ chế trao đổi của «1C:DOANH NGHIỆP» hoạt động ở mức các nguyên mẫu chuẩn của các đối tượng ứng dụng, vì vậy nên ngay từ đầu chúng có tất cả các thông tin cần thiết của loại này. Người lập trình chỉ cần chỉ ra những đối tượng của dạng xác định cần tham gia trao đổi. Tiếp đến, tất cả các thao tác như ghi nhận các thay đổi, tạo các thông báo, kết nhập dữ liệu và giải quyết các vướng mắc sẽ được chương trình thực hiện tự động. Người lập trình dĩ nhiên có khả năng thực hiện bằng tay “tùy chỉnh tinh vi”, (nhưng các tùy chỉnh loại này chỉ được yêu cầu trong một số trường hợp riêng).

Một đặc tính quan trọng khác của cơ chế trao đổi của «1C:DOANH NGHIỆP» là nó phù hợp với các quan niệm tiên tiến toàn cầu về tích hợp hệ thống thông tin. Chúng ta đều hiểu rõ, ngày nay không một người lập trình phần mềm quản lý kinh tế nào lại có thể tự xem mình là “hoàng đế của đỉnh cao”, và tự tách mình ra khỏi mối tương tác chặt chẽ với các hệ thống của những đồng nghiệp khác. Trong giao tiếp này thì các tương tác không chỉ đơn thuần là để biết gọi sử dụng một chức năng nào đó từ các chương trình khác, hay để kết nhập vào nó một số dữ liệu nào đó. Vấn đề ở đây là việc xây dựng các hệ thống khác loại, trong các hệ thống này đã được đưa vào một số các giải pháp ứng dụng khác nhau để tạo nên một quần thể hiện hữu. Trong tương lai, xu hướng này sẽ trở thành một trong số các hệ thống phần mềm quản lý kinh tế chiếm ưu thế trên thị trường thế giới. Các cơ chế trao đổi của «1C:DOANH NGHIỆP» có độ sẵn sàng cao để làm việc trong các hệ thống không đồng nhất hôm nay và cả sau này, không chỉ do vì toàn bộ việc trao đổi được tiến hành trong định dạng XML, mà điều quan trọng hơn là nó phù hợp được ý tưởng của các giải pháp đã được triển khai theo khuynh hướng hiện tại. Các cơ chế trao đổi của «1C:DOANH NGHIỆP» cũng cho phép thực hiện tương tác không chỉ với các ứng dụng riêng lẻ mà còn với các nền tảng tích hợp đầy triển vọng khác.

Cài đặt và cập nhật các giải pháp ứng dụng

Thực tế thì tất cả mọi nhà sản xuất chương trình ứng dụng đều đã đưa vào trong sản phẩm của mình các cơ chế để cập nhật phiên bản mới. Thế nhưng đối với các hệ thống như «1C:DOANH NGHIỆP», những vấn đề tương tự lại có tính chất phức tạp riêng. Điều đó có thể hiểu, thứ nhất là do các hệ thống quản lý được cập nhật thường xuyên hơn so với các hệ thống khác, thứ hai là do các thay đổi của giải pháp ứng dụng của «1C:DOANH NGHIỆP» thường được thực hiện “tại chỗ” để tùy ứng sản phẩm đóng gói sao cho thích hợp với các nhu cầu cụ thể của khách hàng. Đặc tính thứ hai có tính chất phức tạp riêng, bởi vì mỗi khi người lập trình tung ra thị trường bất kỳ một sản phẩm nâng cấp nào đó thì đều nảy sinh yêu cầu mà họ (những người lập trình) phải tiến hành đồng bộ hóa tất cả các thay đổi đã đưa vào để phù hợp điều kiện cụ thể của doanh nghiệp.

Để giải quyết các vấn đề vừa nêu và những nhiệm vụ khác có liên quan với việc cài đặt và cập nhật các giải pháp ứng dụng, trong «1C:DOANH NGHIỆP» đã triển khai hoàn chỉnh bộ cơ chế để trợ giúp người lập trình và người sử dụng. Cơ chế này bao quát toàn bộ quá trình công nghệ cài đặt và hỗ trợ: chuẩn bị các tệp đóng gói, chuẩn bị các bản cập nhật bổ sung và các bộ cài đặt trọn gói, đưa các phiên bản cập nhật lên mạng, tìm kiếm tự động và thực hiện cập nhật, điều khiển thành phần hỗ trợ ở mức các đối tượng cấu hình....:



Một trong số các yếu tố quan trọng hơn của công nghệ cập nhật là *cơ chế* đảm bảo việc đồng bộ hóa các thay đổi do nhà cung ứng giải pháp ứng dụng đã đưa vào để phù hợp với yêu cầu của một doanh nghiệp cụ thể. Cơ chế này đảm bảo các chức năng mạnh để phân tích và so sánh các thay đổi, kể cả việc đảm bảo các công cụ điều khiển đồng bộ hóa. Nhà quản lý hoặc người lập trình có thể thực hiện chi tiết việc đồng bộ hóa các cập nhật đến từng đối tượng, từng thuộc tính và từng thủ tục của các mô-đun. Ví dụ, nếu một chuyên gia chịu trách nhiệm bảo trì giải pháp ứng dụng cho doanh nghiệp thì họ sẽ đánh dấu những đối tượng được dự định hỗ trợ độc lập, các đối tượng này sau đó sẽ không được cập nhật thay đổi khi nhà cung cấp sản phẩm đưa ra phiên bản nâng cấp mới. Nếu các đối tượng cần được hợp nhất thì để tạo đơn giản việc đồng bộ hóa các thay đổi, ta có thể xây dựng thứ tự ưu tiên của việc hợp nhất.

Cơ chế cập nhật cho phép xây dựng hệ thống bảo trì phức tạp nhiều lớp. Nhà cung cấp sản phẩm chuyên dụng có thể dựa vào người tạo ứng dụng tổng hợp để thực hiện công việc hỗ trợ cho khách hàng, đáp ứng riêng biệt cho từng khách hàng. Người sử dụng lúc đó sẽ khai thác giải pháp ứng dụng, còn việc bảo hành/bảo trì các phần riêng của giải pháp sẽ được thực hiện bởi các công ty khác nhau.

Do cơ chế cập nhật phần nhiều là được dựa trên việc mô tả toàn bộ giải pháp ứng dụng bằng metadata, còn lô-gíc nghiệp vụ của ứng dụng được xây dựng trên cơ sở các nguyên mẫu chuẩn của các đối tượng ứng dụng, cho nên hệ thống có khả năng hỗ trợ chiến lược cập nhật có tính đến đặc thù cấu trúc của giải pháp ứng dụng, có khả năng kiểm tra sự toàn vẹn lô-gíc và đáp ứng đầy đủ cho người sử dụng thông tin cần thiết về các thay đổi ở mức độ trực quan.

Và hơn nữa...

Toàn bộ giao diện của giải pháp ứng dụng có thể được hiển thị theo nhiều ngôn ngữ. Mỗi người sử dụng có thể làm việc với một trong số các ngôn ngữ đã được thiết lập trong đó. Khi đó, không cần phải tách các xâu ký tự nội dung vào các tệp riêng biệt. Các phần tử giao diện theo nhiều ngôn ngữ được soạn thảo "tại chỗ" như: trong hộp thoại, khuôn in chứng từ, menu... Trong mọi thời điểm, người lập trình có thể mở lên và soạn giao diện của các hộp thoại theo ngôn ngữ khác. Cũng được phép soạn thảo lần lượt các nhãn cụ thể trên các ngôn ngữ đã được cài đặt trong giao diện. Nếu cần phải mở rộng cơ sở ngôn ngữ, ta có thể nhóm tất cả các phần tử nội dung vào một danh sách chung để cho việc chuyển các nhãn cùng loại được thể hiện ngay trong tất cả các cấu phần của

giao diện. Chiến lược chung của việc quốc tế hóa còn bao gồm việc chuyển giao diện hệ thống, lưu các xâu ký tự theo định dạng UNICODE, định dạng ngày và số phù hợp với các đặc tính của từng nước và các ngôn ngữ khác nhau, xây dựng các trình tự sắp xếp dữ liệu có tính đến các tiêu chuẩn quốc gia...

Việc thực hiện các thuật toán lô-gíc nghiệp vụ có thể được người lập trình chuyển lên Server «1C:DOANH NGHIỆP» nếu thấy cần thiết. Điều này cho phép người lập trình quản lý việc phân tải giữa Client và Server. Khi đó không đòi hỏi người lập trình phải có kỹ năng xây dựng cấu trúc ba tầng, hay kiến thức về nghiệp vụ mạng... Hệ thống có hội tụ đủ mọi chi tiết công nghệ và bảo đảm việc sử dụng hợp lý các tài nguyên Server nhờ sự hỗ trợ của mô hình staterless, bộ nhớ trung gian (cach) và phân chia tài nguyên hệ thống...

Trong hệ thống bảo vệ quyền truy cập của «1C:DOANH NGHIỆP» có thực thi khả năng thiết lập các hạn chế đến mức từng bản ghi, và cả các trường của bảng. Điều đó cho phép, ví dụ, để bảo đảm cho người sử dụng có quyền chỉ truy cập đến tên của người lao động trong giới hạn bộ phận của mình, còn các dữ liệu về lương hay tình trạng gia đình - chỉ dành cho các cán bộ phụ trách trực tiếp. Việc thiết lập các hạn chế chỉ yêu cầu đưa ra điều kiện hình thức, sau đó hệ thống sẽ tự kiểm tra mọi hoạt động của người sử dụng, người lập trình không cần quan tâm đến các hạn chế này một cách rõ ràng khi thực thi các chế độ giao diện khác nhau và các thuật toán lô-gíc nghiệp vụ.

Có bảo đảm hai chế độ kiểm tra quyền truy cập. Tùy thuộc vào bài toán cần giải quyết, hệ thống kiểm tra quyền truy cập có thể dừng truy cập dữ liệu của người sử dụng hoặc đơn giản là ngắt bỏ các bản ghi đã được chọn, sau khi đã đảm bảo cung cấp đủ cho người sử dụng thông tin cần thiết mà họ được phép truy cập.

Một trong số các công cụ mạnh của người lập trình trong hệ thống «1C:DOANH NGHIỆP» là cơ chế so sánh và hợp nhất các giải pháp ứng dụng. Chúng ta cũng đã nhắc đến nó khi nói về các khả năng hỗ trợ, thế nhưng cơ chế này cũng có thể được sử dụng hiệu quả cả trong quá trình lập trình. Công cụ này có thể được sử dụng để phân tích các khác biệt cũng như để chuyển một phần các chức năng từ một ứng dụng này vào một ứng dụng khác. Nó bảo đảm thuận tiện trình bày bằng trực quan các khác biệt giữa các giải pháp ứng dụng, và có các khả năng linh hoạt để thiết lập việc so sánh lô-gíc. Khi so sánh các cấu hình cùng họ, cơ chế này sẽ tự động tiến hành so sánh, thậm chí so sánh cả các đối tượng đã bị đổi tên (bằng cách sử dụng cơ chế đồng nhất (ID) của metadata). Ta có thể thiết lập bằng tay tính tương ứng của các đối tượng được đem so sánh mà không phụ thuộc vào tên của chúng có trùng nhau hay không, sau đó nó sẽ được chú ý đến ở tất cả các tham chiếu lên các đối tượng có trong các đối tượng khác của giải pháp ứng dụng. Ngoài ra, cơ chế so sánh còn chứa các công cụ để hiển thị trực quan các khác biệt của các hộp thoại giao diện cũng như các khuôn in chứng từ. Nhờ sử dụng cấu trúc metadata và các công cụ so sánh trực quan các đối tượng giao diện, nền tảng «1C:DOANH NGHIỆP» đã giải quyết thành công nhiều nhiệm vụ hơn so với trong các hệ thống khác, nơi mà các cơ chế so sánh chỉ dựa trên kết quả so sánh của các tệp nguồn chương trình.

Trong «1C:DOANH NGHIỆP» có hỗ trợ việc lập biên bản ghi nhận thao tác của người sử dụng trong sổ nhật ký riêng. Hệ thống có nhiều mức ghi nhận, chúng có thể hoạt động tự động mà không cần đòi hỏi thêm công sức của người lập trình. Trước hết, dĩ nhiên, cần ghi nhận việc bắt đầu và kết thúc phiên làm việc của người sử dụng, các thao tác quản trị cơ sở thông tin, cũng như các lỗi. Do tất cả các thay đổi trong cơ sở dữ liệu được thực hiện chỉ trong kỹ thuật đối tượng (thông qua các đối tượng đáp ứng chức năng điều khiển dữ liệu), hệ thống có thể tự động ghi nhận lại trong nhật ký và sẽ được thực hiện không phụ thuộc vào việc chúng đã được thực hiện hay chưa (thực hiện trực tiếp hay sử dụng lệnh). Người lập trình cũng có thể đưa vào nhật ký một số nội dung khác bất kỳ mà vì một lý do nào đó hệ thống đã chưa đưa vào, ví dụ, thông tin về việc gửi fax hay về lập báo cáo. Cần lưu ý rằng, nhật ký sự kiện được thực thi như là một cơ chế hệ thống riêng biệt. Cơ chế này không sử dụng cơ sở dữ liệu của «1C:DOANH NGHIỆP» để lưu thông tin, cho nên nó không tạo thêm phụ tải lên hệ thống và không làm ảnh hưởng đến công việc của người sử dụng.

Tổng kết

Trong bài viết này, chúng tôi đã thử nêu ra một số vấn đề công nghệ mới đang được quan tâm trong nền tảng «1C:DOANH NGHIỆP». Chúng tôi cũng đã cố gắng chỉ ra rằng: tất cả các giải pháp được thực thi trong «1C:DOANH NGHIỆP» đều được dựa trên một mô hình và một cấu trúc thống nhất.

Điều quan trọng là tất cả các cơ chế của nền tảng, ở các mức độ khác nhau, đều sử dụng các tính năng của các cơ chế khác. Ví dụ, hầu như tất cả đều dựa trên cấu trúc metadata và hệ thống nguyên mẫu của các đối tượng ứng dụng. Như vậy, mặc dù mỗi một tính năng của «1C:DOANH NGHIỆP» đều rất có giá trị và ý nghĩa rất lớn, nhưng điều đáng quan tâm hơn cả chính là việc xem xét chúng một cách tổng thể trong một mô hình hoàn chỉnh.

Một lần nữa, chúng tôi xin tóm lược tổng thể một số các nguyên tắc chung có trong mô hình «1C:DOANH NGHIỆP»:

Cấu trúc của nền tảng cùng với các phương tiện công cụ là để sao cho người lập trình giải pháp ứng dụng có thể trừu tượng hóa tối đa, tách rời khỏi các vấn đề công nghệ mức thấp và tập trung giải quyết các nhiệm vụ trong lĩnh vực ứng dụng của mình.

Toàn bộ quá trình xây dựng giải pháp, bắt đầu từ việc xác định cấu trúc dữ liệu cho đến việc thiết kế các phần tử giao diện và kết nối các công cụ tích hợp đều được thực hiện trong cùng một hệ thống các khái niệm, và như vậy, cho phép tăng nhanh tốc độ đào tạo đội ngũ chuyên gia và nâng cao hiệu suất lao động.

Nền tảng có sẵn lời giải cho hầu hết các vấn đề nảy sinh khi lập trình giải pháp ứng dụng, từ khi thiết lập cách phản ánh các dữ liệu của lĩnh vực ứng dụng vào cơ sở thông tin, cho đến việc quy trình cài đặt, hỗ trợ và quản trị hệ thống. Như vậy, người lập trình không cần phải nghiên cứu các công nghệ khác nhau và giải quyết vấn đề tương thích khi sử dụng.

Việc có một mô hình thống nhất và xuyên suốt ở mức cao cùng với các công nghệ thực thi trong đó cho phép, theo chúng tôi, giảm được đáng kể (hàng chục lần) các chi phí tạo dựng và hỗ trợ các giải pháp ứng dụng.

Bắt đầu từ 1996, khi bắt đầu công việc theo hướng này, thậm chí, ngay cả sau khi cho ra đời phiên bản đầu tiên của nền tảng, chúng tôi cũng chưa hoàn toàn tin tưởng vào con đường mà mình đã chọn. Trong khoảng thời gian qua, mô hình ý tưởng của hệ thống đã được phát triển thành công, và ngày nay nó đã được minh chứng bằng nền tảng «1C:DOANH NGHIỆP 8.0». Nói một cách nghiêm túc, trên thế giới hiện nay cũng không có nhiều các công nghệ chuyên dụng định hướng đến việc xây dựng nhanh giải pháp ứng dụng. Tuy nhiên, càng ngày có thể thấy rõ rằng, con đường mà chúng tôi chọn được dựa trên định hướng chiến lược của việc phát triển phần mềm quản lý kinh tế.

Theo đánh giá của nhiều chuyên gia mà chúng tôi đã tiếp xúc, tại Nga và các nước thuộc Liên Xô cũ đã hình thành toàn diện một nền công nghiệp phát triển và hỗ trợ các giải pháp ứng dụng dựa trên nền tảng công nghệ chuyên dụng; và theo cách nhìn tổng thể thì tình hình ở đây quả thực là độc đáo. Theo khảo sát của chúng tôi, hiện chưa có một nền tảng chuyên dụng tương tự mà trên đó rất nhiều các công ty khác nhau tạo ra một số lượng lớn các sản phẩm ứng dụng đại trà, đáp ứng được nhu cầu cho một khu vực rộng lớn. Cho nên, đối với nhiều chuyên gia, cách tiếp cận và giải pháp của chúng tôi luôn được quan tâm đặc biệt, bởi vì hiệu quả của việc triển khai đã được đánh giá rất cao trên thực tế.

Dĩ nhiên, tất cả những hình dung trên (bao gồm cả việc phân tích các xu thế hiện đại) chỉ là quan điểm và kết quả khảo sát của chúng tôi. Chúng tôi không hề kỳ vọng đạt đến một chân lý tuyệt đối, mà có lẽ, không hề tồn tại trong các vấn đề được nêu ra. Có lẽ, có rất nhiều bạn đọc có cách nhìn nhận vấn đề khác với chúng tôi. Chúng tôi sẽ luôn chân thành cảm ơn về các nhận xét góp ý (kể cả những phê phán gay gắt) cho bài viết này.

Dịch từ nguyên bản tiếng Nga: Trần Quang Huy, Trần Thân, Ngô Đình Hoàn, Trần Thắng.

Công ty cổ phần Hệ thống 1-V (1VS JSC).

Phòng 1508, tòa nhà Thành Công, 25 Láng Hạ, Ba Đình, Hà Nội

www.1vs.vn

1vs@1vs.vn