электронное научно-техническое издание
НАУКА и ОБРАЗОВАНИЕ
Эл № ФС 77 - 30569. Государственная регистрация №0421100025. ISSN 1994-0408

Свойство однозначности построения алгоритма при проектировании мультиаспектных информационных систем

77-30569/242870

№ 10, октябрь 2011
автор: Виноградова М. В.

УДК 004.415.2

МГТУ им. Н.Э. Баумана
vinogradova.m@gmail.com

Введение

В настоящее время существует множество информационных систем организационного управления, используемых в небольших организациях и подразделениях. Например, АСУ кафедры, деканата и прочих подразделений вуза, а также автоматизированные системы небольших агентств и организаций [1]. Несмотря на разнообразие средств реализации и областей применения, эти системы имеют одинаковую архитектуру и множество сходных черт, что позволяет выделить их в отдельный класс систем. Указанные системы имеют следующие особенности:

- единая база данных и пользовательский интерфейс к ней для выполнения функциональных задач,

- множество пользователей, имеющих разнообразные служебные обязанности и выполняющие пересекающиеся наборы функций,

- необходимость в постоянной модернизации системы: добавлении или изменении функциональных задач пользователей.

Поскольку происходит постоянное пополнение набора функциональных задач, решаемых системой, то процесс модернизации становится частью процесса эксплуатации системы. В данное время модернизация подобных систем требует привлечения группы разработчиков: проектировщиков, специалистов по базам данных и программистов. Это приводит к большим трудозатратам на модернизацию и согласование отдельных компонентов системы. Для повышения эффективности функционирования систем

необходимо сократить трудозатраты и связанные с ними затраты времени на модернизацию и наращивание.

Методы сокращения трудозатрат

Возможно несколько вариантов сокращения трудозатрат. Первый – это использование готовых решений с настройками под конкретную область применения и под конкретного пользователя, подобных ERP-системам. Недостатком использования подобной технологии является навязывание бизнес-процессов. Исходная структура ERP-системы разрабатывается на основе реальных процессов управления, существующих в некоторой сфере деятельности. Для использования ERP-системы в другой организации необходимо установить соответствие между процессами управления организации, в которой происходит внедрение, и процессами управлениями, на основе которых разрабатывалась ERP-система. В данном случае это решение не будет эффективным из-за отсутствия типового набора прикладных задач и больших различий в бизнес-процессах, существующих в организациях.

Вторым вариантом сокращения трудозатрат на разработку и последующую модификацию является декомпозиция всей системы на небольшие и несложные подсистемы, каждая из которых разрабатывается независимо [4, 6]. Возможны различные способы декомпозиции [2]. Система может быть декомпозирована различными методами на подсистемы, модули, компоненты или объекты. Используемый метод декомпозиции зависит от используемого подхода к разработке системы, от метода проектирования, от особенностей предметной области и от требования к независимости элементов системы.

В некоторых случаях стандартные методы декомпозиции не могут быть применены или их применение будет малоэффективным. Такое возможно, если не получается выделить отдельные элементы ИС из-за большой зависимости между ними. То есть, мера сцепления между выделяемыми в процессе разработки модулями или объектами недопустимо высока. Зависимость между объектами приводит к тому, что изменения, внесенные в один элемент системы, необходимо отрабатывать и на других элементах. Это приводит к большим затратам и при итерациях проектирования, и при модернизации ИС на этапе эксплуатации [5]. К числу подобных систем относятся рассмотренные выше информационные системы организационного управления.

Поскольку данные системы нельзя эффективно с точки зрения затрат декомпозировать на подсистемы, модули или объекты [9], то в качестве альтернативы стандартным методам декомпозиции следует рассмотреть аспектный подход к проектированию информационных систем

Аспектный подход к проектированию

Рассмотренные выше информационные системы можно назвать мультиаспектными, поскольку каждая из них должна предоставлять пользователям возможность работать в индивидуальных аспектах. Под аспектом понимается набор функциональных задач, соответствующий служебным обязанностям пользователя. При этом информационная система должна иметь единую базу данных и не содержать избыточных программных модулей. Со структурной точки зрения мультиаспектную систему можно представить как систему, сильносвязанную по данным и функциям [6], с настройкой под конкретного пользователя. Для разработки мультиаспектных систем предлагается использовать аспектный подход к проектированию.

Основной идеей аспектного подхода является разработка методики проектирования, при которой результат проектирования не зависит от конкретного разработчика. Описания задач, полученные в результате проектирования, будут полностью идентичны, если исходные данные для проектирования совпадали. Это позволит избежать процесса согласования программных компонентов, а также затрат на разработку межмодульного интерфейса. Достижение идентичности описаний достигается за счет жесткой формализации процесса проектирования информационной системы.

Аспект пользователя определяется набором функциональных задач. Задача - это действие по вводу, выводу или преобразованию данных. Под данными понимается блок из связанных между собой информационных элементов, который может быть ассоциирован с документом на бумажном носителе. По мере эксплуатации информационной системы появляются новые задачи и изменяются требования к существующим. Добавление новых задач должно происходить без изменения существующих и без разработки интерфейса с существующими компонентами. Изменения, вносимые в одну из задач, не должны приводить к необходимости изменять другие задачи. Поскольку программные реализации задач должны быть независимы друг от друга, то при создании (и модернизации) информационной системы должна выполняться декомпозиция на задачи. Аспектный подход предоставляет возможность независимой разработки функциональных задач. Выделение задач выполняется на основе функциональных потребностей пользователя без ограничений на состав входных и выходных данных и на алгоритм выполнения задачи.

При проектировании задачи составляется формализованное описание задачи на основе сведений о предметной области. Описание задачи состоит из описания структуры данных и описания алгоритма [4]. Общим в описаниях алгоритма и описаниях структуры

данных являются только названия используемых переменных. Описание структуры данных задачи – это формальное описание переменных, используемых для выполнения задачи, и их характеристик в терминах некоторой формальной модели [3]. Исходными данными для построения описания структуры данных являются сведения об объектах предметной области, их связях и реквизитах. Описание алгоритма задачи – это формальное описание функциональных преобразований, необходимых для получения результата выполнения задачи на основе множества входных данных. Исходными данными для построения описания алгоритма задачи являются сведения о входных и выходных данных задачи, а также их алгоритмических зависимостях.

Идентичность описаний задач означает, что на основе одних и тех же сведениях о предметной области, при независимом проектировании одинаковых задач будут получены совершенно идентичные описания структур данных и совершенно идентичные описания алгоритмов. Для получения идентичных описаний при независимой разработке должны выполняться два условия: построение структуры данных и построение алгоритма задачи должны быть однозначным.

Свойство однозначности построения алгоритма

Дадим формальное определение свойства однозначности построения алгоритма и сформулируем требования его достижения.

Задача в аспектном подходе определена как действие ввода, вывода или преобразования данных. Перечисленные действия можно обобщить как преобразование множества входных данных в выходные данные (результат). Описание алгоритма выполнения задачи должно содержать описание входных, промежуточных и выходных данных, перечень операторов, выполняемых над входными и промежуточными данными, а также задавать последовательность применения операторов. Исходными данными для описания алгоритма являются сведения об объектах предметной области, их реквизитах и связях, а также о получаемом результате и правилах его вычисления. Поскольку при описании алгоритма необходимо использовать данные, которые определены в описании структуры данных, то в качестве входных и выходных данных целесообразно использовать атомарные реквизиты. Атомарные реквизиты могут быть связаны не только функциональными, но и алгоритмическими зависимостями. Функциональные зависимости между реквизитами являются следствием зависимостей между объектами и их характеристиками. Алгоритмические зависимости задают функциональное преобразование между реквизитами. Результирующий реквизит алгоритмической зависимости будет вычислен из определяющих реквизитов по некоторой формуле.

Формула может быть задана в виде математического выражения, логического предиката или в другой форме представления функции преобразования. Функциональную зависимость можно рассматривать как частный случай алгоритмической зависимости.

Рассмотрим построение алгоритма задачи. Пусть для задач Z_i и Z_j разработаны описания алгоритмов $S_i = p^{RF\Phi}(R^i_{in}, R^i_{out}, R^i, F^i, \Phi^i)$ и $S_j = p^{RF\Phi}(R^j_{in}, R^j_{out}, R^j, F^j, \Phi^j)$ соответственно. Исходными данными для построения алгоритмов являются множества входных $R^i_{in} \subseteq R^i, R^j_{in} \subseteq R^j$ и выходных $R^i_{out} \subseteq R^i, R^j_{out} \subseteq R^j$ реквизитов, множества функциональных зависимостей (F^i и F^j) и множества алгоритмических зависимостей реквизитов (Φ^i и Φ^j), определенных на множествах реквизитов R^i и R^j. Описания алгоритмов составляются на некотором языке (нотации) с помощью функции преобразования $p^{RF\Phi}$ множества реквизитов и множества зависимостей в описание алгоритма.

Процесс разработки информационной системы обладает свойством однозначности построения алгоритма, если при независимой разработке задач информационной системы будут получены одинаковые описания их алгоритмов при условии совпадения входных и выходных данных:

$$((R^i_{in} = R^j_{in}) \wedge (R^i_{out} = R^j_{out})) => (S_i \equiv S_j).$$

Однозначность построения алгоритмов означает, что при совпадении входных и выходных данных для некоторых задач описания алгоритмов этих задач будут полностью идентичны при независимой разработке.

Описания одинаковых задач (или одной и той же задачи), полученные при независимой разработке, могут различаться по нескольким причинам. Во-первых, для выполнения одной и той же функциональной задачи может существовать множество алгоритмов. Во-вторых, для выполнения задачи могут быть использованы различные множества промежуточных данных. В-третьих, отсутствует формальные правила декомпозиции задачи на операторы. На этапе программной реализации декомпозиция на операторы выполняется на основе библиотеки функций, а на этапе проектирования декомпозиция выполняется по усмотрению проектировщика, что приводит к неоднозначности определения операторов.

Для достижения однозначности определения входных, выходных и промежуточных данных необходима однозначность построения структур данных. Если при построении алгоритма будет выполняться свойство однозначности построения структур данных, то для описания одинаковых задач будут использоваться идентичные описания переменных. Неоднозначность составления алгоритма и неоднозначность декомпозиции на операторы связаны друг с другом, поскольку алгоритм задается как последовательность выполнения операторов. Однозначности декомпозиции на операторы можно достичь, если выделять операторы на основе алгоритмических зависимостей. Если оператор определить как реализацию алгоритмической зависимости между выходным реквизитом и множеством входных реквизитов, то для описания одинаковых алгоритмических зависимостей всегда будут использоваться одинаковые операторы. В этом случае оператор будет определяться набором используемых переменных и семантикой выполняемого преобразования. Конкретная реализация оператора не имеет значения на этапе проектирования задачи, поскольку при переходе от описания задачи к программным компонентам оператор может быть заменен любой процедурой, выполняющей алгоритмическую зависимость оператора. Для того, чтобы оператор точно соответствовал алгоритмической зависимости, результатом его выполнения должен быть только один реквизит, и на вход оператора должны поступать только те реквизиты, которые необходимы для его выполнения. Из определения метрик связности модулей следует, что процедура, реализующая подобный оператор, будет обладать функциональной связностью. Функциональная связность позволяет избежать избыточности алгоритма.

Для описания алгоритма задачи необходимо указать множество реквизитов и множество операторов, а также установить последовательность выполнения операторов. Если последовательность выполнения операторов будет определяться готовностью их операндов, то необходимость определения этой последовательности проектировщиком исчезнет, что позволит избежать неоднозначности. Следовательно, для описания алгоритма следует применять модель вычислений, управляемых потоками данных [4, 6]. Эта модель задает способ «программирования, управляемого готовностью данных», при котором оператор выполняется, если все его операнды определены и имеют некоторые значения. Для достижения однозначности построения алгоритма необходимо, чтобы между множеством входных данных и результатом задачи существовала единственная последовательность операторов. Если при проектировании задачи не будут учитываться все алгоритмические зависимости, существующие между промежуточными данными, то

возможна ситуация, когда для одной задачи будут составлены различные описания. Следовательно, необходима полнота определения алгоритмических зависимостей.

Лемма об однозначности построения алгоритма

Докажем лемму об однозначности построения алгоритма.

Свойство однозначности построения алгоритма будет достигнуто, если:

1) выполняется свойство однозначности построения структур данных;
2) описание алгоритма составляется как последовательность выполнения операторов, управляемая готовностью данных;
3) операторы выделяются на основе алгоритмических зависимостей и обладают функциональной связностью;
4) между множеством входных данных и результатом существует единственная последовательность операторов;
5) при построении алгоритма учитываются все алгоритмические зависимости между входными, выходными и промежуточными данными.

Доказательство: Рассмотрим построение описаний двух задач независимыми разработчиками. Пусть получены два описания: $S_i = p^{RF\Phi}(R^i_{in}, R^i_{out}, R^i, F^i, \Phi^i)$ и $S_j = p^{RF\Phi}(R^j_{in}, R^j_{out}, R^j, F^j, \Phi^j)$. Докажем, что $((R^i_{in} = R^j_{in}) \wedge (R^i_{out} = R^j_{out})) \supset (S_i \equiv S_j)$, при условии выполнения требований, перечисленных в лемме. Из однозначности построения структур данных следует, что если $((R^i_{in} = R^j_{in}) \wedge (R^i_{out} = R^j_{out}))$, то при построении структуры данных задач будут учтены все функциональные зависимости, связывающие входные и выходные реквизиты, и все реквизиты предметной области, необходимые для формирования этих зависимостей. Следовательно, множества реквизитов, используемых при построении алгоритма, будут совпадать: $R^i \equiv R^j$. Также из однозначности построения структуры данных следует, что на основе совпадающих множеств реквизитов будут получены совпадающие множества структур данных, которые при построении алгоритма будут использоваться в качестве описаний переменных. Из требования полноты алгоритмических зависимостей следует, что для обеих задач для одинаковых множеств реквизитов будут определены одинаковые множества алгоритмических зависимостей: $\Phi^i \equiv \Phi^j$ и $F^i \equiv F^j$. Таким образом, совпадение входных и выходных данных в задачах приводит к совпадению всех исходных данных для построения описания алгоритма. Докажем, что

построение алгоритма будет однозначно при выполнении требований леммы. Из определения операторов на основе алгоритмических зависимостей и их функциональной связности следует, что описание алгоритма задачи будет являться множеством зависимостей вида: $D_i = f_k^T(\{D_j\})$, где $D_i, D_j \in D$ - структуры данных, описывающие переменные, f_k^T - оператор, реализующий алгоритмическую зависимость между переменными. Из идентичности множеств структур данных и полноты алгоритмических зависимостей следует, что при построении алгоритма для идентичных реквизитов будут получены идентичные описания операторов, реализующих зависимости между реквизитами. Также из полноты алгоритмических зависимостей следует, что при независимой разработке будут получены идентичные множества описаний зависимостей. Поскольку последовательность выполнения операторов управляется потоками данных, то нет необходимости задавать эту последовательность в явном виде. Следовательно, возможность внесения неоднозначности при описании параллельных действий отсутствует. Однозначность построения алгоритма следует из сочетания требований наличия единственной последовательности выполнения операторов и полноты алгоритмических зависимостей. Если предположить, что при соблюдении указанных требований процесс построения алгоритма неоднозначен, то следует допустить ситуацию, когда при построении алгоритма будут получены описания, различающиеся последовательностью выполнения операторов. Предположить различия в описании самих операторов нельзя, поскольку их идентичность была доказана ранее. Поскольку последовательность операторов задается только готовностью данных, то различие в описании последовательности операторов приводит к различию в операндах операторов. Различие в операндах противоречит требованию полноты функциональных зависимостей. Следовательно, сделанное предположение неверно. Доказательство леммы завершено.

Заключение

Определен класс мультиаспектных систем и выявлены его особенности. Это информационные системы организационного управления с возможностью настройки под задачи конкретного пользователя, имеющие сильную связь компонентов по данным и функциям. Поскольку стандартные методы декомпозиции на компоненты для мультиаспектных систем неэффективны, то для снижения затрат на разработку и модификацию предложен аспектный подход к проектированию систем. Основной идеей аспектного подхода является декомпозиция на задачи и однозначность процесса составления описания задачи при ее разработке.

Определено свойство однозначности построения алгоритма задачи и доказана лемма о том, что свойство однозначности построения алгоритма будет достигнуто, если:

- выполняется свойство однозначности построения структур данных;

- описание алгоритма составляется как последовательность выполнения операторов, управляемая готовностью данных;

- операторы выделяются на основе алгоритмических зависимостей и обладают функциональной связностью;

- между множеством входных данных и результатом существует единственная последовательность операторов;

- при построении алгоритма учитываются все алгоритмические зависимости между входными, выходными и промежуточными данными.

Литература

1. Виноградова М.В. Вопросы проектирования кафедральных информационных систем // Проблемы построения и эксплуатации систем обработки информации и управления / Под ред. В.М. Черненького (М.). –2002. - Вып. 4.– С. 34-40.
2. Данчул А.Н. Декомпозиционные методы в проектировании автоматизированных систем организационного управления: Дисс. ... докт. техн. наук. - М., 1997 - 260 с.
3. Дейт К. Введение в системы баз данных. - М.: Издательский дом Вильямс, 2002. – 1072 с.
4. Лавров С.С. Программирование. Математические основы, средства, теория. – СПб.: БХВ-Петербург, 2001. – 320 с.
5. Мамиконов А.Г., Кульба В.В., Косяченко С.А. Типизация разработки модульных систем обработки данных. – М.: Наука, 1989. – 165 с.
6. Технологии разработки программного обеспечения: Учебник. / Орлов С.А. – СПб.: Питер, 2002. – 464с.
7. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. – СПб.: Питер, 2002. – 496 с.

electronic scientific and technical periodical
SCIENCE and EDUCATION
EL № FS 77 - 30569. №0421100025. ISSN 1994-0408

Uniqueness property of algorithm development when designing multiaspect information systems

77-30569/242870

11, November 2011
author: M.Vinogradova

Bauman Moscow State Technical University

vinogradova.m@gmail.com

Class of multiaspect information systems was defined as class of organized control systems, closely-coupled with data and functions. The problem of decreasing the multiaspect system development cost was set. Standard methods of decomposition and out-of-the-box solutions were recognized ineffective. Aspect approach to designing of information system based on strict formalization of design process was proposed. Uniqueness property of algorithm development was formulated; it allowed to achieve the uniqueness of problem description when outlining it. Requirements that allowed to achieve the uniqueness were specified. Lemma that the uniqueness property of algorithm development was achieved with the fulfillment of listed requirements was proved.

Reference

1. Vinogradova M.V., in: V.M. Chernen'kii (Ed.), Problems of construction and operation of systems of information processing and control, Moscow, Is. 4, 2002, pp. 34-40.
2. Danchul A.N., Dr.Sci.Tech.dissertation, Moscow, 1997, 260 p.
3. Deit K., Introduction to database systems, Moscow, Izdatel'skii dom Vil'iams, 2002, 1072 p.
4. Lavrov S.S., Programming. Mathematical foundations, funds, theory, SPb.: BKhV-Peterburg, 2001, 320 p.
5. Mamikonov A.G., Kul'ba V.V., Kosiachenko S.A., Typification of the development of modular data processing systems, Moscow, Nauka, 1989, 165 p.
6. Orlov S.A., Technology of software development, SPb., Piter, 2002, 464 p.
7. Iakobson A., Buch G., Rambo Dzh., The unified software development process, SPb., Piter, 2002,