

УДК 303.732.4

UDC 303.732.4

**ПРИМЕНЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ШЛЮЗА ДЛЯ ИНТЕГРАЦИИ БИЛЛИНГОВОЙ СИСТЕМЫ И IN-ПЛАТФОРМЫ ПОСРЕДСТВОМ МЕТОДА АВТОМАТИЗИРОВАННОГО ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ УПРАВЛЯЮЩИХ ПРАВИЛ****APPLICATION OF INTELLIGENT GATEWAY FOR INTEGRATION BILLING SYSTEM AND THE IN-PLATFORM BY THE METHOD OF AUTOMATED DECISION ON THE BASIS OF THE RULES GOVERNING**

Кривенко Александр Анатольевич  
Кубанский государственный технологический университет, Краснодар, Московская, 2а  
[alexander.a.krivenko@gmail.com](mailto:alexander.a.krivenko@gmail.com)

Krivenko Aleksandr Anatolyevich  
Kuban State Technological University, Krasnodar, Moscow, 2a  
[alexander.a.krivenko@gmail.com](mailto:alexander.a.krivenko@gmail.com)

В статье рассмотрена общая проблема интеграции биллинговой системы с внешними программными комплексами и проанализированы основные способы решения проблемы на примере конкретной задачи. Проводится сравнительная характеристика традиционных методов, делается вывод о необходимости модификации наилучшего из них. Описывается математическая модель метода, приводится детальное описание архитектуры интеллектуального шлюза. В качестве демонстрации эффективности решения предлагается сравнительный анализ показателей производительности системы и альтернативных вариантов. В заключение делается вывод о более широком применении разработанной системы, а также потенциальных возможностях ее модификации

In this article we consider the general problem of integration with external billing system software complexes and analyze the main methods of the solution of the problem on the example of a specific task. We also carried out the comparative analysis of the traditional methods and came to the need of modifying the best of them. The mathematical model of the method is given, and the detailed description of the architecture of intelligent gateway is shown. To demonstrate the effectiveness of the solution it is proposed to do a comparative analysis of system's performance and alternative variants. At last, the conclusion of the wider applicability of the developed system, as well as the potential modifications of it, is made

Ключевые слова: СИСТЕМНАЯ ИНТЕГРАЦИЯ, БИЛЛИНГОВАЯ СИСТЕМА, IN-ПЛАТФОРМА, ИНТЕЛЛЕКТУАЛЬНЫЙ ШЛЮЗ, WORKFLOW, УПРАВЛЯЮЩИЕ ПРАВИЛА, XML, BPMN

Keywords: SYSTEM INTEGRATION, BILLING SYSTEM, IN-PLATFORM, INTELLIGENT GATEWAY, WORKFLOW, REGULATIONS RULES, XML, BPMN

Системная интеграция – это разработка комплексных решений по автоматизации бизнес-процессов предприятия. Ее конечная цель – максимально эффективное управление организацией. Интеграционные решения ориентированы на организацию информационной поддержки выполнения бизнес-задач, стоящих перед учреждением, обеспечивая прозрачность, последовательность и надежность взаимодействия разнородных прикладных информационных систем, работающих в рамках единого бизнес-процесса. В общем случае системная интеграция является важным инструментом в любой IT-индустрии. В настоящее время она развивается ускоренными темпами. При выборе метода интеграции

следует руководствоваться специализацией приложений и подсистем, являющихся объектами интеграции.

В статье описывается процесс исследования и определяется актуальность работы, формируются предметная область, цель и задачи исследования, а также выдвигаются основные требования к определению метода решения. Проводится аналитический обзор традиционных методов, а также приводится подробное описание развития и модификации приоритетного метода. Кроме того, статья содержит: обоснование выбора инструментальных средств, использованных при разработке, условия необходимые для внедрения и использования решения, оценку эффективности, ограничения и возможные пути их преодоления, а также вывод о более широкой применимости решения.

### **Состояние исследований и актуальность работы**

В рамках научно-исследовательской задачи возникла необходимость интегрировать в единый комплекс две масштабные системы. Первая – телекоммуникационная биллинговая система представляет собой сложный программный комплекс, осуществляющий контроль счетов абонентов сотовой связи, управление подключением, отключением дополнительных услуг, выполнение различных балансовых корректировок и т.п. Она, являясь сущностью высокого уровня и осуществляя управляющее воздействие, нуждается в управляемом объекте [6]. В качестве нее выступает вторая система – IN-платформа, представляющая собой программно-аппаратный комплекс, осуществляющий хранение основных абонентских данных, их счетов и т.д., а также предоставляющая различные способы для управления ими на относительно более низком уровне. С учетом объемов данных, с которыми приходится работать обеим системам, к методам управления данными предъявляются жесткие требования надежности, отказоустойчивости и безопасности. Методы управления

должны поддерживать как возможность непосредственного управления пользователем, так и систем автоматизированного управления.

За основное условие было принято то, что биллинговая система и IN-платформа являются продуктами разных компаний, между ними не предусмотрена прямая связь, позволившая бы им эффективно взаимодействовать друг с другом. Однако так как большинство компаний-производителей IN-платформ позиционируют свой продукт, как универсальный, было учтено дополнительное условие, заключающееся в том, что IN-платформа предоставляет способы взаимодействия с ней [9].

### **Аналитический обзор традиционных методов**

Для взаимодействия приложений обычно используются такие методы, как обмен файлами, общая база данных, удаленный вызов и асинхронный обмен сообщениями. К этому списку также можно отнести прямой обмен данными между базами данных приложений, хотя этот метод ближе не к интеграции приложений, а к перемещению данных.

В данном случае в качестве метода взаимодействия IN-платформа предоставляет интерфейс на основе web-сервиса, а также обмен файлами определенного формата по протоколу FTP. Web-интерфейс описывается файлами wsdl-схем, предоставляемыми компанией-разработчиком платформы. Они предназначены для оказания какого-либо «единичного» воздействия, т.е. управления изменением или получением данных одного абонента. Обмен файлами специализированного формата по протоколу ftp реализует возможность для оказания массового управляющего воздействия на группу абонентских данных на стороне IN-платформы посредством одного запроса.

Обмен файлами, пожалуй, самый распространенный подход к организации взаимодействия. Это связано с относительной простотой реализации, а также существованием стандартных (или «почти»

стандартных) форматов обмена. Например, большая часть корпоративных информационных систем позволяет загружать и выгружать файлы, например, в формате CSV (Comma-Separated Values – «поля, разделенные запятыми»). Однако у этого подхода есть и недостатки; если необходимо оперировать сложными структурами, то простые форматы обмена уже не пригодны. Возникающие в таких случаях специализированные форматы файлов должны «понимать» взаимодействующие системы, что ведет к жесткой зависимости систем друг от друга. Этот недостаток обычно преодолевают всевозможными утилитами конвертации данных. Кроме того, обычно обмен файлами подразумевает участие человека – кто-то должен выгрузить файл, скопировать его на другой компьютер, загрузить. Однако если интегрируемые методом обмена файлами системы имеют возможность автоматической загрузки / выгрузки (например, по расписанию), то данный подход позволяет построить полностью автоматизированное решение, которое вследствие своей простоты обладает высокой надежностью и пропускной способностью.

Другой распространенный способ интеграции приложений – это транспорт на основе очередей сообщений и программное обеспечение промежуточного слоя, ориентированное на обработку сообщений, которое выполняет роль сетевого концентратора. Однако подобный метод можно считать устаревшим, а инструментарий промежуточного слоя чрезмерно дорогостоящим в разработке.

После появления метода межплатформного взаимодействия, Web-сервисов возник новый подход к организации основы для интеграции – сервисная шина предприятия (Enterprise Service Bus, ESB). Реальным отличием ESB от шины сообщений предприятия стала поддержка дополнительных методов взаимодействия между приложениями и хабом, прежде всего, протокола SOAP. Основанный на ESB подход подавался как более дешевая, простая в реализации альтернатива предыдущей

концепции. Архитектура, ориентированная на сервисы (Service-Oriented Architecture, SOA), – это концепция построения информационных систем путем представления их в виде сервисов, доступных для «наружного» использования, в том числе путем их интеграции с другими приложениями.

### **Постановка и решение задачи**

Необходимость реализации шлюза для интеграции биллинговой системы с новой IN-платформой вытекает из двух оснований. Во-первых, на данный момент они абсолютно не совместимы, во-вторых, изменение существующих модулей биллинговой системы с целью их универсализации экономически не эффективно. В пользу реализации интеллектуального шлюза, позволяющего системам эффективно взаимодействовать, выступает еще одно обстоятельство: несмотря на то, что область применимости систем одна и та же – телекоммуникационные технологии, а именно мобильная связь, набор понятий и методов IN-платформы хоть и совпадает в общем случае с понятиями биллинговой системы по ключевым сущностям и методам, но на практике не все сообщения биллинговой системы можно связать с web-методами IN-платформы связью «один к одному». Некоторые сообщения биллинговой системы подразумевают вызов нескольких методов IN-платформы, зачастую, с передачей результатов предыдущего метода в качестве параметров следующего, другие сообщения биллинговой системы, напротив, менее информативные, чем их возможные аналоги IN-платформы, третьи не находят отображения вовсе. Таким образом, очевидна необходимость разработки некой прослойки, позволяющей биллинговой системе в автоматизированном режиме оказывать адекватное управляющее воздействие на IN-платформу. Для этой задачи принято решение реализовать интеллектуальный шлюз, базой которого является,

специально разработанная модификация технологии системы принятия решений на основе управляющих правил.

### **Развитие и модификация наилучшего из методов**

Интеллектуальный шлюз состоит из следующих модулей: модуль коммуникации с биллинговой системой, модуль коммуникации с IN-платформой, модуль выбора управляющих правил, модуль обработки правил преобразования сообщений-запросов, модуль, обработки правил преобразования сообщений-ответов, модуль внутренней логики шлюза, модуль агрегирования и трансформации данных, модуль маршрутизации сервисов IN-платформы, модуль доступа к данным системы, модуль сбора статистической информации и логирования.

Взаимодействие между шлюзом и биллинговой платформой осуществляется синхронно с использованием технологии Net Remoting. При передаче данных от биллинговой системы к IN-платформе синхронно вызывается метод шлюза, при асинхронном вызове – IN-платформой шлюза, шлюз синхронно обращается к биллинговой системе. Взаимодействие интеллектуального шлюза и IN-платформы, при вызове ее методов и биллинговой системе осуществляется посредством обращения к методам Web-сервисов, предоставляемых IN-платформой [3]. Биллинговая система сама определяет необходимость синхронного или асинхронного получения ответов от IN-платформы, вызывая соответственно метод для синхронного или асинхронного взаимодействия у интеллектуального шлюза.

База правил содержит следующую информацию:

— Правила проецирования «понятий» биллинговой системы на «понятия» IN-платформы (Содержат информацию о том, как соотносятся сообщения из «справочников» биллинговой системы с методами web-интерфейсов IN-платформы. Возможны следующие варианты: одному типу сообщения БС соответствует один метод платформы; одному типу

сообщения БС соответствует несколько методов платформы, причем логика их вызова может быть различной; нет соответствия методов платформы типу сообщения БС, используется эмуляция ответа).

— Шаблоны предварительных преобразований сообщений от биллинговой системы;

— Правила формирования сообщений-запросов;

— Правила формирования сообщений-ответов;

— Правила агрегирования и трансформации данных;

— Шаблоны эмуляции ответов от IN-платформы;

— Правила маршрутизации сообщений БС по сервисам IN-платформы.

Как было описано ранее, в основе работы шлюза, как объекта управляющего воздействия, лежит набор правил управляющего воздействия, которые, в свою очередь, условно делятся на две группы: правила-сценарии и правила-шаблоны.

В сущности, комплекс сценариев работы интеллектуального шлюза, а также модель их выполнения представляют собой пример специализированного workflow. Workflow – это процесс, произвольное задание, выполняемое последовательно или параллельно двумя или более участниками рабочей группы с целью достижения общей цели.

- Правила-шаблоны отвечают за непосредственное преобразование «понятий» биллинговой системы в «понятия» IN-платформы. В основе шаблонов преобразований лежат технологии XPath, XSLT и регулярные выражения.

Возможности XSLT и XPath наиболее эффективно применяются при преобразовании сообщений от IN-платформы во внутренние объекты контекста сценария шлюза. Поскольку сервисы IN-платформы являются набором web-методов, то на низком уровне не составляет труда получать синхронные ответы с платформы в виде SOAP-сообщения, которые в свою

очередь являются оберткой над базовыми XML-сообщениями. Кроме того, в контексте сценария также удобно хранить объекты в виде их XML-сериализаций, это значительно упрощает структуру модели контекста сценария, а также позволяют экономить процессорное время на доступ к элементам таких объектов [7].

### **Математическая модель системы интеграции данных**

Предлагается в качестве средства конвертирования данных использовать универсальный конвертор, работающий с XML. Схемой отображения данных в этом случае выступает шаблон преобразования, который и задает соответствие полей в одной структуре полям в другой структуре [7]. Для задания нестандартных соответствий лицу принимающему решение предоставляется возможность вписывать в шаблон преобразования собственные алгоритмические конструкции на специальных языках программирования. Далее представлена математическая модель преобразования данных и шаблонов преобразования данных, выводятся ограничения на данные и структуры.

Пусть задано множество объектов (сущностей)  $O = \{o_1, o_2, \dots, o_n\}$ , при этом каждому объекту  $o_i$  соответствует набор свойств из множества  $P = \{p_1, p_2, \dots, p_n\}$ , то есть у объекта  $o_i$  набор свойств –  $p_i$ ,  $i = (1, n)$ . Сам набор свойств представляет собой кортеж полей  $p_i = (e_{i1}, e_{i2}, \dots, e_{im})$ . Каждое поле состоит из пары  $e_{ij} = (v_{ij}, s_{ij})$ ,

где

$v_{ij}$  – значение поля,

$s_{ij}$  – формат поля (размерность, тип).

Связи между объектами задаются элементами множества  $L = \{l_1, l_2, \dots, l_k\}$ , где каждый элемент является тройкой  $l_j = (o_{j1}, o_{j2}, r_j)$ , указывая на два связанных между собой объекта и наименование (вид) роли, по которой они связываются [2].



Иерархия (дерево вложенности) объектов в конкретном файле данных описывается множеством  $T=\{t_1, t_2, \dots, t_n\}$ , где каждый элемент – это пара  $t_i=(o_i, o_j)$ ,  $i=(1, n)$ , в которой первый элемент соответствует объекту с тем же индексом, а второй указывает на объект, который является родительским по отношению к данному в иерархии (т.е.  $o_i$  непосредственно вложен в  $o_j$ ) [4].

Совокупность всех указанных выше множеств  $U = \{O, P, L, T\}$  образует пакет данных в определенном формате. Тогда шаблон преобразования  $F$  – это отображение двух множеств  $U$ , т.е.  $U^1 @ U^2$ .

Показано, что множество  $T$  можно считать единообразным, т.е. когда  $t_i=(o_i, \Pi)$ ,  $i=(1, n)$ . Предполагается, что верхний индекс  $i$  у объектов, их свойств, связей и т.п. означает соответствующую совокупность  $U^i$ .

В общем случае каждому объекту из  $U^1$  может соответствовать произвольное количество объектов из  $U^2$ , и наоборот: каждому объекту из  $U^2$  может соответствовать произвольное количество объектов из  $U^1$ . Это связано с тем, что в соответствие ставятся не сами объекты, а их свойства. Следует отметить, что одному свойству одного объекта в  $U^1$  может соответствовать набор свойств одного или нескольких объектов в  $U^2$  (обратное также верно).

Таким образом, отображение раскрывается следующим образом:

$$F(o_i^1) = F(e_{i1}^1, e_{i2}^1, \dots, e_{im}^1) = F((v_{i1}^1, s_{i1}^1), (v_{i2}^1, s_{i2}^1), \dots, (v_{im}^1, s_{im}^1)) = ((v_{j1}^2, s_{j1}^2), (v_{j2}^2, s_{j2}^2), \dots, (v_{ju}^2, s_{ju}^2)).$$

Учитывая вышеизложенное, можно определить отображение  $F$  (шаблон преобразования) как набор условий-связей пространств  $F=(f_1, f_2, \dots, f_x)$ . Данные условия-связи устанавливают соответствие между свойствами объектов и связями между объектами.

Каждое такое условие состоит из тройки: элемента  $U^1$ , элемента  $U^2$  и дополнительного условия (функции преобразования). В качестве рассматриваемого элемента может выступать конкретное свойство объекта

или связь между объектами, а в качестве дополнительного условия может значиться изменение формата, добавление дополнительных символов. Отметим, что условия могут быть разными и не обязательно взаимобратными при прямом и обратном преобразовании. Каждая связь-тройка описывается следующим образом (табл. 1).

Таблица 1 – Формальное представление видов связей

Вид связи-тройки	Формальное представление
Один-к-одному	$f_q = ((v_{ij}^1, s_{ij}^1), (v_{kl}^2, s_{kl}^2), h_q)$ или $f_q = (l_i^1, l_k^1, h_q)$
Один-ко-многим	$f_q = ((v_{ij}^1, s_{ij}^1), ((v_{kl1}^2, s_{kl1}^2), (v_{kl2}^2, s_{kl2}^2), \dots, (v_{klw}^2, s_{klw}^2)), h_q)$ или $f_q = (l_i^1, (l_{k1}^2, l_{k2}^2, \dots, l_{kw}^2), h_q)$
Многие-к-одному	$f_q = (((v_{ij1}^1, s_{ij1}^1), (v_{ij2}^1, s_{ij2}^1), \dots, (v_{ijw}^1, s_{ijw}^1)), (v_{kl}^2, s_{kl}^2), h_q)$ или $f_q = (l_{i1}^1, l_{i2}^1, \dots, l_{iw}^1, l_k^2, h_q)$

где  $f_q \in F, q=(1.x); h_q \in H, q=(1.x), H$  – множество функций преобразования.

Таким образом, все  $f_q$  можно разделить на два вида: связывающие свойства объектов (что приводит к связи самих объектов между собой) и связывающие связи между объектами. Отдельно описывается роль и место функций преобразования, заданных специальными алгоритмами. Рассматривается расширение  $f_q \in F$ , обеспечивающее возможность двустороннего преобразования данных с использованием шаблона преобразования.

### Условия внедрения и использования

Для установки компонентов системы необходимы сервер начального уровня, процессор не ниже 2ГГц, ОЗУ не менее 2Гб, HDD не менее 50Гб, с сетевым интерфейсом не менее 2Мбит, желательно, с raid-1 для обеспечения надежности. На сервере должна быть установлена ОС Windows XP\Vista\7 либо Windows Server 2003\2008, .Net Framework 2.0

или выше. В случае использования подключения к СУБД требуется предустановленное клиентское ПО Oracle 9 или Oracle 11.

Установка системы не требует глубоких специальных навыков, т.к. экземпляр распаковывается и устанавливается с помощью кумулятивного инсталлятора, однако для настройки масштабируемых модулей и взаимодействия между ними необходимо привлечение специалиста владеющего навыками исполнения sql-скриптов, а также владеющего технологией XML.

### **Оценка эффективности предложенных решений**

Оценить эффективность внедрения разработанной системы достаточно сложно, поскольку начало проектирования было инициировано необходимостью удовлетворить потребности более масштабных систем в рамках конкретной задачи. До сего момента аналогичные функции не выполнялись какой-либо другой системой или с помощью оператора. Однако, для демонстрации эффективности системы можно провести сравнительные исследования. В качестве альтернативных решений можно использовать те методы, от которых решено было отказаться на этапе проектирования. В частности это прямой вызов web-метода IN-платформы из биллинговой системы посредством созданного прокси-класса, а также оказания управляющего воздействия оператором на IN-платформу по результатам мониторинга выходных сообщений биллинговой системы.

Оба альтернативных метода требуют значительных временных затрат на реализацию в полном объеме, однако для демонстрационных целей можно упростить задачу, сведя ее к минимуму. Так, в первом случае был разработан один единственный прокси-класс, связывающий интерфейс биллинговой системы, разработанный в рамках реализации описанного в статье метода решения проблемы, и соответствующий метод IN-платформы (для простоты выбран по принципу связи один-к-одному).

Во втором случае потребовалось разработать приложение с пользовательским интерфейсом, которое позволяет анализировать сообщения из таблиц базы данных, которую заполняет биллинговая система. Каждая запись таблицы идентифицирует метод, который биллинговая система должна вызвать у IN-платформы, а также содержит параметры вызова. Задача оператора для унификации эксперимента сведена к просмотру выводимой в режиме реального времени информации и выбору соответствующего каждой записи метода из списка путем анализа отображаемой информации (для простоты реализована возможность только того метода, который используется в эксперименте с прокси-классом).

На вход модуля биллинговой системы, отвечающей за обращение к IN-платформе, подается одинаковое количество сообщений, анализ времени в секундах, затраченного на обработку запросов, производится с помощью логирования в моменты времени получения сообщения и вызова соответствующего метода IN-платформы.

В таблице 2 приведены результаты испытаний при обращении биллинговой системы к одному и тому же методу IN-платформы посредством вызова с помощью интеллектуального шлюза, прокси-класса или действий оператора.

Таблица 2 – Временные показатели методов

Число запросов	Интеллектуальный шлюз	Прокси-класс	АРМ оператора
1	0.078	0.034	1.030
10	0.653	0.522	10.120
100	1.123	1.003	100.435
1000	5.231	4.976	-
10000	60.105	53.024	-
100000	645.980	583.323	-

Как видно из данных таблицы 2, время на обработку запросов оператором слишком велико для того, чтобы этот метод рассматривался в

качестве альтернативы. Однако положительная сторона этого метода заключается в том, что оператор может анализировать данные и выбирать нужный метод, в то время как для использования прокси необходим труд разработчика, который создаст все необходимые прокси-классы ко всем методам, а также учтет все виды связей. Кроме того, по мере изменения интерфейсов обеих систем прокси-классы придется переписывать или добавлять новые. Отсутствие возможности простой настройки системы при использовании прокси-классов является ключевым фактором, из-за которого метод был отвергнут. Однако, как видно из данных таблицы 2. временные показатели превосходят результаты работы шлюза. Это вызвано тем, что шлюз затрачивает дополнительное время и ресурсы на выбор нужного правила и применения управляющего воздействия на основе него. Неоспоримым превосходством шлюза является также механизм обработки ответов, который позволяет конфигурировать набор данных, передаваемых в вызывающую систему. Он же дает возможность формировать новое управляющее воздействие на уровне шлюза без возврата ответа в биллинговую систему. В случае с прокси-классом всю логику пришлось бы программировать отдельно для каждого случая, что влечет за собой существенные затраты труда программиста, а в случае оператора обработка ответа представляется практически невозможной из-за значительных временных затрат, которые расходуются на обработку каждого сообщения, в то время как система работает под большой нагрузкой.

### **Выводы, перспективы развития**

Интеллектуальный шлюз, реализованный на основе системы принятия решений с помощью управляющих правил, достаточно универсален для решения широкого круга задач. В частности, он позволяет интегрировать практически любую пару информационных систем, где одна должна оказывать управляющее воздействие на другую.

Реализовав дополнительный набор интерфейсов для получения входных данных, можно расширить функциональность до подключения нескольких систем в качестве управляющих. Кроме того, предусмотренный в шлюзе механизм подключения управляемых систем посредством специальных плагинов-коннекторов позволяет значительно расширить функциональность. Для этого достаточно реализовать набор плагинов, реализующих базовый предопределенный интерфейс, которые бы позволили взаимодействовать с внешними информационными системами по различным протоколам, а не только SOAP и FTP, как это реализовано в настоящее время.

Дополнительно повысить управляемой системы и простоту ее настройки можно, применив более широко используемые механизмы реализации управляющих правил [8], дополнить реализованный формальный язык описания таким образом, чтобы он поддерживал стандартную нотацию описания бизнес-процессов, например BPMN [5].

### **Заключение**

В данной статье проведено исследование методов системной интеграции в телекоммуникационных системах в рамках диссертационной работы. Выделены основные проблемы и обозначены цели, а также сформулирована конкретная задача интеграции биллинговой системы и IN-платформы с учетом правил оказания управляющего воздействия и обратной связи. Проведен анализ методов, потенциально пригодных для решения поставленной задачи. В результате анализа было принято решение о выборе метода взаимодействия на основе архитектуры SOA, однако он в полной мере не удовлетворяет потребностям задачи. Проведена доработка метода путем расширения его собственной реализацией модуля на основе системы принятия решений. В статье приведено подробное описание разработанного модуля и алгоритмов, а

также математическая модель. Далее проведена оценка эффективности метода, в которой производится сравнение разработанного решения с альтернативами. Доказывается превосходство разработанного решения над другими возможными методами, а также делается вывод о более широкой применимости разработанного решения и предлагаются возможные дополнения к нему.

#### Список литературы

1. An introduction to Petri nets. Режим доступа: <http://viking.gmu.edu/http/syst511/vg511/AppC.html>.
2. Фиошин М. Основы  $\pi$ -исчисления. Режим доступа: <http://progr.tsi.lv/research/picalc.pdf>.
3. Van der Aalst, Hofstede A. H. M. YAWL: Yet Another Workflow Language. Режим доступа: [www.citi.qut.edu.au/pubs/technical/yawlrevtech.pdf](http://www.citi.qut.edu.au/pubs/technical/yawlrevtech.pdf).
4. Van der Aalst W. M. P. Pi calculus versus Petri nets: Let us eat “humble pie” rather than further inflate the “Pi hype”. Режим доступа: [tmitwww.tn.tue.nl/staff/wvdaalst/pi-hype.pdf](http://tmitwww.tn.tue.nl/staff/wvdaalst/pi-hype.pdf).
5. Michael zur Muehlen. Process Management Standards Overview. Режим доступа: [www.wfmc.org/standards/docs/Process\\_Management\\_Standards\\_files/frame.htm](http://www.wfmc.org/standards/docs/Process_Management_Standards_files/frame.htm).
6. Коалиция WfMC. Workflow Reference Model. Режим доступа: [www.wfmc.org/standards/docs/TC00-1003\\_10\\_1994.pdf](http://www.wfmc.org/standards/docs/TC00-1003_10_1994.pdf).
7. Коалиция WfMC, язык XPDЛ. Режим доступа: [www.wfmc.org/standards/docs/TC-1025\\_10\\_xpdl\\_102502.pdf](http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf).
8. Коалиция BPMI, язык BPML. Режим доступа: [www.bpmi.org/bpml-spec.esp](http://www.bpmi.org/bpml-spec.esp).
9. Коалиция IBM, Microsoft и BEA, язык BPEL4WS. Режим доступа: <http://www-106.ibm.com/developerworks/library/ws-bpel/>.
10. OMG Unified Modeling Language Specification. Режим доступа: [www.omg.org/docs/formal/03-03-01.pdf](http://www.omg.org/docs/formal/03-03-01.pdf).
11. WfMC. Terminology and glossary. Режим доступа: [www.wfmc.org/standards/docs/TC-1011\\_term\\_glossary\\_v3.pdf](http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf).
12. WfMC. Workflow Standard — Interoperability Wf-XML Binding. Режим доступа: [www.wfmc.org/standards/docs/Wf-XML-11.pdf](http://www.wfmc.org/standards/docs/Wf-XML-11.pdf).