

АВТОМАТИЗАЦИЯ УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ НА ОСНОВЕ ЯЗЫКА GBPR

Антон Рычков

Аннотация: Рассматриваются возможности разработанного автором языка GBPR (General Business Process Representation) для решения проблем интеграции бизнес-процессов структурных подразделений учреждений и предприятий-партнеров, а также оптимизации бизнес-процессов и их динамической настройки. Представленный язык основывается на *pi*-исчислении и является его расширением, ориентированным на автоматизацию управления бизнес-процессами. Основой языка GBPR, как и *pi*-исчисления, являются информационные каналы, по которым передаются сообщения (сервисные каналы – каналы, отвечающие за обращение к сервису или процессу; маршрутные каналы, отвечающие за маршрутизацию потоков управления и данных внутри процесса). Базовой структурой языка GBPR является действие, выполняющее обращение к сервису. Процесс состоит из множества действий, выполняемых параллельно, последовательность действий реализуется через последовательную передачу сообщений между действиями. Процесс является частным случаем сервиса и может быть вызван как сервис посредством обращения по соответствующему информационному каналу. В статье приведено описание языка, его возможностей, в частности, показана возможность процессной интеграции, основанной на ретрансляции сообщений и отношении наследования процессов.

Ключевые слова: бизнес-процесс, автоматизация управления, *pi*-исчисление, наследование, интеграция, оптимизация.

ACM Classification Keywords: H.4 Information Systems Applications: H.4.1 Office Automation – Workflow management.

Conference: The paper is selected from Seventh International Conference on Information Research and Applications – i.Tech 2009, Varna, Bulgaria, June-July 2009

Введение

В настоящее время в области автоматизации управления существуют нерешенные проблемы *интеграции* на основе общих бизнес-процессов различных предприятий с учетом специфики каждого из участников интеграции. Существуют также проблемы *частной оптимизации бизнес-процессов* (оптимизации частных случаев реализации) и *динамического изменения* версий процесса.

Процессно-ролевой подход к управлению является основой современной эффективной организации. Автоматизируются процессы (Workflow Management, WfM) и функции по их управлению (Business Process Management, BPM). Современные информационные технологии позволяют создавать приложения по модели бизнес-процессов, собирая их функциональность из отдельных «кубиков» (Service Oriented Architecture, SOA). Интеграция партнеров в системах B2B (Business-to-Business) также выполняется на основе процессов. Все это приводит к повышению степени важности модели бизнес-процессов организации и повышенным требованиям к скорости ее изменения.

Управление бизнес-процессами

Технология BPM является комплексной и включает в себя WfM, SOA и B2B. Действия, которые составляют управление бизнес-процессами, можно разделить на несколько групп: разработка, моделирование, выполнение, мониторинг и оптимизация. Управление фокусируется на повышении эффективности бизнес-процессов, приносящих «прибыль» организации, и их непрерывной оптимизации.

Таким образом, важными аспектами являются возможности моделирования бизнес-процессов, скорость их внедрения (изменения) и мониторинг показателей.

Современные языки описания бизнес-процессов ориентируются на возможность последующей автоматизации управления этими бизнес-процессами. Для целей автоматизации управления были созданы языки BPEL и BPML, а языки, созданные ранее, были дополнены (ARIS).

Бизнес-процессы организации включают в себя и межорганизационное взаимодействие, которое также может быть описано как процесс, в котором участвуют две и более организаций. Автоматизация таких процессов требует технической реализации обмена данными. В настоящее время стандартом такого рода взаимодействия являются Web-сервисы. Более того, Web-сервисы стали стандартом и для взаимодействия различных информационных систем внутри организации, что позволяет создавать программные приложения на основе существующих бизнес-процессов организации (SOA). Возможности использования Web-сервисов в бизнес-процессах описаны в работах [1, 2].

Web-сервис реализует определенную функциональность, скрывая реализацию этой функциональности, однако в ряде случаев требуется более тесная интеграция, позволяющая организациям или подразделениям разделять один экземпляр бизнес-процесса и контролировать его выполнение от начала до конца. В этом случае Web-сервисы могут предоставлять протоколы нижнего уровня для разделения экземпляра бизнес-процесса между различными информационными системами одной или нескольких организаций, вовлеченных в выполнение процесса. Однако остается проблема реализации методов взаимодействия на верхнем уровне.

Интеграция организаций привела к разделению описания бизнес-процессов на две группы:

- внешние процессы описывают процессы взаимодействия с внешними источниками (другие организации или клиенты);
- внутренние процессы описывают реальные процессы организации, являются более детальными.

Внешние процессы организации являются «видимой» стороной выполнения *внутренних процессов*. Иными словами, внешние процессы описывают внешний интерфейс внутренних процессов организации. Описание внешних процессов открыто для партнеров, в то время как описание внутренних процессов зачастую является коммерческой тайной, так как определяет конкурентное преимущество.

Такое разделение описания процессов требует создания *технологии верификации* соответствия внутреннего процесса внешнему. Одним из вариантов решения этой проблемы является *механизм наследования процессов*. Отношение наследования в данном случае отличается от аналогичного отношения в UML. Так в UML наследуются свойства и методы класса – статические характеристики, в то время как для процесса большее значение имеет поведение – динамическая характеристика. Производный процесс должен поддерживать поведение базового процесса с возможностью его расширения. Таким образом, производный процесс должен выполнять все действия базового процесса в порядке, определенном для базового процесса.

Наследование бизнес-процессов также позволяет решать такие проблемы как:

- проблема динамического изменения – возникает при внедрении новой версии существующего процесса и требует миграции существующих экземпляров процесса на новую версию;
- проблема агрегации управленческой информации выполняющихся процессов в случае наличия различных версий этих процессов [3];
- проблема частной оптимизации – для процесса существуют различные варианты, оптимизированные под частные случаи общего процесса (например, «обслуживание клиента» и такие варианты как «обслуживание частного лица» и «обслуживание организации») [4].

Описание языка GBPR

Для решения приведенных выше задач был разработан язык описания бизнес-процессов GBPR (General Business Process Representation) [7]. Данный язык основывается на *пи-исчислении* [5, 6] и является его расширением, ориентированным на автоматизацию управления бизнес-процессами.

Основой языка GBPR, как и *пи-исчисления*, являются *информационные каналы*, по которым передаются сообщения, но разделяются на группы:

- *сервисные каналы* – каналы, отвечающие за обращение к сервису или процессу;
- *маршрутные каналы* – каналы, отвечающие за маршрутизацию потоков управления и данных внутри процесса.

Базовой структурой языка GBPR является *действие*, выполняющее обращение к сервису:

$$ACTIVITY ::= MESSAGEMAP.[CONDITION] SERVICE(PARAMS)$$

Описание действия включает в себя карту необходимых сообщений (маршрутизация), условие (проверка данных) и обращение к сервису (с указанием сервиса и передаваемых параметров). Так как взаимодействие с сервисами является асинхронным (сервис не может передать ответное сообщение по каналу обращения), то в параметрах указывается канал для ответа в случае, если он требуется.

Карта сообщений состоит из информационных каналов, по которым требуется получить сообщение, связанных различными операциями:

- операция *суммирования* «+» определяет возможные альтернативы (иными словами, для вызова сервиса необходимо получить сообщение *по одному* (не важно по которому) из каналов);
- операция *распараллеливания* «|» определяет список каналов, по которым ожидаются сообщения, и для обращения к сервису необходимо получить сообщения *по всем* каналам из списка.

Описание сервисов в языке GBPR определяет информационный канал для обращения, список необходимых параметров и правила обращения (протокол, адрес и другая техническая информация).

Процесс является частным случаем сервиса, и может быть вызван как сервис посредством обращения по соответствующему информационному каналу. Процесс состоит из множества действий, выполняемых параллельно, последовательное выполнение действий реализуется через последовательную передачу сообщений между действиями. Процесс может содержать блоки обработки исключительных ситуаций и блок завершения.

$$PROCESS ::= ACTIVITY | \{ACTIVITY\}$$

GBPR удовлетворяет большинству шаблонов, описанных в работах [9, 10]. Данные шаблоны описывают встречающиеся ситуации в описании маршрутизации бизнес-процессов. Описания, созданные на других наиболее распространенных языках моделирования бизнес-процессов, могут быть преобразованы в GBPR-описания, как это показано в [8].

Интеграция на основе процессов в GBPR

Как уже говорилось выше, интеграция организаций (или подразделений одной организации) на основе бизнес-процессов является актуальной задачей.

Поскольку маршрутизация потоков управления и данных в языке GBPR выполняется на основе передачи сообщений по информационным каналам, то процесс интеграции сводится к *ретрансляции сообщений* во все системы-участники процесса.

Рассмотрим пример процесса ремонта автомобиля (рис. 1). В процесс вовлечены автосервис «А» и страховая компания «В»:

start. «Приезд клиента» (e1)

e1() + e3(«Состояние»). [«Состояние» != 'Согласовано'] «Составление калькуляции» (e2)

e2(). «Согласование калькуляции» (e3)

e3(«Состояние»). [«Состояние» = 'Согласовано'] «Ремонт автомобиля» (e4)

e4(). «Оплата ремонта» (finish)

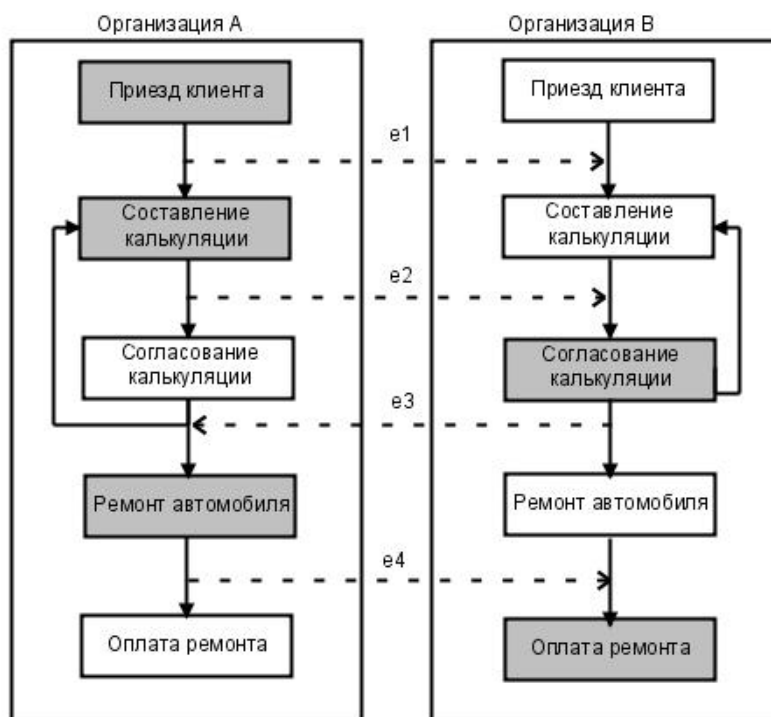


Рис. 1. Процессная интеграция в GBPR

На рис. 1 показано, как ретрансляция сообщений позволяет поддерживать оба экземпляра процесса в идентичном состоянии. При этом могут быть выбраны *различные схемы ретрансляции сообщений*, как *централизованные* (выделенный центральный узел, отвечающий за ретрансляцию), так и *децентрализованные* (система, в которой сгенерировано сообщение, передает его в каждую из систем-участниц процесса). В большинстве случаев подойдет централизованная схема ретрансляции, где в качестве центрального узла выступает система, породившая процесс. Такой подход позволяет снизить издержки, связанные с содержанием каналов для ретрансляции.

Наследование процессов в GBPR

Введение отношения наследования позволяет решить ряд проблем, в том числе и уточнения общего процесса в случае процессной интеграции, за счет чего учитывается специфика каждой организационной участницы процесса и сохраняется в секрете описание внутреннего процесса.

Процесс можно определить как набор работ, набор ресурсов и набор правил. Для упрощения мы не будем рассматривать ресурсную составляющую в описании бизнес-процессов, тогда процесс можно разделить на набор сервисов, выполняемых в бизнес-процессе, и набор правил, определяющих порядок выполнения этих сервисов.

Представим описание процесса как *класс*, содержащий описание необходимых параметров, правил маршрутизации и связей с сервисами, и определим отношение наследования.

Параметры процесса могут быть дополнены и уточнены, иными словами, могут появиться дополнительные параметры, а тип существующих параметров может быть изменен на производный от указанного. Уточнение возможно за счет реализации отношения наследования для классов, определенного в UML.

Отношение наследования определяется в рамках пи-исчисления. При этом, так как сервисы являются внешними по отношению к процессу, то они моделируются посредством исполнителя [5]:

$$Exec(x) ::= x(y).y^{\wedge}.$$

Если процесс Q находится в бинарном отношении наследования от процесса P ($P \leq Q$), то:

- $fn(P)$ принадлежит $fn(Q)$ – все свободные имена процесса P также являются свободными именами процесса Q ;
- для любой транзакции $a(x)$, такой что $P \rightarrow P'$ под действием $a(x)$, найдется такой процесс Q' и последовательность транзакций a_1, a_2, \dots, a_n , таких, что для любого y из области определения x справедливо $Q \rightarrow Q'$ под действием $a_1, a_2, \dots, a_n, a(x)$ и $P'\{y/x\} \leq Q'\{y/x\}$, транзакции a_1, a_2, \dots, a_n могут быть входящими $a(x)$ или исходящими $a^{\wedge}(x)$, транзакция $a(x)$ – входящая, и a является свободным именем для процессов P и Q или является сервисным каналом этих процессов;
- для любой транзакции $a(x)$, такой что $P \rightarrow P'$ под действием $a(x)$, найдется такой процесс Q' и последовательность транзакций a_1, a_2, \dots, a_n , таких, что для любого y из области определения x , справедливо $Q \rightarrow Q'$ под действием $a_1, a_2, \dots, a_n, a(x)$ и $P' \leq Q'$, транзакции a_1, a_2, \dots, a_n могут быть входящими $a(x)$ или исходящими $a^{\wedge}(x)$, транзакция $a(x)$ – любая транзакция, кроме случая из предыдущего пункта;
- для любой транзакции $a(x)$, такой что $Q \rightarrow Q'$ под действием $a(x)$ и не найдется процесс P' , такой, что справедливо $P \rightarrow P'$ под действием $a(x)$, тогда для любого y из области определения x выполняется $P \leq Q'\{y/x\}$, транзакция $a(x)$ – входящая, и a является свободным именем для процесса Q или сервисным каналом этого процесса;
- для любой транзакции $a(x)$, такой что $Q \rightarrow Q'$ под действием $a(x)$ и не найдется процесс P' , такой, что справедливо $P \rightarrow P'$ под действием $a(x)$, тогда для любого y из области определения x выполняется $P \leq Q'$, транзакция $a(x)$ – любая транзакция, кроме случая из предыдущего пункта.

Отношение наследования *не является симметричным*, иными словами, из $P \leq Q$ не следует $Q \leq P$.

С учетом отношения мы можем построить *иерархию процессов*, на самом верхнем уровне которой будет *абстрактный процесс*. Такая иерархия помогает управлять вариантами процесса, определяя частные случаи как производные от общего.

Рассмотрим пример процесса «обращение» и производного от него процесса «обращение от организации»:

```
«обращение» ::= start. «прием обращения» (e1) |
              e1(). «оформление заказ-наряда» (e2) |
              e2(). «оказание услуг» (finish)

«Обращение от организации» ::= start. «прием обращения» (e1) |
                               e1(). «оформление договора» (e2) |
                               e2(). «оформление заказ-наряда» (e3) |
                               e3(). «оказание услуг» (finish)
```

Процесс «обращение» находится на первом уровне иерархии процессов, в то время как процесс «обращение от организации» является производным от него и находится на втором уровне. Аналогично,

если есть производный процесс «обращение от физического лица», то оно также будет находится на втором уровне. Следует заметить, что изменение базового процесса (*эволюционная оптимизация*) не всегда приводит к перепроектированию производных процессов, а создание варианта процесса не приводит к изменению базового процесса. Это упрощает частную оптимизацию (оптимизацию частных случаев процесса).

Аналогично может быть использовано отношение наследования в случае процессной интеграции для уточнения разделяемого процесса с учетом специфики каждой из организаций-участниц интеграции.

Заключение

Язык GBPR позволяет проводить основанную на процессах интеграцию организаций с возможностью контроля всего процесса. Асинхронная передача сообщений, лежащая в основе GBPR, позволяет использовать различные протоколы передачи данных, что придает гибкость интеграции. Возможность прозрачного уточнения общего процесса благодаря отношению наследования для процессов позволяет учесть специфику участников интеграции без ущерба для общего процесса. Наследование помогает также в проведении частной оптимизации и решении проблем динамического изменения версий процессов.

Благодарности

Статья частично финансированна из проекта **ITHEA XXI** Института Информационных теории и Приложений FOI ITHEA и Консорциума FOI Bulgaria (www.ithea.org, www.foibg.com).

Библиографический список

- [1] Feng G., Wang C., Li H. Web services based cross-organizational business process management // Web Technologies Research and Development – APWeb 2005. Springer Berlin, Heidelberg, 2005.
- [2] Papazoglou M., Yang J. Design methodology for Web services and business processes // Technologies for E-Services. Springer Berlin, Heidelberg, 2002.
- [3] van der Aalst W.M.P. Inheritance of business processes: a journey visiting four notorious problems // Petri Net Technology for Communication-Based Systems. Springer Berlin, Heidelberg, 2004.
- [4] Hallerbach A., Bauer T., Reichert M. Managing process variants in the process life cycle // Technical report. University of Twente, Enschede, 2008.
- [5] Milner R., Parrow J., Walker D. A calculus of mobile processes, part 1 // Information and Computation. Academic Press, Duluth, 1992.
- [6] Milner R., Parrow J., Walker D. A calculus of mobile processes, part 2 // Information and Computation. Academic Press, Duluth, 1992.
- [7] Рычков А.Ю. Язык описания бизнес-процессов на основе пи-исчисления // Математика программных систем: Межвузовский сборник научных трудов / Перм. ун-т. Пермь, 2008.
- [8] Рычков А.Ю. Преобразование описаний бизнес-процессов в представление на языке GBPR // Математика программных систем: Межвузовский сборник научных трудов / Перм. ун-т. Пермь, 2008.
- [9] van der Aalst W.M.P., ter Hofstede A.H.M., Kiepuszewski B., Barros A.P. Workflow patterns // Distributed and Parallel Databases, vol.14, num.1. Springer Netherlands, 2004.
- [10] van der Aalst W.M.P., ter Hofstede A.H.M., Kiepuszewski B., Barros A.P. Advanced workflow patterns // Cooperative information systems: 7th International Conference, CoopIS 2000. Springer-Verlag, New York, 2000.

Сведения об авторе

Антон Рычков – аспирант кафедры математического обеспечения вычислительных систем, Пермский государственный университет; ул. Букирева, д. 15, г. Пермь, 614990, Россия;
e-mail: anton@rychkov.biz