White Paper

Combining Information Engineering and Object Orientation with AllFusion® Plex

William A. Hunt
Product Manager, CA
October 2006

AllFusion® Plex

AllFusion Plex from CA is an Architected Rapid Application Development (ARAD) tool that is model-based and supplied with pre-built architectural and design patterns. This is a common methodology among ARAD tools such as AllFusion Plex, originally developed in 1994 as Synon Obsydian.

Studies have indicated that ARAD tools generate significantly higher levels of productivity and flexibility, as opposed to standard RAD tools or enterprise modeling products. ARAD tools also enable developers skilled in RPG or COBOL to become productive with a minimal amount of additional training. Applications are developed in a Microsoft Windows environment and compiled and tested in the target environment. The latter may be the IBM iSeries, Windows or Java. A wide range of application types can be developed, including client-server, Web-based, character-based, batch and wireless device-based. Additionally, SOAP/XML service components can be developed.

AllFusion Plex takes two very powerful methods and mixes them to get the best of both worlds. The first is Information Engineering (IE), where techniques such as Entity Relationship Modeling are used to drive development from a data perspective.

The second is Object Orientation (OO). Application development with AllFusion Plex typically starts with creating an IE-type data model and applying OO-based techniques such as abstraction and inheritance, and modularity and hierarchy.

AllFusion Plex offers:

- A Windows-based visual IDE, complete with GUI screen designers, a language-neutral action language editor, a diagrammer and impact analysis tools;
- A multi-developer repository with built-in configuration management for storing design models across multiple versions, languages and platforms;
- Code generators that automatically create 100% of the native code required, together with HTML and GUI clients, 5250 host screens, server programs and database objects.

AllFusion Plex includes hundreds of reusable business objects called *patterns*, grouped into libraries. Many additional pattern libraries are available from a number of CA Partners. Customers may also customize these patterns to create a standards layer, or they can create their own.

This white paper describes the *primary* functionality and features offered by AllFusion Plex. This is achieved by stepping through a simplified example of the development of an application. In this case, we have used a simple 'Customer, Account and Account Items' scenario.

Modeling for Construction

The application development process using AllFusion Plex is essentially a three-step process - involving Data Modeling, Pattern Matching and Customization - followed by automatic code generation.

The developer will follow a path of iteration through these three steps, using an Architected RAD process to achieve the desired application. AllFusion Plex uses the concept of a model to hold the application design. The model is held within a repository containing both the model as well as all reusable objects.

Data Modeling

AllFusion Plex is used to capture the Entity Types, Attributes and Relationships of the application Data Model. Figure 1 displays the result using the Entity Relationship Diagrammer. A range of diagrams is available for both capturing and displaying design information held in the model. Inset is the Object Browser used to navigate through all objects held within the repository.

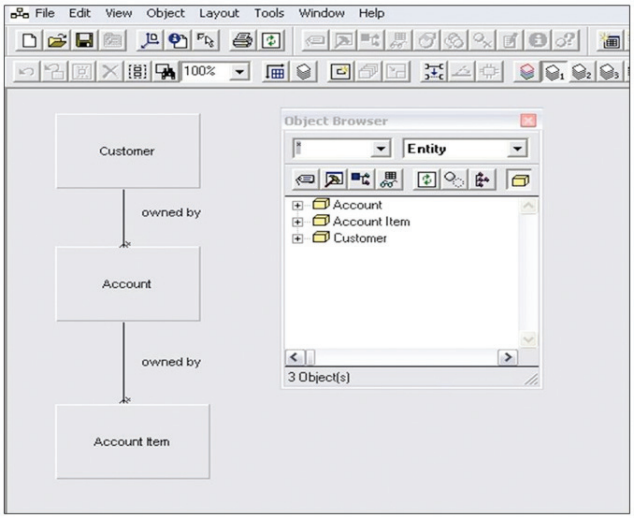

Figure 1. The AllFusion Plex Entity Relationship Diagrammer. Design models can be viewed as a physical diagram or within an Object Browser.

The structure of an AllFusion Plex application, reflecting the Data Model design but also the structure of other aspects of the application functionality (including functions, components and panels) can be captured using a set of statements defined in a Subject, Verb, Object form. This set is known as a Triple. They are captured within a part of the toolset known as the Model Editor.

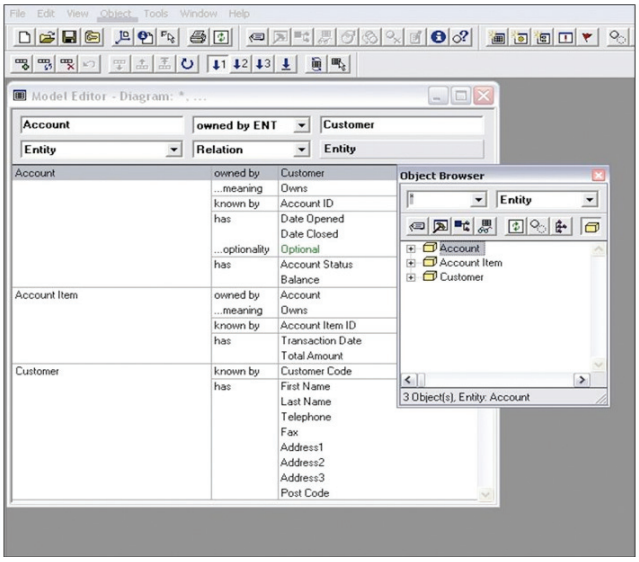

Figure 2. AllFusion Plex 'triples' are designed and defined in the Model Editor.

Pattern Matching and Inheritance

The second part of the development process is to consider the functionality that the application is required to deliver. Examples of this could be Use Cases or the product of a process modeling tool. The idea is to match the functionality required against a set of Design Patterns held within the repository. CA supplies a large set of these Patterns, and part of the power of AllFusion Plex lies in constructing your own set of Patterns to reflect the types of functionality specific to your organization.

Inheritance is the mechanism by which an object or set of objects can acquire the characteristics or design of another object or set of objects. This is the means by which Patterns are implemented in the application design and is always expressed using an 'is a' verb in a triple. Multiple inheritance is a key part of the way applications are developed in AllFusion Plex.

Figure 3 expresses the way that an entity, representing a Customer, has rapidly acquired simple functionality. The entity acquires:

- A table to hold the data about the Customer
- Views of the data
- Access functions to retrieve and update the table
- A suite of user functions to enable the display and capture of Customer data

Note the use of the 'is a' statement.

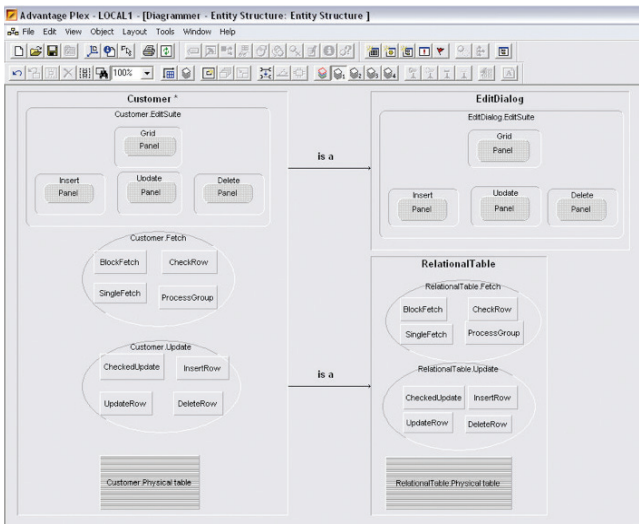

Figure 3. An entity acquires additional functionality using multiple inheritance.

Customization

The third phase of constructing the application is to customize the implemented Patterns. If we think of the Patterns as classes of behavior, held as design information, then our implemented Patterns need to be sub-classed in order to suit the application requirements. Examples of this could be overriding elements of the Pattern structure, replacing part of the pattern design with our own design or adding logic to the application to add behavior to the design. Still further customization involves user interface design or construction of new functionality.

AllFusion Plex contains an action language editor (or Action Diagrammer) where logic can be added or changed within the application design. Essentially, each function will have logic expressed as procedural statements held within an action diagram. Most of this logic will come from inherited design and is shown in grey. Appropriate points are made available for the developer to insert statements. Statements are explicitly type checked to ensure no syntax errors occur. Variables are displayed, as are the available statements and syntax for the developer to choose from.

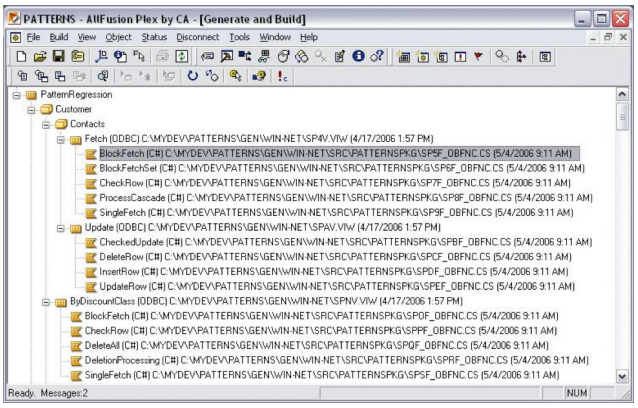

Figure 4. The Generate and Build screen, displaying a complete list of objects contained in a model along with the targeted programming language.

Application Generation

Once the developer has passed through one iteration of the preceding three steps, the application can be built for testing.

Figure 4 shows the Generate and Build screen. The developer selects the objects to be built, generates the code and submits the build, either to the local machine or to a server.

Application Integration and Reverse Engineering

New applications are rarely built in isolation. Typically a new application must reuse or integrate with existing applications and packages. AllFusion Plex provides a rich set of application integration and reverse engineering capabilities including:

- Database schema import to allow new applications to be built upon existing databases.
- COM, .NET and EJB Connectors that allow AllFusion Plex-generated applications to be consumed by external COM-compatible environments such as Visual Basic.
- COM Import capability that allows AllFusion Plex developers to consume existing third-party COM components using the native action language.
- Special capabilities to easily call existing OS/400 programs.
- Ability to embed hand-coded 3GL code within the generated code without compromising platform independence. This allows the developer to accomplish any programming tasks not directly supported by the native action language, including calling of Windows programs written with other development tools.
- Model XML Import/Export capability that allows meta-data to be interchanged with other development tools such as AllFusion ERwin.

.NET Integration and C# Code Generation

An AllFusion Plex "Connector" exposes business logic to third party applications. The tool currently features COM and .NET connectors, which provide the ability to produce components that are useable in the .NET world. Using Visual Studio these components can easily be turned into web services.

AllFusion Plex r6 features C# server code generation based over a 100% managed code .NET runtime. Essentially this makes AllFusion Plex the only ARAD tool available to provide complete support for IBM System i shops to have the flexibility to target the .NET platform for their applications using one set of development techniques.

AllFusion Plex will target the .NET Framework 2.0 and use MSBUILD, Microsoft's new task- based build to provide compatibility with Visual Studio 2005. The generated applications will use OLE DB to provide access to a wide range of databases including Microsoft SQL Server 2005, Oracle 10g and IBM DB2. Additionally, a feature tying in closely with the C# build process is application packaging support. Using this, developers will be able to model and build .NET Assemblies. The same approach will also be used to model application packages for the other platforms supported by Plex - JAR files for the Java platform and, later, ILE Service Programs for i5/OS.

Summary

AllFusion Plex has provided a true application modernization solution for System i shops developers for years - for thousands of developers and thousands of critical business applications. The tool continues to evolve with overall trends, and CA is working with Microsoft in a partnership capacity to ensure that AllFusion Plex evolves along with Microsoft's key technologies. AllFusion Plex uses Visual Studio .NET for C++ builds, and also long featured the ability to deliver MS SQL Server applications.

CA and its partners also recognize the need to bring existing applications into a modern development environment to enable the IT department to respond quicker to changes in the business and regulatory matters. RPG and COBOL applications can be brought into AllFusion Plex to enable developers to take advantage of multiplatform requirements. AllFusion Plex develops for many platforms with one skill set and insulates developers from changes in technologies.

CA's .NET application development support in the AllFusion Plex tool provides not only a powerful and proven development environment, but a tool that truly defines the goals of the Microsoft Midrange Alliance Program - .NET is provided as a choice for the delivery of modernized applications without sacrificing the continued use of or numerous benefits of the System i Server.

For more information, please visit www.ca.com

About the Author

William A. Hunt has served as Product Manager for AllFusion Plex at CA since 2000.

MP308651006