

УДК 004.041

## УПРАВЛЕНИЕ НА ОСНОВЕ МЕТОДА ПРЕЦЕДЕНТОВ

**Кужелев П.Д.**, к.т.н., доцент, МИИГАИК, E-mail: trianon6@yandex.ru,  
Москва, Россия

**Аннотация.** Статья описывает применение метода прецедентов для решения задач управления. Особенностью метода является необходимость комплектования библиотеки прецедентов, которая должна храниться в базе знаний. Процесс применения прецедентов для управления моделируется в виде цикла. В результате данного метода получают три результата: решения для заказчика задачи, решение новой задачи; новый прецедент с условиями и решением.

**Ключевые слова:** информация, управление, искусственный интеллект, правдоподобный вывод, прецедент, решение задачи, база знаний.

## MANAGEMENT BASED METHOD PRECEDENT

**Kuzhelev P.D.**, PhD., Assoc.Prof., E-mail: trianon6@yandex.ru,  
Moscow, Russia

**Abstract.** This article describes the application of the method of precedents for control tasks. feature of the method is the need for acquisition of library of precedents, which should be stored in the knowledge base. The process of applying precedents for controlling a simulated cycle. As a result, this method produced three results: solutions for the customer's problem, the solution of a new problem; new precedent with the conditions and goals.

**Keywords:** information. management, artificial intelligence, believable conclusion precedent, the decision problem, the knowledge base

**Введение.** Современные исследования по развитию систем и технологий управления связаны с проблемой моделирования правдоподобных рассуждений (так называемых рассуждений «здорового смысла») [1,2]. Такие методы моделирования рассуждений включают: индуктивный, дедуктивный и абдуктивный - вывод; нечеткие логики, методы аналогий и методы прецедентов. Наряду с методами правдоподобного вывода в последнее время стали активно применяться методы принятия решений на основе прецедентов [3-9].

Данные методы могут быть эффективны в системах поддержки принятия решений (СППР), при обработке и анализе больших данных[10], а также для поиска решения в проблемных ситуациях, а также для поддержки интеллектуальных транспортных систем [11].

**Основная часть.** В данной концепции будем рассматривать управление, включая в рассмотрение: объект управления (управляемую систему в дальнейшем система),

систему управления (управляющую систему в дальнейшем СУ), среду, в которой находится объект управления. В теории прецедентов важным понятием является «Актер».

*Актер* представляет собой связанное множество ролей, которые пользователи прецедентов исполняют во время взаимодействия с ними. Актерами могут быть люди, информационные или интеллектуальные системы. Актеры сущности, находящиеся вне системы. Обычно актер, представляет роль, которую в данной системе исполняет человек или система. Экземпляр актера представляет собой результат процесса моделирования от общего к частному, который называется экземпляция. Актеры являются одной из важнейших составляющих моделей управления, они не являются частью системы, так как существуют вне ее.

*Процесс* (process) описывает последовательность событий и действий, необходимых для достижения поставленной цели.

Прецедент (use case) специфицирует поведение системы или ее части и представляет собой описание множества последовательностей действий (включая варианты), выполняемых системой для того, чтобы актер мог получить определенный результат.

*Прецедентом* называется описание множества последовательностей действий, включая варианты, выполняемых системой для того, чтобы актер мог получить определенный результат. Графически прецедент изображается в виде эллипса. Нотация прецедента похожа на нотацию кооперации.

С помощью прецедентов можно описать поведение разрабатываемой системы, не определяя ее реализацию. Таким образом, они позволяют достичь взаимопонимания между управленцами, разработчиками, системными аналитиками, экспертами и конечными пользователями продукта.

Любой прецедент имеет имя, отличающее его от других прецедентов. Оно должно быть уникально и чаще представляет собой текстовую строку. Взятое само по себе оно называется простым именем. Составное имя включает спереди имя пакета, в котором находится прецедент. Обычно, при изображении прецедента указывают только его имя.

Актеров можно связывать с прецедентами только отношениями ассоциации. Ассоциация между актером и прецедентом показывает, что они общаются друг с другом, возможно, посылая или принимая сообщения. Можно определить общие типы актеров, а потом специализировать их, создавая разновидности, с помощью отношения специализации.

Прецедент описывает, что делает система (подсистема, класс или интерфейс), но не определяет, каким образом она это делает. В процессе моделирования всегда важно разделять внешнее и внутреннее представления.

Можно специфицировать поведение прецедента путем описания потока событий в текстовой форме - в виде, понятном для постороннего читателя. В описание необходимо включить указание на то, как и когда прецедент начинается и заканчивается, когда он взаимодействует с актерами и какими объектами они обмениваются. Важно обозначить также основной и альтернативный потоки поведения системы.

Дадим полный список определений различных видов прецедентов, которые создаются на различных этапах планирования и анализа. Прецеденты высокого уровня (high-level use case) - это краткие описания процессов, обычно состоящие из 2-3 предложений. Такой тип описания удобно использовать на начальном этапе формулирования требований к системе для быстрого осознания степени сложности и функций системы. Прецеденты высокого уровня - это лишь краткое описание, имеющее слабое отношение к конкретным управленческим решениям.

Развернутые прецеденты (expanded use case) представляют собой более подробное описание, чем прецеденты высокого уровня. Они оказываются полезными для углубленного понимания происходящих процессов и требований. Очень часто развернутые прецеденты имеют форму диалога между исполнителем и системой.

Основной особенностью этого прецедента является наличие раздела "Типичный ход событий", в котором описывается последовательность событий. На этапе формулирования требований в развернутом формате целесообразно представлять лишь наиболее важные и значительные прецеденты, а более подробное описание остальных прецедентов отложить до того цикла разработки, в котором они должны быть реализованы. Жесткого формата на представление развернутых прецедентов нет. Один из вариантов их описания приведен ниже. В верхней части развернутой формы содержится обобщенная информация, большая часть которой берется из соответствующего прецедента высокого уровня.

Следующий раздел описания "Типичный ход событий", является основной частью развернутого прецедента. В нем отображается содержание подробного диалога между исполнителями и системой. В этом разделе описывается наиболее стандартная, или типичная, последовательность событий - нечто среднее между видами деятельности и успешным завершением процесса.

Таблица 1.

**Развернутый прецедент**

Прецедент	Имя прецедента
Агенты	Перечень внешних агентов, а также те из них, кто инициирует данный прецедент
Цель	Цель прецедента
Краткое описание	Копия содержимого прецедента высокого уровня или некоторая аналогичная обобщенная информация
Тип	1. Главный, второстепенный или дополнительный 2. Идеальный или реальный
Ссылки	Связанные прецеденты и (или) функции системы.

Альтернативные ситуации в типичную последовательность не включаются. «Типичный ход событий» включает пронумерованные действия исполнителей и пронумерованные описания откликов системы, Типичный ход событий отделяется от альтернатив. Завершающий раздел, "Альтернативный ход событий", должен содержать важные альтернативы, которые могут возникать в ходе типичной последовательности. Если они слишком сложны, их разворачивают в собственные прецеденты.

По мере уточнения требований к системе потоки событий прецедента удобнее записать в графической нотации, используя диаграммы взаимодействий. Обычно для главного потока (типичного хода событий) используют диаграмму последовательностей, а для дополнительных (альтернатив) ее варианты.

Прецедент описывает не одну, а множество последовательностей, поэтому выразить все детали прецедента с помощью одной последовательности просто невозможно. Один прецедент, обычно, описывает несколько последовательностей, или сценариев, каждый из которых представляет одну из возможных вариаций данного потока событий.

Таким образом, сценарий - это некоторая последовательность действий, иллюстрирующая поведение системы. Сценарии находятся в таком же отношении к прецедентами, как экземпляры (объекты) к классам, то есть сценарий - это экземпляр прецедента.

Сложная система содержит несколько прецедентов, каждый из которых может разворачиваться в несколько сценариев. Для любого прецедента можно выделить основные сценарии, описывающие важнейшие последовательности, и вспомогательные, описывающие альтернативы.

Прецедент описывает желательное поведение системы (подсистемы, класса или интерфейса), но не определяет их реализацию. Это важнейшая особенность, так как

анализ системы, по результатам которого специфицируется ее поведение, по возможности не должен учитывать проблемы реализации, иными словами, как это поведение должно быть материализовано.

Для реализации прецедентов необходимо создать совокупность классов и других элементов, в результате совместного поведения которых будет достигнуто желаемое поведение. Такая совокупность называется кооперацией. Реализацию прецедента с помощью кооперации можно специфицировать явно, используя отношение реализации. Но чаще всего один прецедент реализуется в точности одной кооперацией, так что явно это отношение описывать ненужно, в некоторых инструментальных средствах даже нет соответствующей пиктограммы, а наоборот некоторые из них, так или иначе, поддерживают это отношение.

Нахождение минимального набора хорошо структурированных коопераций, реализующих определенный во всех прецедентах системы поток событий, основная задача системной архитектуры.

Организации прецедентов. Для организации прецедентов их группируют в пакеты и помещают в библиотеку, которая является частью базы знаний. Прецеденты можно организовать, определив между ними отношения обобщения, включения и расширения. Отношения применяют, для выделения общего поведения. Общее поведение оценивают на основе анализа и обобщения других прецедентов. Отношение обобщения между прецедентами подобно отношению между классами. В соответствии с ним прецедент-потомок наследует поведение и семантику своего родителя, может замещать его или дополнять его поведение.

Прецедент-потомок может быть получен на основе экзemplяции из его родителя. Родитель и потомок могут иметь также конкретные экземпляры.

*Отношение обобщения* между прецедентами изображаются точно так же, как и обобщения между классами - в виде линии с незакрашенной стрелкой.

*Отношение включения* между прецедентами означает, что в некоторой точке базового прецедента инкорпорировано поведение другого прецедента. Включаемый прецедент никогда не существует автономно, а инстанцируется только как часть объемлющего прецедента. Можно считать, что базовый прецедент заимствует поведение включаемых.

Благодаря наличию отношения включения удастся избежать многократного описания одного и того же потока событий, поскольку общее поведение можно описать в виде самостоятельного поведения, включаемого в базовые.

Отношение включения является примером делегирования, при котором ряд

обязанностей системы описывается в одном месте - во включаемом прецеденте, а остальные прецеденты, когда необходимо включают эти обязанности в свой набор.

Отношения включения изображаются в виде зависимости со стереотипом *include*. Чтобы специфицировать место в потоке событий, где базовый прецедент включает поведение другого, пишется слово *include*, за которым следует имя включаемого прецедента.

*Отношение расширения* применяют для моделирования таких частей прецедента, которые пользователь воспринимает как необязательное поведение системы. Тем самым можно разделить обязательное и необязательное поведение.

Отношение расширения используется также для моделирования отдельных субпотоков, выполняемых только при определенных обстоятельствах. Их применяют для моделирования нескольких потоков, которые могут включиться в некоторой точке сценария в результате явного взаимодействия с актером.

Отношение расширения изображают в виде зависимости со стереотипом *extend*. Точки расширения базового сценария перечисляют в дополнительном разделе. Они являются просто метками, которые могут появиться в потоке базового прецедента. Прецедент может содержать несколько точек расширения, причем каждую по несколько раз, идентифицируемых по именам. Если определено несколько точек расширения, то расширяющие прецеденты будут последовательно выполняться в собственных потоках.

Организация прецедентов путем выделения общего поведения (отношение включения) и различных вариаций (отношение расширения) является важной составной частью процесса разработки простого сбалансированного и понятного набора прецедентов системы.

Прецеденты являются классификаторами и могут иметь атрибуты и операции, которые изображаются так же, как для классов. Атрибуты можно считать объектами внутри прецедента, которые требуются для описания его внешнего поведения, а операции - действиями системы, необходимыми для описания потока событий. Эти объекты и операции разрешается включать в диаграммы взаимодействия, чтобы специфицировать поведение прецедента. Как и ко всем остальным классификаторам, к прецедентам можно присоединять автоматы.

*Поведение элемента* Часто помощью прецедентов моделируют поведение элемента: системы в целом или подсистемы. При этом концентрируют внимание на том, что должен делать элемент, а не то, как он это будет делать.

Применение прецедентов к элементам представляет важность по трем причинам.

Во-первых, моделируя поведение элемента с помощью прецедентов, эксперты в предметной области (системные аналитики) могут описать взгляд на систему извне с такой степенью детализации, что разработчики сумеют сконструировать ее внутреннее представление. Прецеденты дают возможность экспертам, системным аналитикам, конечным пользователям и разработчикам общаться на одном языке.

Во-вторых, прецеденты позволяют разработчикам понять назначение элемента. Система, подсистема или класс могут быть сложными образованиями с большим числом операций и других составных частей. Описав прецеденты элемента, можно разобраться в том, как с ними обращаться.

В-третьих, прецеденты являются основой для тестирования каждого элемента на всем протяжении его управления. Постоянно сравнивая функционирование каждого элемента с прецедентами, можно контролировать его управление.

Моделирование поведения элемента осуществляется следующим образом:

А. Идентифицируют актеров, взаимодействующих с данным элементом. К числу кандидатов в актеры относятся агенты, которые требуют определенного поведения для выполнения своих задач или необходимы, прямо или косвенно, для выполнения функций элемента.

Б. Организуют актеров, выделив общие и специализированные роли.

В. Для каждого актера рассматривают основные пути его взаимодействия с элементом. Рассматривают также взаимодействия, изменяющие состояния элемента или его окружения или предполагающие реакцию на некоторое событие.

Г. Рассматривают альтернативные (исключительные) способы взаимодействия актеров с элементом.

Д. Организуют выявленное поведение в виде прецедентов, применяя отношения включения и расширения для выделения общего и исключительного поведения.

Е. По мере развития модели выявляют тенденцию к объединению прецедентов в концептуально и семантически близкие группы (пакеты).

Хорошо структурированный прецедент обладает следующими свойствами: именуется идентифицируемое поведение системы или ее части, выделяет общее поведение, извлекая его из всех прецедентов, выделяет вариации, помещая некоторое поведение в другие прецеденты, которые его расширяют, описывает поток событий в степени, достаточной для понимания посторонним читателем, описывается с помощью минимального набора сценариев, специфицирующих его нормальную и дополнительную семантику.

Моделирование контекста системы основано на обведении системы

воображаемой линией и выявлении актеров, которые находятся за этой линией и взаимодействуют с системой.

Любая система содержит внутри себя одни сущности, другие сущности остаются за ее пределами. Сущности внутри системы отвечают за реализацию поведения, которого ожидают сущности находящиеся снаружи. Внешние сущности составляют контекст системы. Таким образом, контекстом называется окружение системы.

Моделирование контекста системы состоит из следующих шагов:

А. Идентифицируют окружающих систему актеров. Для этого находят группы, которым участие системы требуется для выполнения их задач; группы, которые необходимы для осуществления системой своих функций; группы, взаимодействующие с ее внешними программными и аппаратными средствами, а также группы, выполняющие вспомогательные функции администрирования и поддержки.

Б. Организуют похожих актеров с помощью отношений обобщения/специализации.

В. Вводят стереотипы для каждого актера.

Г. Помещают актеров на диаграмму прецедентов и определяют способы их связи с прецедентами системы.

Тот же метод позволят моделировать и контекст подсистемы.

Моделирование требований к системе предполагает указание на то, что система должна делать с точки зрения внешнего наблюдателя, независимо от того, как она должна это делать.

Диаграммы прецедентов нужны для специфицирования желаемого поведения системы. Они позволяют рассматривать всю систему как "черный ящик": вы видите все, что находится вне ее, наблюдаете за ее реакцией на события, но ничего не знаете о внутреннем устройстве. Требование (requirement) - это особенность проекта, свойство или поведение системы. Сбору требований позволяет описывать условия взаимодействия между системой и внешними сущностями. Совокупность требований декларирует, что система должна делать. Хорошо спроектированная система должна полностью выполнять все требования предсказуемо и надежно. Ее создание начинается с определения цели (назначения), которая определяет требования. В ходе разработки совокупность требований может изменяться.

Требования можно выразить по-разному. Это формализованное описание или неструктурированный текст. Большая часть функциональных требований к системе может быть выражена в виде прецедентов использования.



Моделирование требований осуществляется следующим образом:

А. Устанавливают контекст системы, идентифицировав окружающих ее актеров.

Б. Для каждого актера рассматривают поведение, которого он ожидает или требует от системы.

Г. Выделяют общие варианты поведения как прецеденты.

Д. Выделяют общее поведение в новые прецеденты, которые будут использоваться другими;

Е. Выделяют вариации поведения в новые прецеденты, расширяющие основные потоки событий.

Ж. Моделируют эти прецеденты, актеров и отношения между ними на диаграмме прецедентов

И. Дополняют прецеденты примечаниями, описывающими нефункциональные требования, некоторые из таких примечаний можно присоединить к системе в целом.

Как правило, процесс вывода на основе прецедентов включает четыре основных этапа, образующих так называемый цикл рассуждения на основе прецедентов или CBR-цикл [12]. Предшествует этому циклу создание базы знаний, которая хранит библиотеку прецедентов. CBR-цикл направлен на решение новой задачи, поэтому возникновение задачи является основанием для применения данного подхода. Основными этапами CBR-цикла являются (рис.1):

- извлечение наиболее адекватного прецедента для сложившейся ситуации или по условиям поставленной задачи из библиотеки прецедентов (БП);
- повторное использование извлеченного прецедента для решения текущей задачи;
- адаптация в случае необходимости полученного решения в соответствии с требованиями задачи;
- получение решения и выдача его заказчику;
- сохранение нового решения как части нового прецедента;
- сохранение нового прецедента в базе знаний в библиотеке прецедентов.



Рис.1. Циклическое использование прецедента

**Заключение.** Использование метода прецедентов дает в руки управленца аппарат анализа сложных ситуаций и возможность их сопоставления, если даже они не могут быть формализованы или описаны формальными методами. Перспектива этого метода состоит в использовании его в сочетании с когнитивными моделями [13, 14] управления.

### Список литературы

1. Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных актов. - М.: Радио и связь, 1989. - 184 с.
2. Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М. В. Достоверный и правдоподобный вывод в интеллектуальных системах. 2-е издание // Под редакцией В.Н. Вагина, Д.А. Поспелова. - М.: ФИЗМАТЛИТ, 2008. - 712 с.
3. Варшавский П. Р., Еремеев А. П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений //Искусственный интеллект и принятие решений. – 2009. – №. 2. – С. 45-57.
4. Карпов Л. Е., Юдин В. Н. Адаптивное управление по прецедентам, основанное на классификации состояний управляемых объектов //Труды Института системного программирования РАН. – 2007. – Т. 13. – №. 2.
5. Карпов Л. Е., Юдин В. Н. Методы добычи данных при построении локальной метрики в системах вывода по прецедентам //Препринт/ИСП РАН).–Режим доступа: [http://citforum.univ.kiev.ua/consulting/BI/data\\_mining/5.shtml](http://citforum.univ.kiev.ua/consulting/BI/data_mining/5.shtml). – 2006.

6. Кравченко Ю. А. Метод создания математических моделей принятия решений в многоагентных подсистемах // Известия ЮФУ. Технические науки. – 2011. – №. 7. – С. 120.
7. Спицын В. Г., Цой Ю.Р. Представление знаний в информационных системах: учебное пособие. - Томск: Изд-во ТПУ, 2007. - 160с.
8. Цветков В.Я. Извлечение знаний для формирования информационных ресурсов. - М.: Госинформобр. 2006. - 158с
9. Цветков В.Я. Геореференция как инструмент анализа и получения знаний // Международный научно-технический и производственный журнал «Науки о Земле». 2011. — №2. - с.63-65
10. Майер-Шенбергер В., Кукьер К. Большие данные: Революция, которая изменит то, как мы живем, работаем и мыслим. – Манн, Иванов и Фербер, 2014 -240с.
11. Коваленко Н.И. Извлечение знаний для интеллектуальных транспортных систем // Перспективы науки и образования- 2014. - №5. – с.45-52.
12. Aamodt A., Plaza E. Case-based reasoning: foundational issues, methodological variations, and system approaches // AI Communications. IOS Press. Vol. 7: 1. 1994. - PP. 39-59
13. Tsvetkov V.Ya. Cognitive information models. // Life Science Journal -2014. - 11(4). – pp.468-471
14. Майоров А.А. Пространственное когнитивное моделирование // Перспективы науки и образования- 2014. - №1. – с.33-37