

# Использование механизмов интеграции приложений для автоматизации процессов управления информационными ресурсами

Нестеренко А.К.  
alexn@ccas.ru

Данилина А.А.  
adanilina@yandex.ru

Сысоев Т.М.  
tim@ccas.ru

Бездушный А.Н.  
bezdushn@ccas.ru

Серебряков В.А.  
serebr@ccas.ru

Вычислительный центр им. А.А. Дородницына РАН

## Аннотация

В работе рассматриваются способы автоматизации процессов по манипулированию информационными ресурсами, анализируются механизмы построения систем разработки и управления потоками работ, рассматриваются проблемы формализации описаний таких процессов и их взаимодействия с внешними информационными системами и пользователями, рассматривается возможность применения системы управления потоками работ для задачи формирования данных электронной библиотеки.

## 1 Введение

Процесс выполнения тех или иных видов работ по управлению и обработке информационных ресурсов (документооборот, библиотеки, архивы данных и т.д.) представляет из себя регламентированный набор действий, который необходимо выполнить для достижения необходимого результата. При этом в процессе подготовки входных и выходных артефактов каждого этапа потока работ исполнители используют обширный набор инструментальных программных продуктов для частичной автоматизации своего участка работ. Такая частичная автоматизация «ручной деятельности» конечно имеет ряд положительных моментов, но задача упрощения координации процесса по обработке информационных ресурсов в целом данным подходом не решается. Следующие ресурсоемкие задачи не могут быть решены путем простой автоматизации деятельности сотрудников на местах:

- автоматизированная подготовка входных и выходных артефактов каждого этапа процесса;

- координация потока управления и потока данных;
- полная автоматизация отдельных участков потока работ (способность взаимодействия с «программными» исполнителями заданий в рамках некоторого унифицированного интерфейса);
- возможность быстрого, с минимальными трудозатратами создания нового описания регламента, с возможной модульностью (декомпозицией на подпроцессы) для повторного использования описаний, с поддержкой быстрой и безболезненной для участников процесса модификацией имеющихся регламентов;
- эффективная реакция потока работ на возникновение непредвиденных обстоятельств на пути его выполнения (например, недоступность в данный момент тех или иных ресурсов), в том числе четко регламентированные действия по устранению последствий некорректно выполненного этапа процесса;
- хорошая управляемость процессом с доступом к данным любого активного этапа;
- четкое ролевое разделение участников процесса (в том числе поддержка динамической взаимозаменяемости исполнителей в случае недоступности нужных ресурсов);
- возможность сбора статистики выполнения процесса для последующей оптимизации.

Для эффективного решения перечисленных выше задач большая часть усилий разработчиков программного обеспечения на текущий момент сконцентрирована вокруг теории автоматизированных потоков работ (Workflow[1]) и

---

Труды 8<sup>ой</sup> Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2006, Суздаль, Россия, 2006.

систем, способных эффективно решать задачи их исполнения и координации (Workflow Management Systems[16]). Количество подобных информационных систем (интеграционных платформ, интеграционных серверов), в основу которых на формальном уровне заложена базовая концепция интеграции распределенных ресурсов (как программных систем, так и человеческих ресурсов) для выполнения некоторой общей задачи, увеличивается очень быстрыми темпами. При этом новые решения приводят к появлению новых задач, адресованных системам исполнения потоков работ. К основным из них можно отнести следующие:

- построение автономных систем исполнения потоков работ (вне контекста конкретного варианта их использования) требует строгой формализации описаний потоков работ на некотором языке;
- поскольку основной контингент пользователей подобных систем составляют исполнители на местах, аналитики, менеджеры (т.е. люди неискушенные в программировании маршрута потока работ с использованием некоторых формальных языков), система разработки потоков работ (являющаяся частью интеграционного сервера) должна предоставлять развитые средства по созданию и модификации описаний потоков работ. При этом должен использоваться наиболее интуитивно понятный обычному пользователю способ формирования потоков работ – визуальные диаграммы (так как практически каждый управляющий процессом человек привык использовать для таких целей некоторое прикладное средство, такое как, например, диаграммы активности деятельности UML[2]). Итого, помимо эффективного «языка программирования» потоков работ пользователям должна быть представлена удобная графическая нотация для их описания с достаточным уровнем абстракции;
- интеграционный сервер должен предоставлять создателям описаний потоков работ полный набор средств для проверки их работоспособности до начала опытной эксплуатации. Такие средства должны включать как простые статические верификаторы корректности созданных описаний потоков работ, так и динамические отладчики;
- интерактивные средства взаимодействия с пользователями являются особо важной составляющей частью интеграционного сервера, поскольку именно пользователи управляют ходом выполнения потока работ;
- большую важность имеют задачи обеспечения безопасности данных и поддержки транзакционности автоматизированных потоков работ.

По данным направлениями организациями-законодателями теоретических основ систем

исполнения потоков работ выполняется большая работа по стандартизации универсальных решений и технологий. При этом для создания гибкой, расширяемой архитектуры интеграционного сервера необходима четкая декомпозиция процесса разработки и исполнения потока работ на отдельные этапы и анализ задач, возникающих на каждом из них. В данной статье описан подход к созданию интеграционного сервера, автоматизирующего весь жизненный цикл потока работ от создания до отладки и исполнения.

## 2 Формальное представление потока работ

В основе каждого автоматизируемого потока работ заложено понятие так называемой модели потока работ, которая представляет собой формализованное описание потоков работ, отражающее реально существующую или предполагаемую деятельность в рамках некоторого реального производственного процесса.

Модель потока работ должна давать ответы на вопросы:

- какие процедуры (функции, работы) необходимо выполнить для получения заданного конечного результата?
- в какой последовательности выполняются эти процедуры?
- какие механизмы контроля и управления существуют в рамках рассматриваемого потока работ?
- кто выполняет процедуры процесса?
- какие входящие документы/информацию использует каждая процедура процесса?
- какие исходящие документы/информацию генерирует процедура процесса?
- какие ресурсы необходимы для выполнения каждой процедуры процесса?
- какая документация/условия регламентирует выполнение процедуры?
- какие параметры характеризуют выполнение процедур и процесса в целом?

В настоящее время разрабатываются многочисленные стандарты, целью которых является интеграция существующих методов и языков моделирования потоков работ и создание единого методического и технологического базиса моделирования автоматизированных потоков работ.

Метод SADT (Structured Analysis and Design Technique) создан Дугласом Россом (SoftTech, Inc.) еще в 1969 г. и поддерживается Министерством обороны США, которое было инициатором разработки семейства стандартов IDEF (Integrated DEFinition Methods). Метод SADT реализован в одном из стандартов этого семейства - IDEF0, который был утвержден в качестве федерального стандарта США в 1993 г. Это процессный метод управления. То есть система рассматривается именно как набор потоков работ и не строится

организационно - штатная структура. Эта модель описывается как на естественных языках, так и при помощи диаграмм. В описании данной методологии указано, что все диаграммы модели SADT взаимосвязаны и организованы в иерархию. Вершина иерархии описывает систему в целом, это самое общее описание, а листья - самые детализированные описания. После описания системы в целом проводится разбиение ее на крупные фрагменты. Этот процесс называется функциональной декомпозицией, а диаграммы, которые описывают каждый фрагмент и взаимодействие фрагментов, называются диаграммами декомпозиции. После декомпозиции контекстной диаграммы проводится декомпозиция каждого большого фрагмента системы на более мелкие и так далее до достижения нужного уровня подробности описания.

В конце 1980-х годов был разработан метод моделирования IDEF3, являющийся частью семейства стандартов IDEF. Предполагалось использование метода для моделирования работы ВВС США. С помощью этого метода стало возможным моделировать последовательность действий в рамках некоторого процесса. Основой модели IDEF3 служит так называемый сценарий процесса, который выделяет последовательность действий и подпроцессов анализируемой системы. Главной единицей модели IDEF3 является диаграмма. Другой важный компонент модели - действие, или в терминах IDEF3 "единица работы" (Unit of Work). Каждое действие имеет идентификатор. Существуют связи между действиями (однаправленные стрелки) и три типа таких связей: временное предшествование, объектный поток, нечеткое отношение (вид взаимодействия каждый раз оговаривается отдельно).

Также для описания потоков работ используется моделирование потоков данных. Диаграммы потоков данных (Data Flow Diagrams - DFD) представляют собой иерархию процессов, которые связаны между собой потоками данных. Диаграммы показывают, как обрабатывает информацию каждый процесс, как процессы связаны друг с другом, а также как работает сама система, каким образом она обрабатывает поступающие данные.

К настоящему времени количество средств, методов и стандартов описания потоков работ сильно увеличилось. Качественно новым шагом в моделировании потоков работ стало появление нотации Process Modeling Notation (BPMN[3]), представленной консорциумом Business Process Management Initiative (BPMI) в 2003 году. Целью этого проекта является создание общей нотации для различных категорий специалистов: от аналитиков и экспертов до разработчиков ПО. BPMN модель визуализируются с помощью диаграммы под названием Business Process Diagram (BPD). Диаграмма BPD имеет два основных достоинства. Во-первых, она проста в использовании и

понимании. Применяя ее на начальном уровне сложности, можно найти общий язык с нетехническим персоналом. Во-вторых, поднимаясь на более высокий уровень сложности описания, можно постепенно подойти к естественному отображению на языке исполнения потоков работ. Определяя достаточный уровень абстракции нотация BPMN позволяет наглядным образом описывать модели потоков работ безотносительно среды их функционирования. На базе такой «скелетной» реализации процесса можно получать различные варианты «исполняемого кода». Помимо этого графическая нотация BPMN 1.0 определяет ряд дополнительных функциональных возможностей, делающих ее наиболее выгодным средством для перевода реальных потоков работ в формальную объектную модель:

- широкий набор элементов декларации потока работ сочетающий простоту визуализации и богатую выразительность;
- возможность прямого отображения в различные исполняемые языки потоков работ (например, BPEL4WS[4] и BPML[5])
- возможность экспорта в распространенных форматах описания потоков работ, таких как диаграммы активности UML 1.0 и 2.0 в виду схожей базовой метамодели.

## 2.1 Особенности стандарта BPMN

Члены рабочей группы BPMI консорциума создали спецификацию BPMN 1.0 на основе различных существующих стандартов, таких как UML Activity Diagram, IDEF, ebXML и других. В соответствии с [3] выделяются два основных назначения BPMN-диаграммы. Во-первых, она должна быть понятна различным пользователям от аналитиков и разработчиков потоков работ до работников компании, которые будут управлять потоком работ (это требование должно выполняться для всех нотаций, созданных для описания потоков работ). Во-вторых, с помощью BPMN нотации могут быть визуализированы XML-языки реализации потоков работ, такие как BPEL4WS (Business Process Execution Language for Web Services), BPML (Business Process Modeling Language).

В [3] описывается BPMN-метамодель в виде таблиц, содержащих информацию о BPMN-сущностях, атрибутах и их семантике, а также находится информация, необходимая для отображения диаграммы потоков работ в BPEL4WS формат.

В [3] обозначены соглашения по изображению диаграмм:

- Изображение графических элементов. Программное средство обязано изображать элементы BPMN в соответствии с формами, оговоренными в спецификации, запрещено менять форму элемента (например, нельзя заменить квадрат треугольником). Есть свобода

в выборе цвета, размера, положения текста. Множество элементов может быть расширено в реализации за счет добавления маркеров и индикаторов к элементам, использования различных цветов элементов, видов линий, если это не противоречит спецификации. Могут быть добавлены новые виды комментариев к диаграмме, если их вид не будет конфликтовать с уже существующими элементами.

- Семантика BPMN элементов. Программное средство должно изображать связи элементов в соответствии с их семантикой, определенной в спецификации. То есть, невозможна графическая связь между двумя элементами, если в спецификации не указана семантика их взаимодействия.

В соответствии со спецификацией можно выделить следующие основные категории элементов, которые могут быть изображены на диаграмме потоков работ (Business Process Diagram):

- Объект потока работ (Flow Object):
  - Событие (Event). События влияют на ход выполнения процесса и могут иметь причину (trigger) или некоторый результат (result).
  - Действие (Activity). Действие характеризует работу, которую выполняет компания. Действие может быть как атомарным (не разбивается на другие действия), так и составным.
  - Шлюз (Gateway). Шлюзы используются для маршрутизации различных потоков управления.
- Связь (Connecting Object):
  - Поток управления (Sequence Flow). Поток управления показывает очередность выполнения действий в ходе процесса.
  - Поток сообщений (Message Flow). Этот элемент указывает поток сообщений между двумя сущностями процесса, одна из которых получает, а другая – отправляет сообщение.
  - Ассоциация (Association). Ассоциация используется для привязки некоторой текстовой или графической информации к объектам потока работ.
- «Плавательная дорожка», обозначающая участника потока работ (Swimlane):
  - Контейнер (Pool). Контейнер символизирует участника процесса, а также служит границей, отделяющей множество действий процесса от других контейнеров.
  - Дорожка (Lane). Дорожка разделяет контейнер на части и наследует его длину и ширину, она необходима для организации и классификации действий.
- Артефакт или комментарий (Artifact):
  - Объект данных (Data Object). Этот элемент не влияет на поток управления или поток

сообщений процесса, но содержит информацию о данных необходимых для выполнения действия и/или информацию о том, что является результатом действия.

- Группа (Group). Группа необходима для объединения действий с целью документации и не влияет на поток управления.
- Аннотация (Text Annotation). Данный элемент необходим для указания дополнительной информации о диаграмме.

## 2.2 Проблема выбора обменного формата

На данный момент не существует стандарта для обмена BPMN-диаграммами. Хотя два возможных варианта развития событий указаны в [3]: это направление по созданию XML схемы – надстройки над существующей схемой BPEL, а также использование стандартного формата для обмена метаданными XMI (XML Metadata Interchange). Также в настоящее время консорциум OMG продолжает разрабатывать метамодель описания потоков работ BPDM (Business Process Definition Metamodel), охватывающую нотацию BPMN. Спецификация BPDM будет основана на стандартах MOF и XMI, и как следствие предоставит обменный формат для BPMN-диаграмм.

## 2.3 XPDL 2.0, характеристика XPDL 2.0 как обменного формата для BPMN моделей

Активное участие в описании и стандартизации потоков работ приняла организация WfMC (Work Flow Management Coalition). Разработанный коалицией WfMC в 1999 г. язык определения потоков работ WPD (Workflow Process Definition Language) был описан с помощью формул Бэкуса - Наура. В 2002 г. язык WPD был переписан, и его новая версия – XPDL (XML Process Definition Language) 1.0 уже была основана на XML. В марте 2004 г. WfMC призвала своих членов вносить предложения по расширению языка XPDL, и в октябре 2005 года появилось расширение - XPDL 2.0 [23].

В [23] указано, что XPDL 2.0 обратно совместим с XPDL 1.0. Все элементы BPMN, которые не были представлены в XPDL 1.0, есть в XPDL 2.0. Таким образом, все сущности графической нотации BPMN отображаются в соответствующие элементы и атрибуты XML-языка XPDL 2.0. XPDL 2.0 может использоваться как общий обменный формат, а также как некоторое формальное представление между разработкой и средой исполнения - описание процесса происходит в одной среде, а интерпретация в другой.

Для обозначения графической информации (координаты объекта на диаграмме) в XPDL 2.0 используется специальный элемент <NodeGraphicsInfo> с атрибутами, указывающими графическую информацию об объекте. Важным атрибутом здесь является ToolID (идентификатор,

приложения, которое изображает диаграммы) – каждый редактор может рисовать диаграмму по-своему, поэтому допускается несколько элементов <NodeGraphicsInfo> с различными данными и идентификаторами приложения. Метамоделю BPMN не предполагается конструкций для хранения графических данных.

XPDL является расширяемым языком, и существует возможность добавлять новые элементы (если это необходимо редактору) с помощью элемента <Extended Attributes>.

Кроме того, что существует возможность явно указать приложения, которые участвуют в действии, с помощью специальной структуры, можно реализовать действие как одно из BPMN-заданий (атомарных действий).

Особенность языка XPDL заключается в том, что элементы созданные для интерпретации выносятся отдельно от остальных элементов, это позволит отделить информацию, относящуюся к BPMN-диаграмме от информации, необходимой для интерпретации процесса. Внесение элементов интерпретации не влияет на саму конструкцию документа. Редактор может читать этот документ и корректно строить диаграмму. Необходимо лишь игнорировать «излишние» элементы.

Все основные элементы BPMN отображаются в сущности языка XPDL, причем также отображаются и все атрибуты основных элементов, за исключением нескольких атрибутов элемента диаграммы потоков работ, которых не было в предварительной версии спецификации BPMN 1.0. Информация об атрибутах диаграммы потоков работ, которые не нашли своих аналогов в XPDL, может быть сохранена с помощью расширяющих элементов XPDL 2.0.

6 декабря 2005 года фирма Fujitsu объявила о выходе редактора потоков работ Interstage Business Process Manager Studio [6], основанного на спецификации XPDL 2.0 и поддерживающего графическую нотацию BPMN.

#### 2.4 Стандарт для описания метамоделей MOF и стандарт обмена моделями и метамоделями XMI

Спецификация XMI (XML Metadata Interchange) [24], как и не разлучный с ней стандарт MOF (Meta Object Facility) [25], были созданы OMG в 2000 году. MOF сейчас существует в версии 2.0 и представляет из себя набор сущностей таких как классы, пакеты, ассоциации, типы данных и предназначен для описания метамоделей, также как UML (Unified Modeling Language) предназначен для описания моделей. Сам язык UML является метамоделю и может быть записан на MOF, а диаграмма сделанная на UML является моделью, на основе которой уже создаются объекты. Это может быть проиллюстрировано с помощью классической четырех уровневой архитектуры метаданных, изображенной на рисунке 1.

### M 3 MOF (Мета-метамоделю)

### M 2 UML (Метамоделю)

### M 1 UML Моделью

### M 0 Объекты

Рис. 1. Четырехуровневая архитектура метаданных

Спецификация XMI предназначена для обмена моделями и метамоделями в рамках MOF и сейчас существует в версии 2.1, содержит набор расширенных формул Бэкуса-Наура (EBNF) для реализации отображения метамоделей, записанной на MOF в язык описания структуры XML документа - XML Schema, в ранних версиях XMI спецификации в DTD (Document Type Definition). Так, например мы представляем UML на MOF, а потом генерируем с помощью формул UML DTD. Этот документ прилагается к некоторым версиям XMI спецификации. В спецификации XMI также указаны EBNF для получения XMI файла по модели. Файл должен валидироваться по XML Schema или DTD файлу, полученному по заданной метамоделю в соответствии с рисунком 2.

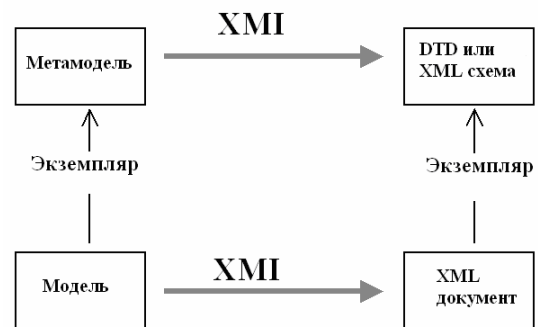


Рис. 2. XMI отображения

В статье [26] следующим образом описывается назначения стандартов MOF и XMI: «Стандарт MOF позволяет описывать метамоделю и метаданные, а стандарт XMI дает возможность представлять их единообразно в формате XML.

Возможность оперирования и обмена метаданными в едином формате позволяет программам взаимодействовать и обмениваться данными, даже если они не имеют информации друг о друге, т. е. о форматах данных, используемых другой программой. Обмен данными между разнородными источниками может управляться формальными описаниями метаданных о типах, преобразованиях и типизированных отображениях».

Обмен BPMN диаграммами на основе стандарта XMI давно обсуждается в обзорных публикациях о сериализации диаграмм потоков работ [27,34], тем

не менее, пока подобная обменная схема не была предложена. Этот факт побудил к созданию, описанию и публикации обменного формата, опирающегося на стандарты MOF и XMI. Для построения формата, необходимо изобразить BPMN метамодель на MOF и сделать отображение этой метамодели в XML-схему по правилам стандарта XMI. Последняя версия спецификации XMI – [24]. Сама BPMN метамодель не предполагает хранение графической информации о расположении объектов на диаграмме, поэтому она была расширена соответствующими конструкциями. Как и в формате XPD 2.0 была реализована возможность хранить графические данные различных редакторов.

Отметим схожесть языков MOF и UML: у них достаточно много общих конструкций (класс, пакет, ассоциация, композиция и др.), и, скорее всего в будущем OMG произведет слияние этих двух стандартов. В настоящее время UML предоставляет графическую нотацию для MOF. Для изображения MOF метамодели можно пользоваться UML редакторами.

### 3 Понятие среды функционирования процесса

Следующий этап после формализации потока работ в терминах объектной модели – выбор среды функционирования потока работ, включая представление интерфейсов его участников. Среда функционирования потока работ обеспечивает привязку абстрактной модели потока работ к реальным участникам (программным компонентам) и структурам оперативных данных процесса.

Наибольшее распространение по представлению среды взаимодействия автономных программных приложений (за которыми могут стоять и пользовательские рабочие места) получила сервис-ориентированная модель. Сервис-ориентированная модель обеспечивает поддержку распределенных, повторно используемых, взаимозаменяемых компонентов. Большинство созданных на текущий момент систем управления потоками работ в качестве среды функционирования автоматизированных участников потока работ (сервисов) придерживаются сервис-ориентированной модели взаимодействия, представленной архитектурой WEB-сервисов (WSA[7]). WEB-сервисы – это распределенные сервисы, обрабатывающие XML-сообщения, передаваемые по протоколу SOAP[9]. Интерфейс WEB-сервисов описывается с помощью языка WSDL[8]. Web-сервисы обеспечивают интероперабельность между программными компонентами, которые могут размещаться в различных инфраструктурах.

Архитектура WSA – это инициатива, направленная на реализацию в полном объеме преимуществ распределенных задач, выполняющихся в рамках автоматизированного потока работ, которые допускают объединение

различными способами новых и ранее созданных программных средств с целью поддержки развивающихся требований к результатам работы процесса. Акцент на повторное использование компонентов обуславливает потенциал для значительного повышения продуктивности разработки и вытекающей из этого факта способности разработчиков потоков работ быстро реагировать на новые требования.

Быстрое развитие архитектуры WSA привело к появлению ряда сопутствующих стандартов, позволяющих эффективно решать задачи координации потоков управления и данных в рамках автоматизированного потока работ:

- поддержка транзакционности (WS-Transaction[10]);
- безопасные протоколы передачи сообщений (WS-Security[11]);
- протоколы гарантированной доставки сообщений (WS-ReliableMessaging[12]);
- каталогизация описаний WEB-сервисов (UDDI[13]).

Таким образом, использование архитектуры WSA в качестве среды функционирования потока работ позволяет построить универсальную, допускающую повторное использование существующих компонентов, безопасную и надежную архитектуру интеграционного сервера.

### 4 Переход от абстрактного описания потока работ к исполняемому

Имея описание модели потока работ и подготовленный набор программных компонентов (WEB-сервисов), выполняющих автоматизированные участки процесса, можно переходить к следующему этапу – получение по описанию модели процесса «программного кода».

На текущий момент существует ряд XML-языков для описания потоков работ в виде композиций WEB-сервисов. Наиболее быстрыми темпами развивается язык BPEL4WS (Business Process Execution Language for Web-services).

К своему нынешнему виду BPEL эволюционировал из слияния XLANG[14] (языка категории XML BP, созданного в Microsoft на основе[17] пи-исчисления[18]) и разработанного в IBM языка WSFL[15] (также языка категории XML BP, но использующего[17] сети Петри[19]). Позже некоторые дополнения были предложены BEA, а потому и эту компанию включают в число авторов BPEL. В данный момент BPEL проходит финальную стадию доработки в техническом комитете OASIS BPEL TC[20].

Язык BPEL содержит набор базовых управляющих конструкций и конструкция для работы с данными для описания исполняемого потока работ, участниками которого являются WEB-сервисы. Для него не определена строгая графическая нотация. Поэтому задача разработки

описания потоков работ на языке BPEL может быть эффективно решена с использованием декомпозиции общего процесса разработки:

1. разработка графического представления потока работ и его модели с помощью нотации BPMN (выполняется аналитиками);
2. генерация исполняемого кода на языке BPEL по полученной модели потока работ (автоматизированный этап разработки, т.к. спецификация BPMN 1.0 описывает способ отображения BPD диаграмм в конструкции языка BPEL);
3. доработка и отладка полученного BPEL описания (выполняется техническим специалистом).

## **5 Интеграционный сервер как единое окно к сервисам системы исполнения потоков работ**

Задача исполнения формализованных описаний потоков работ – одна из основных задач интеграционного сервера. В целом интеграционный сервер предоставляет функциональную поддержку в следующих областях:

- предоставление средств разработки описаний потоков работ;
- интерпретация и исполнение формализованных описаний потоков работ;
- предоставление внешним пользователям доступа к подсистеме исполнения потоков работ для интерактивной работы.

В соответствии с описанным выше подходом последовательность действий по разработке нового потока работ имеет следующий вид:

1. построение абстрактной модели потока работ;
2. реализация программных компонент, автоматизирующих исполнение отдельных задач потока работ;
3. получение по абстрактной модели «исполняемого кода» с привязкой к реальным участникам процесса;
4. верификация и отладка полученного описания процесса.

Подсистема исполнения потоков работ обеспечивает интерпретацию описания потока работ, координацию потока управления и потоков данных. Она позволяет осуществлять взаимодействие процесса с программными компонентами и людьми, выполнять ведение статистики выполнения процесса, поддерживать безопасность и целостность оперативных данных процесса.

Для взаимодействия с работниками на местах интеграционный сервер должен обеспечивать возможность интерактивного взаимодействия с пользователями для решения следующих задач:

- управление жизненным циклом потока работ;
- предоставление внешних данных процессу;
- мониторинг состояний;
- реализация модели ролевого участия пользователей в потоке работ.

В рамках проекта «Разработка средств интеграции научных приложений» по программе фундаментальных исследований ОМН РАН выполняется работа по анализу, разработке и прототипированию сервера интеграции научных приложений [21,22], обеспечивающего автоматизацию жизненного цикла потоков работ по указанным выше направлениям. Для разработки описаний потоков работ сервер интеграции приложений предоставляет набор визуальных средств для разработки диаграмм в нотации BPMN 1.0 и XML-описаний исполняемых процессов на языке BPEL4WS 1.1. Развитые средства интерпретации описаний потоков работ на языке BPEL4WS позволяют выполнять как исполнение процесса в реальном времени, так и динамическую отладку. Средства мониторинга и интерактивного взаимодействия с пользователем обеспечивают хорошую управляемость потоков работ.

## **6 Разработка обменных форматов для описаний потоков работ интеграционного сервера**

В рамках представленных работ были реализованы механизмы сохранения модели BPMN 1.0 в формате XPD 2.0, а так же обратный экспорт элементов BPMN диаграмм в XPD 2.0 форме. Механизмы экспорта/импорта были использованы для поддержки обмена BPMN описаниями потоков работ в рамках интеграционного сервера. Преимуществом данного обменного стандарта является то, что он является распространенным и на данный момент существует множество редакторов, поддерживающих XPD 2.0 и обратно совместимый с ним XPD 1.0.

Помимо этого была разработана модель BPMN с помощью MOF и получена XML-схема для XMI-формата BPMN-описаний потоков работ. В качестве примера рассмотрим получение части XML-схемы для элемента BPMN-диаграмма потока работ (Business Process Diagram). Итак, метамодель изображается в UML редакторе (Rational Rose 2003). На рисунке 3 показана диаграмма для элемента – диаграмма потока работ (Business Process Diagram).

Схема была получена автоматически с помощью специальной программы (Eclipse Modeling Framework [28]) генерации XML-схемы в соответствии с XMI правилами по MOF метамодели, и преобразована вручную: были указаны кардинальности для элементов, так как они не были выставлены программой. В силу того, что

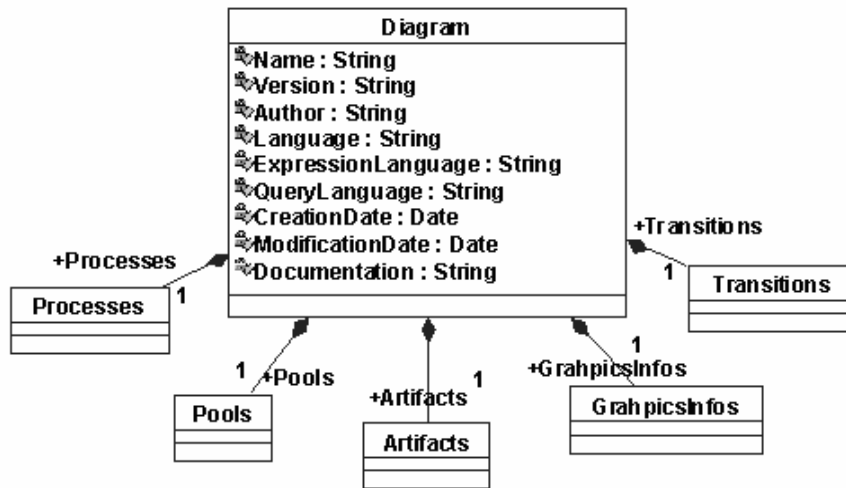


Рис. 3. Элемент диаграмма потока работ (Business Process Diagram)

ссылки класса на другие классы переходят и в элементы, находящиеся внутри элемента – класса, и в атрибуты, получается избыточность, соответствующие элементы были удалены и ссылки на другие классы осуществляются только по средствам атрибутов с использованием уникальных идентификаторов. Если необходимо указать один XML-элемент внутри другого – это обозначается с помощью композиции на MOF.

Отметим, что схема создана на основе метамодели BPMN 1.0, которая была расширена за счет внесения графической информации, а также классов – конвертов, предназначенных для группировки информации. Всего в полученной схеме 84 класса, 23 перечислимых типа.

Ценность представляет не только созданный обменный формат, но также само описание метамодели BPMN на MOF. Вместе с BPMN моделью программе, поддерживающей стандарт MOF, может передаваться также информация и о метамодели (XML - схема), что позволит программе автоматически адаптироваться к предложенным условиям и уже самостоятельно оперировать моделями BPMN. Более того, формальное описание BPMN на MOF позволяет создавать на основе стандарта MOF QVT (Query/View/Transformation) [29] преобразование в другие MOF метамодели. Возможна также генерация программного кода по MOF метамодели, так например компанией Sun был создан стандарт JMI (Java Metadata Interface) [30] для автоматического создания java кода на основе метамодели, созданной на MOF.

Диаграммы BPMN и UML Activity (диаграмма деятельности UML) схожи, поэтому в качестве дальнейших шагов по развитию проекта планируется создать отображение между BPMN и известным UML - XMI форматами с целью обеспечения совместимости описаний BPMN процессов со стандартными UML редакторами

(такими как Rational Rose, Poseidon for UML и т.д.). В настоящее время известны работы по отображению диаграмм деятельности UML в XPDL 2.0 [31].

Редактор сохраняет данные в собственном XML - формате, поэтому для реализации отображений был использован XML – язык преобразований XSLT (Extensible Stylesheet Language Transformations) [32], созданный W3C консорциумом. На этом языке можно написать шаблоны, которые будут применены к входному XML – файлу (исходному дереву) и на выходе будет построено конечное дерево, которое может являться как XML – структурой так и просто текстом. Для преобразования с помощью языка XSLT необходим XSLT-процессор, который это преобразование будет выполнять. В качестве такого процессора разработчиком редактора была выбрана Java-библиотека с открытыми исходными текстами, созданная в соответствии с рекомендациями W3C – Xalan-Java 2.7.0 [33]. Всего для редактора было создано четыре шаблона для реализации функций импорта и экспорта в выбранные форматы (импорт из формата XPDL 2.0, импорт из формата BPMN XMI, экспорт в формат XPDL 2.0 и экспорт в формат BPMN XMI).

Общая схема работы подсистемы импорта/экспорта BPMN редактора представлена на рисунке 4.

Подсистема экспорта/импорта осуществляет двустороннюю трансформацию внутреннего XML-формата представления диаграмм к нужным внешним форматам. Трансформация выполняется с помощью конвейеров (Pipeline). Конвейеры трансформации могут быть однопоточными (конвейеры импорта и экспорта) или двухпоточными (двусторонние конвейеры). Конвейеры первого типа применяются для реализации функций импорта и экспорта данных редактором. Конвейеры второго типа – для





Рис. 4. Схема импорта и экспорта данных BPMN редактора

реализации функцию «Новая диаграмма», «Открыть диаграмму», «Сохранить как». В случае импорта и экспорта в форматы BPMN XML и XPD 2.0 используются двусторонние конвейеры, так как построенные отображения в эти форматы полностью сохраняют информацию о BPMN модели.

## 7 Использование интеграционного сервера при решении задачи формирования ресурсов электронной библиотеки

В рамках проекта по программе информатизации РАН была сформулирована задача макетирования электронной библиотеки научного наследия. При этом одной из основных задач в процессе сопровождения библиотеки является процесс подготовки электронных документов.

В качестве основы для представления цифровых копий удобно использовать формат Adobe PDF:

- PDF позволяет сохранять файл после распознавания в режиме «текст под изображением», а значит полностью исключить процедуру ручного исправления ошибок распознавания;
- средствами PDF достаточно легко можно организовать полнотекстовый поиск. После распознавания текста отсканированных бумажных изданий каждая электронная копия содержит «невидимый» слой распознанного текста, по которому организовывается полнотекстовый поиск;
- возможности сжатия файлов в PDF достаточны для размещения на одном CD-R необходимого количества отсканированных страниц.

Весь технологический процесс получения электронных версий бумажных изданий может быть представлен в виде ряда элементарных операций, при этом работа может быть организована по принципу конвейера: каждый участник процесса

выполняет поставленную ему задачу. В целом технологический процесс можно представить в виде следующей последовательности действий:

1. Просмотр и подготовка бумажного издания.
2. Определение оборудования, на котором будет проходить сканирование и параметров сканирования.
3. Сканирование (или оцифровка) бумажных изданий, возможно, с параллельным приведением к другим физическим формам, например, микрофишированием.
4. Контроль сканирования, исправление ошибок (повторное сканирование «бракованных» или пропущенных страниц).
5. Автоматическая постраничная разрезка отсканированных разворотов.
6. Контроль автоматической разрезки и исправление ошибок.
7. Постраничная обработка (удаление дефектов сканирования и восстановление истинных размеров страницы) и размещение электронных версий в промежуточном хранилище.
8. Распознавание в Adobe Fine Reader целых или частей некоторых материалов.
9. Дополнительное сжатие PDF файлов.
10. Размещение электронной версии в постоянном хранилище создание резервных копий.
11. Сопровождение полученных электронных версий дополнительными метаданными (автор, аннотация, оглавления электронной публикации и т.д.).
12. Классификация электронной версии документа (привязка к рубрикам и разделам электронной библиотеки).
13. Определение параметров безопасности и шифрование PDF файла.
14. Размещение в оперативном хранилище для online публикации.
15. Подготовка к online публикации (индексирование, кэширование).
16. Размещение полученной электронной версии в открытом доступе.

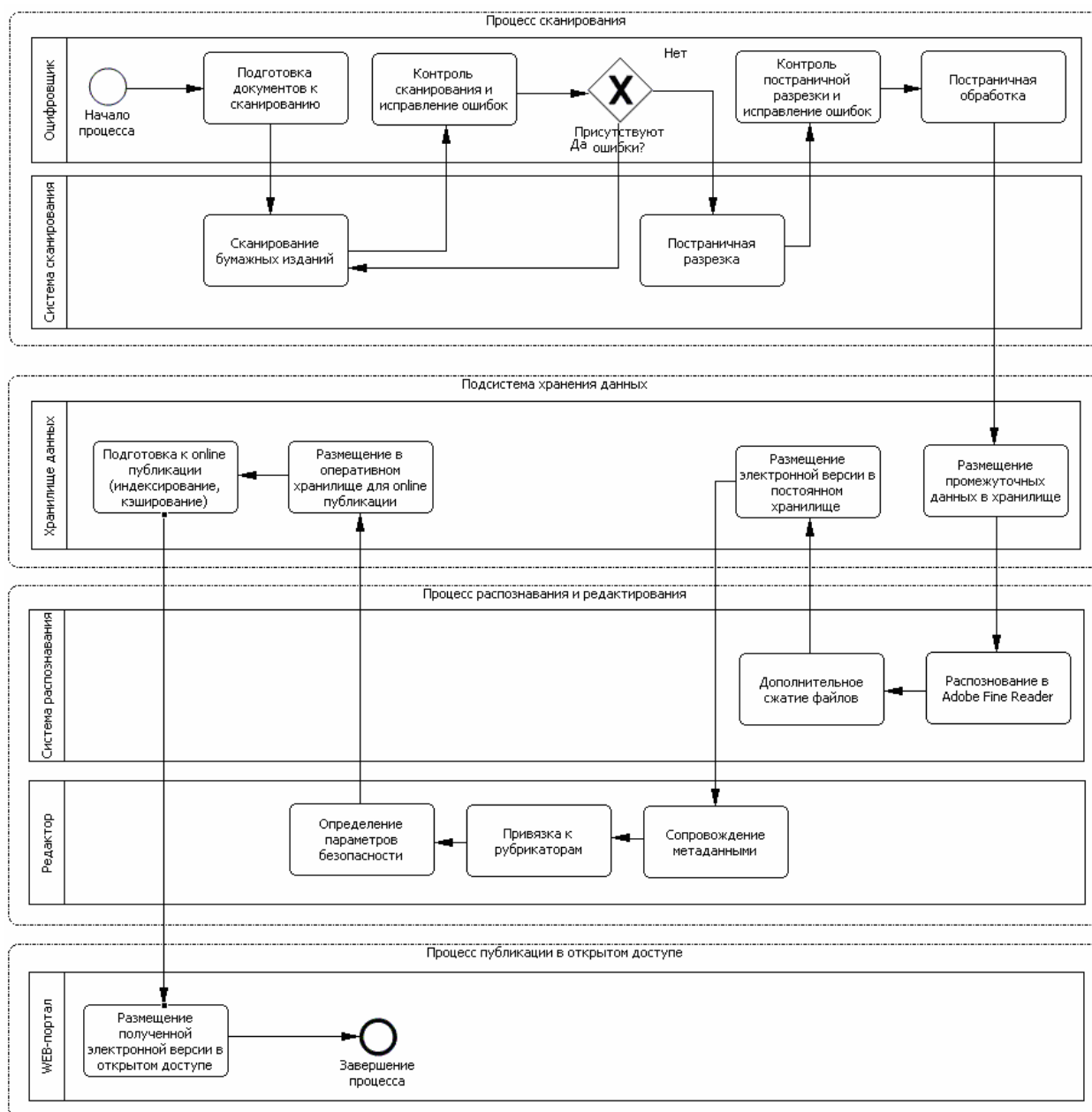


Рис. 5. BPD диаграмма процесса создания электронных версий бумажных изданий

На рисунке 5 представлено крупноблочное описание данного процесса в нотации BPMN.

В данном процессе условно выделено три типа подпроцессов:

- процесс сканирования бумажных носителей;
- операции по работе с хранилищем данных;
- процесс распознавания и редактирования электронных изданий;
- процесс публикации электронных изданий в открытом доступе.

В данном процессе выделены следующие роли участников:

- «Оцифровщик» – технический специалист, ответственный за подготовку бумажных изданий к сканированию, управление

процессом сканирования, исправление ошибок в полученных электронных версиях.

- «Система сканирования» – автоматизированная информационная система, управляющая сканирующим оборудованием и выполняющая дополнительные прикладные задачи (например, разбивка на страницы). Имеет внешний интерфейс (WEB-сервис).
- «Подсистема хранения данных» – автоматизированная информационная система, представленная хранилищем данных и внешним интерфейсом (WEB-сервисом) для доступа к операциям извлечения и модификации информации.
- «Система распознавания» – автоматизированная информационная система, выполняющая распознавание текста

полученных на предыдущих этапах электронных документов и выполняющая дополнительные прикладные задачи (сжатие, шифрование файлов и т.д.). Имеет внешний интерфейс (WEB-сервис).

- «Редактор» – технический специалист, ответственный за управление процедурой распознавания текста, сопровождение электронных документов дополнительной метаинформацией и размещение их в открытом доступе.
- «WEB-портал» – WEB-приложение, предоставляющее внешним пользователям доступ к ресурсам электронной библиотеки, а так же ряд дополнительных сервисов.

Указанные процедуры допускают автоматизацию в виде потока работ как с полностью, так и с частично автоматизированными этапами. Использование для целей автоматизации таких процессов сервера интеграции приложений позволит обеспечить высокую производительность процесса подготовки электронных версий бумажных изданий в целом.

## 8 Заключение

На данный момент разработана BPMN-модель потока работ по подготовке электронных изданий, а так же получено исполняемое BPEL-описание процесса. Работы по данному проекту планируется завершить к концу 2006 года. В качестве дальнейших работ можно выделить:

- разработка WEB-сервисов, автоматизирующих исполнение отдельных этапов процесса;
- создание рабочих пользовательских мест (WEB-интерфейсов) участников процесса;
- отладка созданного процесса в подсистеме динамической отладки интеграционного сервера.

В направлении работ по интеграционному серверу планируется решение следующего ряда задач:

- интеграция с UDDI реестром для каталогизации сервисов и описаний процессов;
- поддержка возможности конвертирования BPEL-описаний процессов к исполняемым форматам, поддерживаемым другими системами (Oracle BPEL Process Manager, Microsoft BizTalk Server, IBM WebSphere, InterSystems Ensemble);
- поддержка протоколов безопасной и гарантированной передачи данных в рамках процесса (WS-Security[11] и WS-ReliableMessaging [12]);
- расширение функционала подсистемы сбора статистики и аудита хода исполнения процесса.

Подсистема импорта/экспорта интеграционного сервера позволяет обмениваться данными с

различными редакторами потоков работ, поддерживающими формат XPDL. Как было указано ранее, планируется также реализовать функции импорта и экспорта в формате UML XMI. Наличие различных форматов выгрузки и загрузки данных предоставляет пользователю возможность частично или полностью редактировать потоки работ в привычных для него редакторах, а также использовать уже созданные шаблоны потоков работ. Разработанный на основе стандарта XMI формат обмена BPMN-описаниями рабочих процессов обеспечивает обмен на основе метамодели BPMN с программами, поддерживающими стандарт MOF.

## Литература

- [1] Carol Prior, Workflow and Process Management // [http://www.wfmc.org/information/Workflow\\_and\\_Process\\_Management.pdf](http://www.wfmc.org/information/Workflow_and_Process_Management.pdf), 2005
- [2] OMG UML 2.0 // <http://www.omg.org/cgi-bin/doc?formal/01-09-67>, 2001
- [3] OMG Business Process Modeling Notation Specification // <http://bpmn.omg.org/Documents/BPMN%20V1-0%20May%203%202004.pdf>, 2004
- [4] Business Process Execution Language for Web Services Version 1.1. // <http://www-106.ibm.com/developerworks/library/ws-bpel/>
- [5] BPML working draft March 25, 2002. // <http://xml.coverpages.org/bpml.html>
- [6] Fujitsu Software Corporation. Fujitsu Introduces Interstage Business Process Manager Studio, Resets the Standard for Seamless Process Collaboration between Business and IT // <http://www.fujitsu.com/global/services/software/interstage/wnew/pr120605.html>
- [7] Web Services Architecture // <http://www.w3.org/2002/ws/>
- [8] Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language // <http://www.w3.org/TR/2004/WD-wsdl20-20040326/>
- [9] SOAP Version 1.2 Part 1: Messaging Framework // <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>
- [10] Web Services Transaction (WS-Transaction). // <http://www-106.ibm.com/developerworks/webservices/library/ws-transpec/>
- [11] WS-Security Specification // [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=ws](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws)
- [12] WS-ReliableMessaging Specification // [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsr](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsr)
- [13] Universal Description, Discovery and Integration specification. // <http://www.uddi.org>
- [14] Web Services for Business Process Design. // <http://xml.coverpages.org/xlang.html>

- [15] Web Services Flow Language. // <http://xml.coverpages.org/wsfl.html>
- [16] А.К. Нестеренко, А.А. Бездушный, Т.М. Сысоев, А.Н. Бездушный. Возможности службы управления потоками работ по манипулированию ресурсами репозитория ИСИР // Сборник научных трудов X научно-практического семинара "Новые технологии в информационном обеспечении науки". Москва: 2003, с.206-231.
- [17] Леонид Черняк, BPM: близкие перспективы и далекие горизонты // Электронный журнал «Открытые системы», выпуск №11, 2006
- [18] Davide Sangiorgi, David Walker, The pi-calculus: a Theory of Mobile Processes // Cambridge University Press, 2001, [http://www.cs.unibo.it/~sangio/Book\\_pi.html](http://www.cs.unibo.it/~sangio/Book_pi.html)
- [19] W. Reisig, Petri Nets, An Introduction // EATCS, Monographs on Theoretical Computer Science, Berlin, 1985
- [20] OASIS Web Services Business Process Execution Language (WSBPEL) TC // [http://www.oasis-open.org/committees/tc\\_cat.php?cat=ws](http://www.oasis-open.org/committees/tc_cat.php?cat=ws)
- [21] Мартынов М.Ю., Нестеренко А.К., Бездушный А.Н., Ляшков А.С., Ярошук И.О., Технология научных потоков работ как средство автоматизации исследований гидрофизических процессов // Тихоокеанский океанологический институт ДВО РАН, 2006, <http://info2006.febras.net/program.doc>
- [22] Нестеренко А.К., Бездушный А.А., Сысоев Т.М., Бездушный А.Н., Ярошук И.О., Моделирование распределенных научных вычислительных процессов посредством применения технологии рабочих процессов // Седьмая Всероссийская конференция "Электронные библиотеки: перспективные методы и технологии, электронные коллекции", Ярославль, 2005.
- [23] Workflow Management Coalition. Workflow Standard. Process Definition Interface-- XML Process Definition Language. Version 2.00. // [http://www.wfmc.org/standards/docs/TC-1025\\_xpdl\\_2\\_2005-10-03.pdf](http://www.wfmc.org/standards/docs/TC-1025_xpdl_2_2005-10-03.pdf)
- [24] Object Management Group (OMG). XMI Mapping Specification, v2.1. // <http://www.omg.org/docs/formal/05-09-01.pdf>
- [25] Object Management Group (OMG). MOF Core Specification, v2.0. // [http://www.omg.org/technology/documents/formal/MOF\\_Core.htm](http://www.omg.org/technology/documents/formal/MOF_Core.htm)
- [26] Кузнецов. М. MDA — новая концепция интеграции приложений // Открытые системы. 2003. № 09.
- [27] Jan Mendling, Gustaf Neumann, Markus Nuttgens, Vienna University of Economics and BA. A Comparison of XML Interchange Formats for BusinessProcess Modelling. // <http://wi.wu-wien.ac.at/home/mendling/publications/04-EMISA.pdf>
- [28] Eclipse Modeling Framework (EMF) // <http://www.eclipse.org/emf/>
- [29] Object Management Group (OMG). MOF QVT Final Adopted Specification. // <http://www.omg.org/docs/ptc/05-11-01.pdf>
- [30] Java Metadata Interface (JMI) // <http://java.sun.com/products/jmi/>
- [31] Barbara Gallina, Nicolas Guelfi, Amel Mammam. Software Engineering Competence Center Faculty of Sciences, Technology and Communication University of Luxembourg. Structuring Business Nested Processes Using UML 2.0 Activity Diagrams and Translating into XPDL. // [http://se2c.uni.lu/tiki/se2c-bib\\_download.php?id=1962](http://se2c.uni.lu/tiki/se2c-bib_download.php?id=1962)
- [32] W3C Candidate Recommendation XSL Transformations (XSLT) Version 2.0. // <http://www.w3.org/TR/xslt20/>
- [33] Apache XML. Xalan-Java 2.7.0. // <http://xml.apache.org/xalan-j/>
- [34] Michael Guttman. "Real MDA" Stands Up // Software Magazine. 2006. №05. (<http://www.softwaremag.com/L.cfm?Doc=950-5/2006>)

### **Applying application integration approaches to information resources management process automatization**

Nesterenko A.K. Danilina A.A. Sysoev T.M.  
Bezдушny A.N. Serebryakov V.A.

In this article some approaches to the information resources management process automatization are considered. There is an analysis of the workflow management systems architecture. Some problems of formal workflow definition and its interaction with external information systems and users are considered. The possibility of using workflow for electronic library resources forming process automatization is examined.