

Динамическое связывание аспектов в крупномасштабных системах технологического управления

С. К. Андриюшкевич, С. П. Ковалёв

*Конструкторско-технологический институт вычислительной техники СО РАН,
Новосибирск, Россия*

e-mail: andryushkevich@gmail.com, kovalyov@nsc.ru

Рассмотрены вопросы проектирования и формального анализа способов динамического связывания аспектов в составе систем технологического управления на базе единого журнала и реактивной модели событий. Представлена и обоснована формальная модель динамического связывания с применением конструкций теории множеств.

Ключевые слова: аспектно-ориентированное программирование и проектирование, реактивная модель событий, распределенные системы технологического управления.

Введение

Крупномасштабные автоматизированные системы технологического управления (АСТУ) обеспечивают поддержку деятельности широкого круга технических специалистов и руководителей предприятий, задействованных в выполнении производственных процессов с характерным циклом принятия решений от нескольких часов до нескольких недель. В структуре автоматизации предприятий системы класса АСТУ располагаются между системами локальной автоматизации производственных объектов (нижний уровень), такими как АСУ ТП, и автоматизированными системами управления предприятиями (верхний уровень). Они обеспечивают сбор, обработку, достоверизацию первичной информации, поступающей с нижнего уровня, и передачу этой информации на верхний уровень. Производственными процессами, обеспечением которых занимается АСТУ, являются, например, техническое обслуживание и ремонт, учет энергоресурсов, диспетчерское управление и т. д. В зарубежной литературе АСТУ часто рассматриваются под названием MES (Manufacturing Enterprise System) [1].

Производственный процесс представляет собой сценарий выполнения определенных задач. Поскольку одни и те же задачи могут входить в несколько производственных процессов, автоматизация процессов по отдельности приводит к дублированию функциональности, сложности комплексирования и выполнения сквозных процессов. Для решения этих проблем предлагается применять аспектно-ориентированный подход [2]. Последний позволяет модуляризовать производственные задачи, задействованные в различных процессах, в виде аспектов, явно выделять условия их связывания, а также идентифицировать элементы системной архитектуры, участвующие в их автоматизации. Таким образом, комплексная система автоматизации производственного процесса формируется путем объединения различных аспектов в одно целое.

Связывание аспектов АСТУ может выполняться на уровне структур данных (статическое связывание [3]) и на уровне сценариев выполнения процессов (динамическое связывание).

В настоящей работе рассматриваются вопросы динамического связывания аспектов АСТУ. Связывание выполняется на базе единого технологического журнала АСТУ с применением реактивной модели событий. Построена формальная модель динамического связывания аспектов на базе представления совокупностей событий частично упорядоченными множествами. Разделение на аспекты приводит к разметке событий идентификаторами групп — классов задач, для решения которых они служат. Отметим, что разнообразные помеченные структуры часто фигурируют в литературе в качестве формальных моделей распределенных систем, однако природа меток и способы их синтеза обычно не исследуются.

1. Подсистема мониторинга состояний АСТУ

Ход выполнения производственных процессов отражается в потоке событий, порождаемых компонентами системы. Источниками событий могут быть любые виды объектов системы: оборудование, пользователи, программные модули АСТУ и т. д.

Для реализации единой схемы автоматической обработки событий системы все события регистрируются в едином журнале с обязательной фиксацией таких параметров как тип события, объект события, дата и время возникновения события, регистратор события и т. д. Это позволяет упорядочить события по времени их возникновения в хронологическом порядке.

Многие участки конкретного производственного процесса обладают строгой детерминированностью, поэтому события, отражающие их ход, имеют причинно-следственную зависимость и образуют цепочки. Состав цепочек событий не однороден, так как включает события различных производственных задач. Для выделения событий, формируемых в рамках выполнения одной производственной задачи, вводятся группы событий, например, сбора данных, технического обслуживания, формирования отчетов. Таким образом, с одной стороны, цепочка может состоять из событий различных групп, с другой — события одной группы могут быть никаким образом не взаимосвязаны.

При проектировании АСТУ необходимо выделять подсистемы и модули, обеспечивающие выполнение специализированных групп функций. В большинстве производственных процессов можно определить такие классы задач как сбор, хранение, обработка, формирование отчета, предоставление интерфейса пользователя для доступа к данным и функциям и т. д. Традиционным подходом к выделению подсистем является автоматизация “от процесса”, когда одна подсистема или модуль автоматизируют весь процесс или его существенную часть. Достоинства этого подхода — низкие затраты на проектирование, высокая скорость реализации первых процессов, недостатки — дублирование функций, обеспечивающих автоматизацию различных классов производственных задач, а также существенные ограничения по расширению системы, так как связи между модулями подсистемы заданы жестко и реализуются посредством прямых программных вызовов.

Альтернативный вариант декомпозиции АСТУ — выделение функциональных подсистем согласно классам автоматизируемых производственных задач. В этом случае обеспечение выполнения целого производственного процесса возлагается на специальный компонент — компоновщик, который выполняет динамическое связывание функ-

циональных подсистем по заранее заданным правилам. К плюсам данного подхода относятся отсутствие дублирования функциональности, высокая адаптируемость системы под различные производственные процессы, удобство мониторинга и диагностики функционирования компонентов системы, к минусам — относительно высокие затраты на проектирование на начальном этапе и необходимость разработки значительного набора инфраструктурных компонентов на ранних стадиях проекта.

Типовая функциональная структура АСТУ, построенная с выделением функциональных подсистем по классам задач (рис. 1 [4]), состоит из семи подсистем:

— информационная модель объекта управления (ИМОУ) — сбор, унификация и хранение нормативно-справочной информации, поддержка жизненного цикла объектов управления;

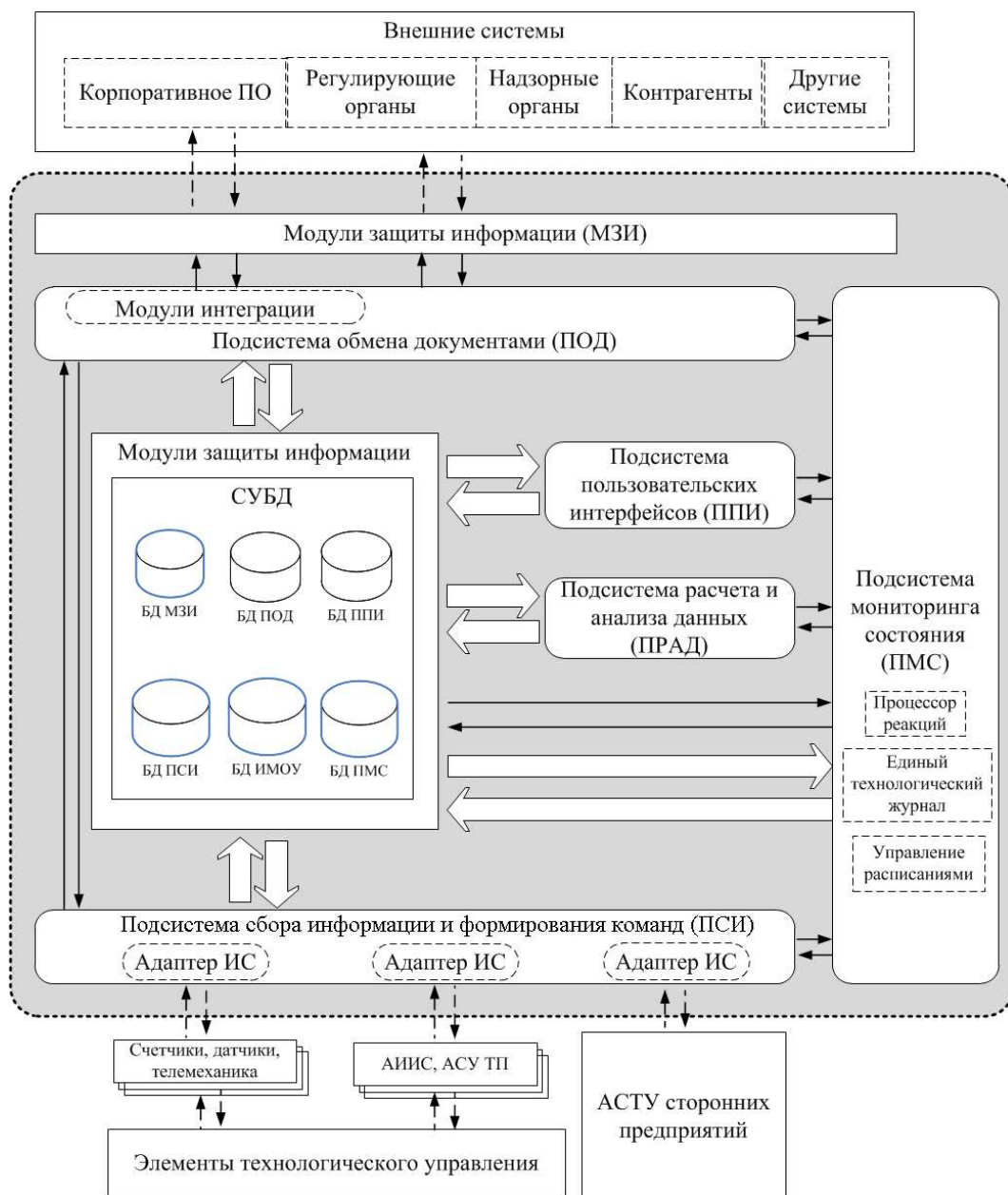


Рис. 1. Схема функциональной структуры АСТУ [4]

— подсистема сбора информации и формирования команд (ПСИ) — сбор и хранение данных со средств измерений и от внешних измерительных систем; функции телеуправления;

— подсистема мониторинга состояний (ПМС) — сбор и анализ событий, управление расписаниями, мониторинг производственных процессов, управление автоматической обработкой событий, диагностика компонентов системы;

— подсистема расчета и анализа данных (ПРАД) — оценка уровня достоверности результатов измерений, расчет учетных показателей, анализ и прогнозирование;

— подсистема обмена документами (ПОД) — организация эффективного обмена информацией с разнородными внешними системами, формирование регламентных и пользовательских отчетов;

— подсистема пользовательских интерфейсов (ППИ) — предоставление пользователям системы единой точки доступа к информации и к функциям управления;

— модуль защиты информации (МЗИ) — защита данных и функций от несанкционированного доступа.

Регистрацию и анализ событий АСТУ выполняет подсистема мониторинга состояний. Динамическое связывание производственных задач, в том числе построение цепочек событий, выполняется в рамках задачи мониторинга функционирования производственных процессов АСТУ, тесно связанной с задачей анализа функционирования распределенных систем [5], где успешно применяются принципы автономных вычислений (Autonomic Computing), основанные на реактивной модели работы системы. Для реализации механизма динамического связывания производственных задач в рамках производственных процессов реактивную модель событий предлагается адаптировать следующим образом.

Реактивная модель событий состоит из двух основных элементов:

— шаблоны — заранее заданный перечень событий или значений их параметров, активирующий реакцию;

— реакции — заранее заданный набор управляющих воздействий на компоненты системы, включая оповещение компонентов и пользователей о событии.

Анализом фактических событий системы и формированием управляющих воздействий занимается процессор реакций. Наличие информации о расписании производственного процесса (сценарии его нормального выполнения) позволяет контролировать и корректировать ход его выполнения. Контроль осуществляется путем сверки фактического потока событий с ожидаемым. Корректировка хода выполнения и обработка ошибок выполняются в случае нарушения фактического потока событий — формирования неверного события или превышения допустимого времени ожидания события.

В качестве иллюстрации взаимосвязи производственных процессов рассмотрим пример со счетчиком электрической энергии. Счетчик как физическое устройство участвует в ряде производственных процессов:

— монтаж/демонтаж оборудования;

— измерение электрической энергии;

— сбор данных;

— поверка и калибровка измерительных приборов;

— техническое обслуживание и ремонт оборудования.

Монтаж счетчика обуславливает постановку его в планы проведения метрологических мероприятий и технического обслуживания. Подключение счетчика к устройствам сбора данных в свою очередь обуславливает выполнение и контроль выполнения

процедур сбора. При отсутствии данных со счетчика необходимо проведение процедур технического обслуживания и ремонта, случай выдачи неверных значений должен сопровождаться внеплановой поверкой и калибровкой и т. д.

Рассмотренные производственные процессы не являются изолированными, они активируются динамически по факту возникновения соответствующих событий так же, как активируются функции других процессов в ходе их выполнения.

2. Аспектно-ориентированный подход к проектированию ПМС

В конце 1990-х годов был предложен аспектно-ориентированный подход, который позволяет выявлять пересекающиеся системные элементы и обеспечивать их консолидацию в ходе проектирования и функционирования системы. Наиболее известная область применения этого подхода — аспектно-ориентированное программирование (АОП) [2]. Его использование для проектирования систем развито слабо. В настоящей работе предлагается расширение аспектно-ориентированного подхода для решения задач проектирования АСТУ путем динамического связывания аспектов.

Выполнение одного производственного процесса не может быть локализовано в одном программном модуле, так как для реализации его производственных задач необходимо задействовать наборы функций, рассредоточенные (scattered) по нескольким функциональным подсистемам. Для того чтобы модуляризовать рассредоточенную функциональность, предлагается использовать подход, основанный на аспектно-ориентированном принципе. В данном подходе классы производственных задач являются аспектами функционирования системы. Аспектная декомпозиция дает возможность явно выделить те части системной архитектуры (интерфейсы взаимодействия, функции, алгоритмы, события, группы событий, элементы БД и т. д.), которые задействованы в решении задачи. Для обеспечения прозрачности и двусторонней трассируемости производственных задач к системным элементам на уровне системной архитектуры выделены специальные структуры: системные функции (статическое связывание) [3] и группы событий (динамическое связывание).

Основные концепции АОП [6] реализуются в рамках реактивной модели подсистемы мониторинга состояний АСТУ следующим образом:

- точки соединения или точки вызова (joint point). Единый технологический журнал, в котором регистрируются все события системы, формирует базу для реактивной модели. Любое событие является точкой соединения;

- конструкция отбора точек соединения — срез (point cut). Набор точек соединения (типов событий и их параметров), определяющий условия активации (переплетения) выполнения различных функций в рамках реализации производственной задачи. В реактивной модели термину “срез” соответствует “шаблон”;

- конструкция изменения поведения системы — совет (advice). Управляющие воздействия на подсистемы АСТУ. В реактивной модели термину “совет” соответствует “реакция”;

- модульный элемент, объединяющий все перечисленные выше конструкции пересечения, — аспект (aspect). В терминах реактивной модели модульным элементом является совокупность “шаблонов” и “реакций”, обеспечивающих обработку некоторой группы событий;

- средство связывания, компоновщик (weaver). Функции компоновщика выполняет процессор реакций.

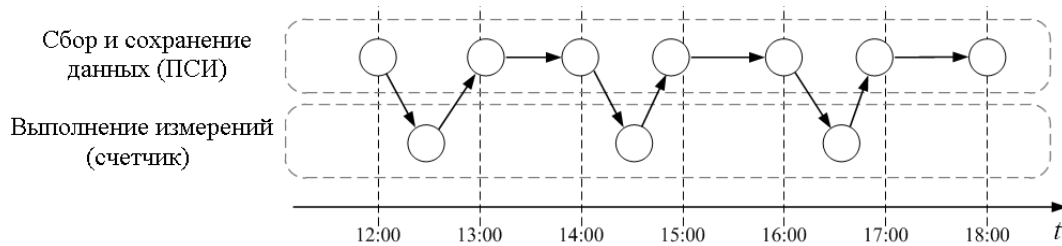


Рис. 2. Схема пересечения аспектов измерения и сбора данных в производственном процессе “Сбор данных с приборов учета” (в скобках — наименование элементов функциональной структуры АСТУ, сформировавших рассматриваемые события; то же на рис. 3)

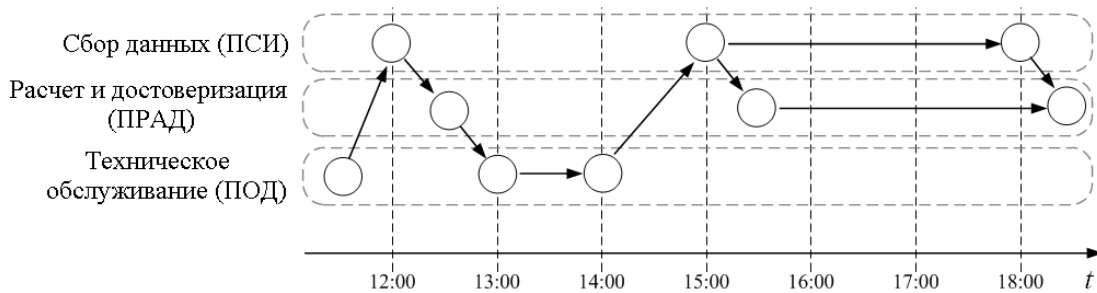


Рис. 3. Многоуровневое связывание аспектов в рамках процесса учета энергоресурсов в процессе “Учет энергоресурсов”

На рис. 2 приведена схема примера — производственного процесса “Сбор данных с приборов учета”. В рамках данного процесса требуется выполнить задачи измерения, опроса приборов учета и сохранения результатов в БД. На схеме представлены события аспектов измерения и сбора данных. События названных аспектов связаны причинно-следственной связью, поэтому их активация выполняется в “перемежающемся” режиме.

На рис. 3 показана часть схемы процесса “Учет энергоресурсов”, демонстрирующей многоуровневое связывание аспектов АСТУ. В рамках выполнения данного процесса были сформированы следующие события. Событие о завершении монтажа прибора учета активировало процесс сбора данных с прибора. Процессы сбора и анализа данных выполняются периодически: после завершения сбора происходит анализ. На схеме показано, что события анализа данных активировали процессы технологического обслуживания, например, ввиду выявления факта поступления недостоверных данных. Двум событиям технического обслуживания соответствуют события необходимости замены прибора и регистрации этого факта. После замены прибора учета процессы сбора и анализа данных выполняются без активации других процессов.

3. Формальная модель динамического связывания

Опишем динамическое связывание аспектов АСТУ на формальном языке. Формализация позволяет дать строгую однозначную трактовку введенных понятий, не зависящую от способа их программной реализации, выявить условия корректности связывания, представить типовые приемы синтеза систем в лаконичной форме абстрактных шаблонов (patterns). Однозначность трактовки особенно важна для аспектно-ориентирован-

ных методов проектирования ввиду того, что к настоящему времени еще не сформировалось единого понимания сути АОП [7].

Формальному моделированию программных систем в терминах потоков событий посвящена обширная литература. Базовым математическим представлением совокупности событий, составляющей некоторый сценарий функционирования системы, является частично упорядоченное множество (poset). Его элементы отвечают событиям, а частичный порядок строится путем рефлексивно-транзитивного замыкания отношения причинно-следственной зависимости (антисимметричность гарантируется физическим законом причинности) (см., например, [8]). Поддержка аспектов заключается в разметке сценария группами событий, преобразующей его в частично упорядоченное мультимножество (pomset) [9]. Мы рассматриваем разметку как сюръективное отображение множества событий на множество имен аспектов, к которым они относятся, поскольку сценарий не может содержать “пустых” аспектов. Непустой сценарий, все элементы которого помечены одинаково, естественным образом называется аспектом.

Определение 1. *Аспектно-ориентированным формальным сценарием называется тройка вида $\langle E, \leq, l \rangle$, где E — множество, элементы которого называются событиями; \leq — частичный порядок на E , порожденный причинно-следственной зависимостью событий; l — сюръективное отображение E во множество, называемое аспектной структурой (разметкой) сценария.*

Действиями по интеграции аспектно-ориентированных сценариев компонентов в сценарии системы являются в точности все гомоморфизмы, поскольку ни события, ни причинно-следственные связи, ни аспекты не могут быть “забыты”. Напомним, что гомоморфизмом сценария $\langle E, \leq, l \rangle$ в $\langle E', \preceq, l' \rangle$ является любое отображение $f : E \rightarrow E'$ такое, что $x \leq y$ влечет $f(x) \preceq f(y)$ для любых $x, y \in E$, и существует такое (обязательно единственное) отображение переименования меток $a_f : l(E) \rightarrow l'(E')$, что $a_f \circ l = l' \circ f$ (см. [9]).

Частным случаем интеграции сценариев является ведение единого журнала событий. Журнал представляет собой аспект над множеством вещественных чисел \mathbb{R} с естественным линейным порядком, описывающим стрелу физического времени. Регистрация событий, образующих сценарий X , в журнале формально описывается гомоморфизмом $t : X \rightarrow \mathbb{R}$.

Динамическое связывание аспектно-ориентированных сценариев с формальной точки зрения состоит из двух шагов. Сначала (виртуально) создается достаточное количество копий сценария совета (реакции), по одной на каждую точку соединения, с маркировкой соответствующих им точек вызова в базе (журнале). Далее эти точки склеиваются друг с другом так, чтобы не разрушить аспектную структуру базы и совета. Обозначим формальный сценарий базы через B , совета — через W , процессора реакций, выполняющего связывание, — через C . Процессор реакций интегрируется с базой в точках соединения, а с советом — в точках вызова, поэтому соответствие обрабатываемых событий (точек соединения) точкам вызова реакций имеет вид пары гомоморфизмов $j : B \leftarrow C \rightarrow W : e$ (такое сокращенное обозначение используется для пары $j : C \rightarrow B, e : C \rightarrow W$). Первый шаг связывания может быть формализован как построение произведения $C \times W$, в котором точки вызова маркируются посредством отображения $\check{e} : C \rightarrow C \times W$, сопоставляющего каждому $x \in C$ пару $(x, e(x))$. Второй шаг состоит в построении амальгамы отображений j и \check{e} (стандартная конструкция склеивания элементов множества [10]): прямая сумма $B \amalg (C \times W)$ факторизуется

по отношению эквивалентности \sim , порожденному отношением $\{(j(x), \check{e}(x)) \mid x \in C\}$. Чтобы при этом частичный порядок не стал беспорядком, проводится дополнительная факторизация по транзитивному замыканию следующего отношения:

$$x \lesseqgtr y \Leftrightarrow \exists x' \exists y' \exists x'' \exists y'' (x \sim x' \wedge x \sim x'' \wedge y \sim y' \wedge y \sim y'' \wedge x' \leq y' \wedge y'' \leq x'').$$

Единицы аспектной структуры, помечающие элементы одного класса эквивалентности, отождествляются. Связывание будет корректным, если аспектная структура его результата также является амальгамой, так как в данном случае аспектные структуры базы и совета не разрушаются. Докажем, что это условие выполняется не всегда: процессор реакций должен поддерживать конкурентность (concurrency) в том смысле, что он не должен детерминировать порядок вызова различных аспектов совета, подшиваемых к одной точке соединения.

Определение 2. Связыванием пары гомоморфизмов сценариев $j : B \leftarrow C \rightarrow W : e$ называется амальгама пары гомоморфизмов $j : B \leftarrow C \rightarrow C \times W : \check{e}$ (схемы связывания), обозначаемая через $j \odot e$, если ее аспектная структура представляет собой амальгаму отображений a_j и a_e .

Теорема. Связывание пары $j : B \leftarrow C \rightarrow W : e$ существует тогда и только тогда, когда для любых $x, y \in C$ условия $j(x) = j(y)$ и $x \leq y$ влекут $l(v) = l(e(x))$ для любого $v \in W$ такого, что $e(x) \leq v \leq e(y)$.

Доказательство. Для того чтобы аспектная структура результата связывания была амальгамой, необходимо и достаточно, чтобы факторизация по отношению \lesseqgtr^∞ не привела к отождествлению различных меток. Это равносильно тому, что для любых x, y из условия теоремы отображение a_e устанавливает биекцию между множествами меток множеств $\{z \mid x \leq z \leq y\}$ и $\{(z, v) \mid x \leq z \leq y \wedge e(x) \leq v \leq e(y)\}$, т. е. тому, что все элементы множества $\{v \mid e(x) \leq v \leq e(y)\} \subseteq W$ помечены одинаково. \square

Определение связывания отражает ряд его ожидаемых свойств, например наличие неразрушающего включения базы в результат. Действительно, при амальгамировании различные точки базы попадают в разные классы эквивалентности, поэтому существует каноническое изоморфное вложение $b : B \rightarrow (j \odot e)$. С его помощью можно определить многошаговое связывание взаимно независимых советов с общей базой: если имеются связываемые пары $j : B \leftarrow C \rightarrow W : e$ и $j' : B \leftarrow C' \rightarrow W' : e'$, то их можно естественным образом собрать в одно целое путем связывания пары $b \circ j' : (j \odot e) \leftarrow C' \rightarrow W' : e'$, которое существует в силу теоремы. Непосредственно проверяется, что результат многошагового связывания не зависит от порядка подшивки отдельных советов: он совпадает с результатами связывания пары $b' \circ j : (j' \odot e') \leftarrow C \rightarrow W : e$, где $b' : B \rightarrow (j' \odot e')$ — каноническое изоморфное вложение.

Приведем примеры проявления эффекта спутывания при связывании пары $j : B \leftarrow C \rightarrow W : e$. Если реакция вызывается на каждое событие, т. е. отображение j является изоморфизмом, то, как и следует ожидать, связывание существует и его результат представляет собой произведение $B \times W$. Другим примером является случай, когда совет W является аспектом. Здесь связывание также существует, и поскольку отображение a_e является изоморфизмом, аспектная структура результата связывания изоморфна аспектной структуре базы. Таким образом, база связывания в некотором смысле поглощает аспект, затрудняя его идентификацию в составе системы. Эта проблемная ситуация явилась главным стимулом к созданию АОП [2].

С формальной точки зрения идентификация аспекта в составе сценария представляет собой вычисление прообраза одноточечного множества относительно отображения разметки. Возможность рассматривать аспект как единицу традиционного модульного проектирования (“черный ящик”) возникает в случае, когда все составляющие его события имеют одинаковую структуру причинно-следственных связей с окружением [11].

Определение 3. Аспект $l^{-1}(t) \subseteq E$ сценария $\langle E, \leq, l \rangle$, где t — некоторая метка, называется модуляризируемым, если для любых x, x', y таких, что $l(x) = l(x') = t$ и $l(y) \neq t$, выполняются условия $x \leq y \Leftrightarrow x' \leq y$ и $y \leq x \Leftrightarrow y \leq x'$.

Например, аспект очевидным образом модуляризируем, если он представляет собой компонент связности частичного порядка сценария — такой аспект никак не связан с остальными. Также модуляризируемыми являются одноточечные аспекты. В качестве примера трудной задачи синтеза сценариев приведем реализацию архитектуры клиент — сервер: аспекты, перемежающиеся (interleaving) в составе линейно упорядоченного сценария, не модуляризируемы, и в то же время сценарий не может быть синтезирован из них посредством связывания. Ввиду этого взаимодействие клиента с сервером обычно реализуют непосредственно, без посредничества процессора реакций на события.

Понятия, обзор которых представлен в данном разделе, можно обобщить на широкий класс формальных методов синтеза программных систем. Действительно, гомоморфизмы, произведения, амальгамы, прообразы являются частными случаями универсальных абстрактных конструкций, изучаемых в рамках алгебраической дисциплины — теории категорий [12]. В работе [13] описана общая формальная схема (метамодель) обогащения произвольной методологии проектирования систем, описанной на языке теории категорий, аспектно-ориентированными концепциями и приемами связывания и модуляризации аспектов. При этом моделирование поведения систем помеченными сценариями выступает в роли операционной семантики аспектно-ориентированного подхода, поскольку оно непосредственно выражает идею связывания. Частный случай сценарного моделирования, охватывающий упрощенный вариант АОП, известен как трассовая семантика аспектов [14].

Заключение

Применение аспектно-ориентированного подхода к задачам проектирования АСТУ позволяет обеспечить гибкий механизм компоновки сложных процессов путем задания правил связывания различных производственных задач, которые в данном случае оформляются в виде модульных элементов, называемых аспектами. Для аспекта задаются формальные правила активации на основе шаблонов событий. За счет этого обеспечивается локальность правил связывания для каждой производственной задачи без привязки ее к конкретным процессам. Фактические контуры производственных процессов явно не выделяются, так как формируются динамически на основании правил активации аспектов.

Представленные в работе результаты были апробированы при создании интеграционной платформы “Энергиус” [4], на базе которой построены предметно-ориентированные АСТУ в области энергетики, успешно функционирующие на ряде предприятий федерального масштаба.

Список литературы

- [1] KLETTI J. Manufacturing Execution Systems MES. Berlin: Springer, 2007. 275 p.
- [2] KICZALES G., LAMPING J., MENDHEKAR A. ET AL. Aspect-oriented programming // Proc. 11th European Conf. on Object-Oriented Programming (ECOOP 1997). Jyvaskyla, Finland / Lecture Notes in Computer Sci. Vol. 1241, No. 1. Berlin: Springer-Verlag, 1997. P. 220–242.
- [3] АНДРИЮШКЕВИЧ С.К. Построение информационной модели крупномасштабных объектов технологического управления с применением аспектно-ориентированного подхода // Вестник НГУ. Информационные технологии. 2010. Т. 8, № 3. С. 34–45.
- [4] КОВАЛЁВ С.П., АНДРИЮШКЕВИЧ С.К., ГУСЬКОВ А.Е. Свидетельство о государственной регистрации программы для ЭВМ № 2009613359. Интеграционная платформа учета и управления энергообеспечением “Энергиус”. 2009.
- [5] КУЗНЕЦОВ А.А., КОВАЛЁВ С.П. Тестирование и мониторинг в распределенных автоматизированных системах технологического управления // Вычисл. технологии. 2009. Т. 14, № 4. С. 57–69.
- [6] LADDAD R. AspectJ in Action. Enterprise AOP with Spring Applications. N.Y.: Manning, 2010. 519 p.
- [7] STEIMANN F. The paradoxical success of aspect-oriented programming // Proc. OOPSLA’06. Portland, 2006. P. 481–497.
- [8] RAYNAL M., SINGHAL M. Logical time: capturing causality in distributed systems // IEEE Computer. 1996. Vol. 29, No. 2. P. 49–56.
- [9] PRATT V.R. Modeling concurrency with partial orders // Intern. J. Parallel Program. 1986. Vol. 15, No. 1. P. 33–71.
- [10] ГОЛДБЛАТТ Р. Топосы. Категорный анализ логики. М.: Мир, 1983.
- [11] GLABBEK R.J. VAN, GOLTZ U. Refinement of actions and equivalence notions for concurrent systems // Acta Informatica. 2000. Vol. 37, No. 4–5. P. 229–327.
- [12] МАКЛЕЙН С. Категории для работающего математика. М.: Физматлит, 2004.
- [13] KOVALYOV S.P. Modeling aspects by category theory // Proc. of the 9th Workshop on Foundations of Aspect-Oriented Languages. Rennes, France, 2010. P. 63–68.
- [14] DOUENCE R., FRADET P., SUDHOLT M. Trace-based aspects // Aspect-Oriented Software Development. Boston: Addison Wesley, 2004. 800 p.

*Поступила в редакцию 2 июня 2011 г.,
с доработки — 3 августа 2011 г.*