

УДК 519.7

А.А. Мамедов

Харьковский национальный университет радиоэлектроники, Харьков

МОДЕЛЬ ИНФОРМАЦИОННОГО СОПРЯЖЕНИЯ ПРОГРАММНЫХ ОБЪЕКТОВ

Рассматривается модель информационного сопряжения двух или нескольких программных объектов, под которым понимается процесс преобразования множества их общих данных к форме, согласующейся с представлением структуры каждого из объектов.

Ключевые слова: сборочное программирование, информационное сопряжение, программный объект, программный элемент.

Введение

Развитие сборочного, объектно-ориентированного и других методов программирования идет по пути создания разного типа моделей, как одного из способов определения разных сторон реализуемой предметной области и их трансформации к конечному результату. Приведем набор моделей, возникших в объектно-ориентированном программировании (ООП) и в других видах программирования:

– объектная модель, модель архитектуры (статическая и динамическая), модель окружения (взаимодействия со средой) и использования [1, 2];

– модели метода Шлеера и Меллора (информационная модель, модель поведения системы, модель процессов) [3];

– модели UML (use case), используемые для описания требований, структуры системы и ее преобразование в исполняемый код [4];

– модели процессов разработки ПС в RUP [5] и др.;

– модель ПрО;

– модельно-управляемая разработка – MDD (Model Driven Development), ориентированная на непосредственный перевод описания модели в исполняемый код [6];

– модельно-управляемая архитектура – MDA (Model Driven Architecture) для создания ПС на основе построенных моделей, которые транслируются в конкретную реализацию системы [6].

Объектная модель – одна из первых моделей ООП, получившая реализацию в разных инструментальных системах (CORBA, RUP и др.) В ООП процессы проектирования выполняют формирование моделей на каждом процессе.

Данная модель создается в процессе анализа предметной области, объекты которой определяют реальные ее сущности и операции над ними. В процессе проектирования эта модель дополняется требованиями и функциями, которые программируются средствами C++, Java и др., и передаются заказчику для решения задач ПрО, поиска ошибок и внесения

изменений как в состав объектов, так и в методы их реализации.

Средствами ООП создаются: модель ПрО, модель архитектуры, модель окружения и использования. Эти модели воплощаются в программный продукт, который реализует связи между объектами, набор операций и состояний, порождаемых взаимосвязями объектов в модели окружения.

Модель архитектуры включает в себя:

– статическую модель описания структуру системы в терминах классов объектов и взаимоотношений (обобщения, расширения, использования и др.) между ними;

– динамическую модель, которая определяет взаимодействие между объектами системы во время выполнения, инициируется запросами к сервисам объектов и реакцией после их выполнения.

К статической модели относится модель окружения, а к динамической – модель использования. Обе модели взаимно дополняют друг друга через модель связи со средой.

Разработка модели информационного сопряжения

Модель – это математическое или другое представление системы, процессов ее изготовления из готовых объектов (путем их выбора, определения интерфейсов, интеграции и др.) и управления этими процессами. В сборочном программировании – это модель сопряжения программных объектов и модель управления ими. Указанные модели названы основными, так как проблемы передачи данных и управления – главные задачами при объединении любых программных объектов с применением соответствующих методов сборочного программирования и инструментальных средств.

В объектно-ориентированном программировании сформировались другие модели: объектная модель, информационная модель, модель состояний, модель процессов и др. Общее, что их объединяет, – это обеспечение интерфейса объектов при создании из них ПС.

Под информационным сопряжением двух или нескольких программных объектов понимается процесс преобразования множества их общих данных к форме, согласующейся с представлением структуры каждого из объектов.

В дальнейшем, не снижая общности процесса анализа, рассмотрим информационное сопряжение только пары разных объектов.

Модель информационного сопряжения – это совокупность формальных описаний данных взаимодействующих объектов и функций их преобразования к релевантному виду.

Процесс сборки объектов в интегрированный комплекс (ИК) – это создание сложного программного объекта путем объединения готовых более простых программных элементов.

Пусть $P = \{p^i\}_{i=1, \bar{s}}$ – множество программных компонентов, входящих в состав создаваемого ИК. С каждым p^i связано множество данных D^i , с помощью которых осуществляется информационный обмен между интегрируемыми компонентами.

Множество $D = \{d_j^i\}_{j=1, \bar{t}}$ состоит из переменных d_j^i , каждая из которых характеризуется тройкой: именем (идентификатором переменной) N_j^i , типом T_j^i и текущим значением V_j^i).

Рассмотрим два программных компонента p^i и p^k (p^k выполняется после p^i) с множествами данных D^i и D^k соответственно. В общем случае в D^i и D^k могут входить переменные, общие для p^i и p^k с точки зрения их семантической обработки. Эти переменные образуют подмножества \tilde{D}^i и \tilde{D}^k . Задача информационного сопряжения состоит в преобразовании подмножества данных \tilde{D}^i в представление, согласующееся с \tilde{D}^k .

Введем следующие обозначения:

$$N^i = \{N_j^i\}_{j=1, \bar{t}}^{-i},$$

$$T^i = \{T_j^i\}_{j=1, \bar{t}}^{-i},$$

$$V^i = \{V_j^i\}_{j=1, \bar{t}}^{-i}.$$

В \tilde{D}^i им соответствуют множества из троек – N^i , T^i и V^i .

В общем случае для преобразования множества данных D^i необходимо построить преобразование для этих имен, \tilde{N}^i , \tilde{T}^i и \tilde{V}^i .

Имеют место следующие два случая.

1. Каждой переменной $d_j^i \in \tilde{D}^i$ соответствует только одна переменная $d_j^k \in \tilde{D}^k$. Тогда преобразование (отображение)

$$F^{ik} : \tilde{D}^i \rightarrow \tilde{D}^k \quad (1)$$

состоит из множества преобразований для отдельных переменных: $F^{ik} = \{F_{ji}^{ik}\}$.

При этом формально

$$F_{ji}^{ik} = \{FN_{ji}^{ik}, FT_{ji}^{ik}, FV_{ji}^{ik}\}.$$

Вводя обозначения

$$FN_{ji}^{ik} = \{FN_{ji}^{ik}\},$$

$$FT_{ji}^{ik} = \{FT_{ji}^{ik}\},$$

$$FV_{ji}^{ik} = \{FV_{ji}^{ik}\},$$

определяем преобразования:

$$FN^{ik} : \tilde{N}^i \rightarrow \tilde{N}^k,$$

$$FT^{ik} : \tilde{T}^i \rightarrow \tilde{T}^k, \quad (2)$$

$$FV^{ik} : \tilde{V}^i \rightarrow \tilde{V}^k$$

соответственно для множеств идентификаторов, типов данных и значений.

2. Между переменными d_j^i и d_j^k не существует однозначного соответствия. Это тогда, когда несколько элементов из \tilde{D}^i соответствуют одному элементу из \tilde{D}^k и наоборот. Сложная связь, при которой несколько элементов из \tilde{D}^i соответствуют нескольким элементам из \tilde{D}^k , в практике сборочного программирования, как правило, отсутствует, что связано с отдельной разработкой отдельных программных компонентов.

Соответствие нескольких переменных одной и наоборот свидетельствует об изменении уровня структурирования данных. Пусть \tilde{d}_j^i соответствует несколько элементов из \tilde{D}^k . Обозначим их через $\tilde{d}_{j1}^k, \dots, \tilde{d}_{jr}^k$, а через S – функцию селектора, снижающую уровень структурирования данных:

$$S(\tilde{d}_j^i) = (\tilde{d}_{j1}^k, \dots, \tilde{d}_{jr}^k), \quad (3)$$

где каждому \tilde{d}_{jv}^k соответствует \tilde{d}_{jv}^i , при $v = \overline{1, r}$. Замещая \tilde{d}_j^i в \tilde{D}^i элементами $\tilde{d}_{j1}^i, \dots, \tilde{d}_{jr}^i$ получаем множество $\tilde{\tilde{D}}^i$.

Построение отображения

$$F^{ik} : \tilde{\tilde{D}}^i \rightarrow \tilde{D}^k$$

производится аналогично, как в случае 1.

При соответствии нескольких элементов из $\tilde{\tilde{D}}^i$ одному элементу из \tilde{D}^k поступаем следующим образом. Вместо функции селектора вводим функцию конструирования вида

$$C(\tilde{d}_{j1}^i, \dots, \tilde{d}_{jr}^i) = \tilde{d}_j^k, \quad (4)$$

где \tilde{d}_j^i соответствует единственному элементу из \tilde{D}^k . Модифицируя элементы множества \tilde{D}^i и рассматривая отображение $F^{ik} : \tilde{D}^i \rightarrow \tilde{D}^k$, приходим к аналогичному результату.

Проведем анализ отображений FN, FT и FV (индексы для простоты опущены). Из построения следует, что множества \tilde{N}^i и \tilde{N}^k содержат одинаковое количество элементов. Поэтому FN только переупорядочивает идентификаторы переменных в соответствии с последовательностью, принятой при описании программного компонента p^k .

Отображение преобразования множества типов данных FT более сложное, оно связано с наличием практически неограниченного количества типов. По определению тип данных характеризуется парой

$$T = (X, \Omega),$$

где X – множество значений, которые могут принимать переменные рассматриваемого типа, Ω – множество операций, выполняемых над этими переменными. T можно рассматривать как алгебраическую систему. В ней преобразование типа $T_j^i = (X_j^i, \Omega_j^i)$, в тип $T_j^k = (X_j^k, \Omega_j^k)$, соответствует преобразованию множества значений X_j^i в X_j^k , при котором семантическое содержание операций из Ω_j^i эквивалентно операциям из Ω_j^k .

В общем случае преобразование T_j^i в T_j^k может быть односторонним. Однако для повторного использования данных, что характерно для многократного вызова программных компонентов, обрабатывающих одни и те же структуры данных, требуется и прямое и обратное преобразования. Для достижения этого необходимо, чтобы отображение между T_j^i и T_j^k было изоморфизмом. Иными словами, построению преобразования между двумя типами данных будет соответствовать нахождению изоморфного отображения между двумя алгебраиче-

скими системами. При практической реализации модель информационного сопряжения целесообразно рассматривать как совокупность моделей для пар программных компонентов $P = \{P^{ik}\}$ в создаваемой программе.

Модель для каждой из них имеет вид

$$M^{ik} = (\tilde{N}^i, \tilde{T}^i, \tilde{V}^i, N^k, T^k, V^k, FN^{ik}, FT^{ik}, FV^{ik}). \quad (5)$$

Отдельные составляющие данной модели и принципы их построения описаны выше.

Выводы

В данной статье представлен материал, связанный с проблемами объединения (комплексирования, интеграции) модулей в более сложные программные образования. Основу процесса объединения составляют модели сборочного программирования. Показан новый подход к разработке программных систем – модельный.

Список литературы

1. Буч Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, А. Джекобсон; пер. с англ. – М.: ДМК, 2000. – 432 с.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Г. Буч. – М.: "Издательство Бином", СПб.: "Невский диалект", 2001. – 560 с.
3. Шлеер С. Объектно-ориентированный анализ: моделирование мира в состояниях / С. Шлеер, С. Меллор. – К.: Диалектика, 1993. – 238 с.
4. Буч Г. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, А. Джекобсон; пер. с англ. – М.: ДМК, 2000. – 432 с.
5. Лаврищева Е.М. Сборочное программирование / Е.М. Лаврищева, В.М. Гриценко. – К.: Наук. думка, 1991. – 213 с.
6. Лаврищева К.М. Програмна інженерія: підручн. / К.М. Лаврищева. – К.: Академперіодика, 2008. – 319 с.

Поступила в редколлегию 11.03.2016

Рецензент: д-р техн. наук, проф. С.Ф. Чалый, Харьковский национальный университет радиоэлектроники, Харьков.

МОДЕЛЬ ІНФОРМАЦІЙНОГО СПОЛУЧЕННЯ ПРОГРАМНИХ ОБ'ЄКТІВ

А.О. Мамедов

Розглядається модель інформаційного сполучення двох або декількох програмних об'єктів, під яким розуміється процес перетворення множини їх загальних даних до форми, що узгоджується з представленням структури кожного з об'єктів. Основу процесу об'єднання складають моделі складального програмування. Показаний новий підхід до розробки програмних систем – модельний.

Ключові слова: складальне програмування, інформаційне сполучення, програмний об'єкт, програмний елемент.

INFORMATION MODEL INTERFACE SOFTWARE OBJECTS

A.O. Mamedov

A model of the information interface of two or more software objects, which refers to the process of converting a plurality of shared data to a form that is consistent with the presentation of the structure of each of the objects. The basis of the integration process up assembly programming model. It showed a new approach to the development of software systems – model.

Keywords: assembly programming, pairing information, software object, program element.