СЦЕНАРНАЯ МОДЕЛЬ И ЯЗЫК ОПИСАНИЯ ЗНАНИЙ ДЛЯ ОЦЕНКИ И ПРОГНОЗИРОВАНИЯ СИТУАЦИЙ

ТРОЦКИЙ Д.В., ГОРОДЕЦКИЙ В.И.

УДК 381.3.

Троцкий Д.В., Городецкий В.И. **Сценарная модель знаний и язык описания процессов для оценки и прогнозирования ситуаций.**

Аннотация. Работа посвящена проблеме оценки ситуаций и прогнозирования ее развития в приложениях, в которых требуется иметь средства для гибкого изменения сценариев поведения в зависимости от достигнутых состояний системы и текущего состояния внешней среды в реальном времени. Дается обзор и анализируются достоинства и недостатки существующих языков описания процессов и показывается, что традиционные языки спецификации систем, способные представлять, главным образом, реактивное поведение, не обладают необходимыми выразительными возможностями и потом не в состоянии справиться с поставленной задачей. Особенности рассматриваемой задачи спецификации, оценки и прогнозирования ситуаций демонстрируются на задачи управления фрагментом системы заправки стартового ракетного комплекса. В работе предлагается новый язык, который предназначен для описания знаний о сценариях, позволяющий оценивать текущее состояние исполнения сценария, прогноз его развития и выбора варианта продолжения в зависимости от достигнутых состояний и состояния внешней среды. Дается описание основных элементов языка, их графической нотации и описывается его операционная семантика. Возможности разработанного языка демонстрируются на примере описания модели диагностики нештатных ситуаций в процессе функционирования фрагмента системы заправки. Для этого приложения представлены примеры спецификации процесса в терминах разработанного языка сценариев.

Ключевые слова: сценарные знания, оценка ситуации, прогнозирование ситуации, нештатные ситуации.

Trotsky D.V., Gorodetsky V.I. **Scenario–based knowledge model and language for situation assessment and prediction.**

Abstract. This paper is devoted to the problem of situation assessment and prediction in application where it required to have a flexible means to change the scenario behavior, depending on the progress of the system and the current state of the environment in real time. Paper reviews and analyzes the advantages and disadvantages of existing workflow languages and it is shown that the traditional language of the specification that can represent primarily reactive behavior, lack the expressive possibilities and then not be able to cope with a task. Features of the problem specification, evaluation and prediction of the situations shown in the task management system, a fragment of rocket fuel starting complex. The paper proposes a new language, which is intended to describe the knowledge of the scenarios to assess the current state of performance scenario, to predict of its development, and to select the option to continue depending on the progress of states and the state of the environment. A description of the basic elements, graphical notation and operational semantics of language are presented. Developed language capabilities are demonstrated by the example of describing a model diagnostic of unexpected situations in the process of a fragment of fuel. This application presents examples of process specifications developed in terms of language scenarios.

Keywords: knowledge model, scenario, process, unexpected situation.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

1. Введение. Существо практически любой современной системы управления в организационно–технической сфере во многом определяется множеством реализуемых ею процессов и их взаимодействием. Как правило, системы такого типа являются крупномасштабными, включают в себя множество разнородных подсистем и обычно включают в контур управления человека. В соответствии с современными взглядами большинство таких систем попадает под понятие систем управления бизнес–процессами. К ним, например, относятся системы управления логистикой предприятия и технологическими процессами, реализуемыми на предприятии, в частности, это относится к распределенным и виртуальным предприятиям; системы управления в чрезвычайных и аварийных ситуациях; управление логистикой и технологическими процессами в аэропорту и многие другие. Хорошим примером системы управления, где главным аспектом является управление распределенными взаимодействующими процессами, является технологический процесс подготовки и пуска ракет и космических аппаратов. В нем необходимо организовать согласованную работу множества различных компонент стартового технологического оборудования и номеров расчетов, увязать множество взаимосвязанных процессов, нормальная или нештатная реализация которых приводит к тем или иным ситуациям, в которых необходимо быстро принимать обоснованные решения, поскольку задержки или ошибки могут привести к катастрофическим последствиям. Практически то же самое имеет место и в других системах управления бизнес–процессами. Различие, главным образом, касаются масштабов системы, скорости протекания процессов, сложности и разнообразия отношений и ограничений на множестве процессов, возможностей по возникновению нештатных ситуаций, а также масштабов и опасности последствий ошибок в управлении.

При разработке таких систем, а также непосредственно при управлении ими в режиме использования ключевую роль играет модель функционирования системы. Известно, что среди передовых подходов к процессу проектирования и разработки важная роль отводится методологии, которая опирается на использование моделей. То же самое относится и к управлению процессом функционирования, когда текущее состояние процесса работы системы, анализ возникающих ситуаций и управление ими рассматриваются в контексте модели. Естественно, что качество процесса проектирования и разработки системы, как и качество процесса управления им в реальном времени, существенно зависят от качества используемых

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

моделей. К сожалению, выразительные возможности современных средств спецификации систем рассматриваемого типа достаточно ограниченны. Прежде всего, это относится к языкам описания процессов и их взаимодействия.

Основной недостаток существующих средств моделирования состоит в том, что в них, как процессы проектирования, так и процессы функционирования описываются в достаточно «жесткой» форме, когда все возможные варианты и ветви развития процесса, а также варианты управления должны быть перечислены заранее. При этом сложно, если возможно вообще, использование экспертных знаний и механизмов рассуждений. Проблема обусловлена тем, что в настоящее время делаются только первые шаги для разработки структур для представления и использования знаний о процессах, которые, по существу, должны обеспечить возможность гибкого реагирования на текущую ситуацию, причем как при ее оценивании, так и при прогнозировании с целью выбора наилучшего продолжения сценария.

В соответствии с современными представлениями такие возможности средств моделирования должны быть заложены в язык описания сложных взаимосвязанных сценариев поведения системы. Он должен содержать в себе возможность представления сценарных знаний и их использования в режиме реального времени. Язык должен быть элементом среды разработки и среды управления, иметь графическую нотацию и средства компиляции в программный код языка высокого уровня типа JAVA, C++ и тому подобные.

Такой язык предлагается в данной статье. Целью данной работы является описание построенного языка, его сравнение с известными языками спецификации процессов, а также его демонстрация на примере из области управления средствами заправки носителя на стартовом комплексе, в котором (в примере) делается акцент на обнаружение и диагностику нештатных ситуаций, а также на их обработку. Данная статья является практически первой публикацией авторов на рассматриваемую тему. Разработанный язык пока не имеет средств реализации. Поэтому он демонстрируется на простом и достаточно понятном примере.

Последующий материал данной статьи организован следующим образом. В разделе 2 представлен обзор основных работ по языкам описания процессов. В разделе 3 описан пример процессов во фрагменте системы заправки ракет на стартовом комплексе. В разделе 4 представлено детальное описание разработанного языка сценарных

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

знаний, приведен пример его использования для описания процесса функционирования фрагмента заправочного стартового комплекса. В заключении резюмируются основные результаты и предстоящие исследования и разработки.

2. Обзор известных языков спецификации процессов. На сегодняшний день существует около десятка наиболее популярных языков описания процессов. В данном разделе приводится обзор некоторых из них. Главная задача этого обзора – проанализировать их возможности и оценить их достоинства и недостатки, что необходимо для формулировании требований к разрабатываемому языку.

Исторически модель сети Петри была одной из первых формальных моделей, предназначенных для спецификации моделей процессов[1]. Эта модель хорошо описана в научной литературе [5], поэтому здесь о ней даются только краткие сведения, касающиеся той его части, которая используется в других моделях. Оценивая его в целом, следует отметить, что эта модель весьма абстрактна, а потому ее выразительные возможности достаточно ограничены. В частности, она не рассматривает вопросы, касающиеся среды исполнения, а потому не имеет выразительных средств для спецификации ряда важных функциональных аспектов.

Для элементов сетей Петри предложена понятная и удобная графическая нотация, которая во многом определила привлекательность этой модели для разработчиков. Модель сети Петри представляется графически в виде *двудольного* графа (рис.1),

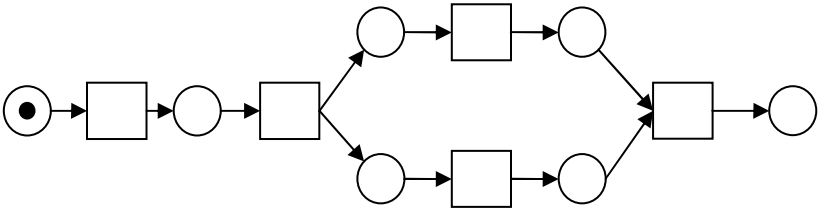


Рис. 1. Пример диаграммы сети Петри.

[1] Конечно, первой моделью такого назначения была модель STRIPS, основанная на исчислении предикатов первого порядка. Однако она оказалась бесполезной для моделирования практических задач. Впрочем, это имеет место для большинства разработок, базирующихся на языке исчисления предикатов первого порядка и его неразрешимых фрагментах.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

состоящего из вершин двух типов–позиций и переходов, соединенных дугами. В ней вершины одного типа не могут быть соединены друг с другом. В графической нотации сети Петри прямоугольники представляют переходы, а круги–позиции. Каждый переход имеет как входные, так и выходные позиции. С точки зрения семантической интерпретации в терминах моделируемой проблемы, позиции ставятся в соответствие данным, частичным состояниям или целям, в то время как переходы ставятся в соответствие операциям (процессам, процедурам), которые выполняют преобразование данных, реализуют реакции на входные события и т.п. Полагается, что переходы выполняются мгновенно. В сетях Петри для представления динамики процесса используются маркеры (метки), которые могут быть помещены в позиции. «Движение» маркеров от позиции к позиции отражает динамику работы сети, а их исходное размещение определяет конкретный экземпляр процесса. Текущее распределение меток в позициях задает текущее состояние исполнения процесса, моделируемого сетью Петри.

Ввиду слабых выразительных возможностей этой модели и ее расширений, например, цветных, ингибиторных, иерархических сетей Петри и других, на практике эта модель используется, главным образом, в качестве основы для других языков. Например, сеть потоков работ (*Workflow net)*, которая рассматривается далее, имеет в своей основе модель сети Петри. Популярность модели сети Петри, которая была очень высокой до 1990-х годов, в настоящее сильно упала. Главная причина этого состоит не только в осознании слабости ее выразительных возможностей, но и в слабости средств для представления операционной семантики. В частности, появление модели диаграмм переходов состояний, решающих проблему представления операционной семантики гораздо эффективнее, безусловно, повлияло на падение популярности сетей Петри.

Важный шаг в усилении выразительных возможностей языков моделирования процессов, в частности, для представления в рамках одной модели различных аспектов процесса, был сделан в языке, известном под названием «*Event-Driven Process Chain*» («*Цепь процессов, управляемых событиями*», далее *EDPC*). Этот язык был предложен в работе [6]. В нем процесс описывается множеством взаимосвязанных мета моделей, каждая из которых отражает определенный «взгляд» на процесс, поддерживая таким образом современную концепцию многомодельности. В частности, этот язык

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

позволяет пописывать следующие аспекты общей модели (подмодели) процесса:

1) *Функциональная модель*. Процесс в целом и составляющие его подпроцессы представляются в виде функций, которые в приложении соответствуют «работам», «действиям», причем как сложным, так и атомарным (не подлежащим декомпозиции).
2) *Модель данных*. В ней описываются объекты данных, которые необходимы для исполнения действий (подпроцессов). Объекты данных моделируются с помощью диаграмм отношений.
3) *Модель выхода*. Эта модель описывает результаты процессов, например, те продукты или сервисы, которые были получены или исполнены в результате реализации процесса.
4) *Организационная модель*. Она описывает организационную структуру, в которой реализуется (бизнес–) процесс, т.е. участвующие в нем сущности и их отношения, должности и возложенные на них функции, роли и т.п.

Язык EDPC получил определенное распространение, а некоторые его базовые черты использованы при создании стандартных графических нотаций языка для описания бизнес–процессов, разработанных законодателем мод в этой области консорциумом OMG [4].

Некоторые идеи EDPC используются в языке, предложенном в данной работе. К ним относятся, во-первых, представление процесса, ориентированное на цели, когда каждое сложное или атомарной действие приводит к достижению определенной цели (достижению процессом определенного состояния), во-вторых, активное использование онтологии для представления знаний мета уровня, и, в-третьих, представление связей предусловий и постусловий.

Язык *Workflow Net* (*WN*, «*Сеть потоков работ*») был предложен как расширение модели сети Петри с одной стороны, а с другой – он имел целью преодолеть некоторые недостатки языка *EDPC* [2]. Действительно, язык *EDPC* выглядит очень удобным для пользователей вследствие ясности концептуальной модели, но его недостатки, прежде всего, в отсутствии какой-либо формальной или технологической поддержки. Таким образом, язык *EDPC* может быть использован для формальной спецификации модели системы управления потоками работ при ее проектировании, но не при программной реализации. С другой стороны, сети Петри с их слабыми выразительными возможностями не обладают рядом средств описания, которые свойственны *EDPC*. Поэтому язык *WN* может

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

оцениваться как некий симбиоз двух названных языков. В частности, язык *WN* использует графическую нотацию сети Петри с дополнительными ограничениями, а именно [7]:

1) В ней присутствует позиция, которая не имеет входных дуг (начальная позиция);
2) В ней присутствует позиция, не имеющая выходных дуг (конечная позиция);
3) В ней каждая позиция и каждый переход находятся на пути, который образуют начальная и конечная позиции.

Пример процесса, описанного с помощью *WN*, представлен на рис.2.

Последовательность управляющих конструкций может быть легко представлена средствами *WN*. Рис. 2 демонстрирует логические операции «*и*» («*And split»), «или» («And join»), «исключающее или»(«Xor split»)* и *«(«простое соединение»)* в терминах языка *WN*. В операции «*неявное эксклюзивное или*» («*implicit exclusive or split»)* оба перехода происходят в один и тот же момент. Однако в такой модели только один переход может быть помечен, получив метку от входного узла. Правило перехода должно определить, какую из ветвей процесса выполнять (рис.3). Данная операция называется «*эксклюзивное или*» («*exclusive or split*»). Поскольку здесь решение откладывается до последнего момента времени, этот паттерн управляющий называется «*отложенный выбор*».

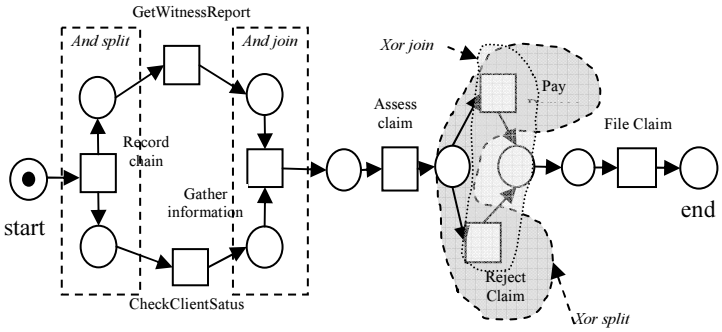


Рис. 2. Пример *WN* процесса.

Резюмируя особенности языка *WN*, отметим, что:

1) Он позволяет выражать формальную семантику потоков работ, т.е. явно определяет правила перехода между узлами.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

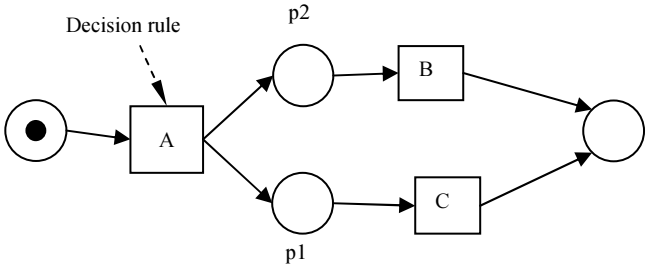


Рис. 3. Правила выбора на основе разделения.

2) Графическое представление модели процесса в языке *WN* достаточно выразительно. В частности, оно позволяет легче понять логику процесса и связи между вложенными процессами.
3) Он позволяет анализировать некоторые формальные свойства модели процесса, используя фактически все возможности, предоставляемые для этого языком сети Петри.

Однако этот язык обладает рядом очевидных недостатков. В нем нет средств для явного представления данных, которые используются в модели процесса. В описании переходов не предусмотрено описание временных аспектов. Это, конечно, очень серьезный недостаток *WN*, поскольку, например, нарушение верхнего порога продолжительности работы является одним из часто используемых признаков при диагностике нештатных ситуаций. Недостатком языка является и используемое предположение о том, переходы выполняются мгновенно, а на практике за время выполнения действия система может изменить свое состояние. По этой же причине по окончании выполнения некоторого действия система не генерирует событие, которое могло бы проинформировать другие процессы о достижении системой нового состояния, что на практике чрезвычайно важно, например, для синхронизации процессов реального времени. Для реализации некоторых переходов в модели требуется явное написание выражений, дополняющих структурную модель сети Петри. Например, это требуется для однозначности выбора переходов, требующих логики управления, соответствующей операциям «*или*» и «*исключительное или*».

Язык «*Yet another Workflow Language*» (*YAWL*) для моделирования (бизнес–) процессов обогащает ранее описанные языки множеством дополнительных нотаций, позволяющих описывать новые паттерны модели ([1], [13], [5]). Кроме того, он позволяет явно описывать

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

операционную семантику процессов использованием модели диаграммы переходов состояний. Он также предоставляет возможности уже на этапе проектирования описывать ситуации, в которых некоторое действие должно выполняться несколько раз. В языке имеются средства для явной синхронизации паттернов процессов в случае, когда необходимо реализовать логику управления процессами типа «*разделительное или*» и «*объединительное или*». Он также предоставляет возможность использования «внешней» синхронизации паттернов процессов, когда, например, в результате выполнения процесса возникает ситуация, требующая отмены не только текущего действия, но и некоторых других действий в общей структуре процесса. Что очень важно, язык *YAWL* предоставляет средства для описания поведения, вызываемого появлением исключительных ситуаций. Подчеркнем, что все перечисленные возможности отсутствуют в других языках, которые базируются на использование структурной компоненты сети Петри.

Очень важной особенностью языка *YAWL* является возможность описания различных аспектов управления процессами. В нем предоставляется возможность описания в модели сложных зависимостей между потоками управления в различных задачах [11]. Для этих целей в нем определено около 40 специализированных паттернов, представляющих варианты паттернов потока управления, например, параллелизм, различные ситуации выбора продолжения процесса, несколько видов синхронизации и др. Другой полезный аспект–это описание потоков данных. Язык *YAWL* предоставляет средства для определения типов данных, процессов передачи информации между паттернами модели процесса [9]. Специальные средства имеются для описания ресурсов и их распределения между

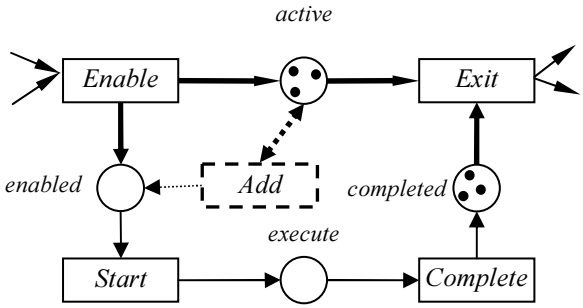


Рис. 4. Диаграмма перехода состояний для экземпляра задачи.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

задачами [12]. Наконец, как уже отмечалось ранее, язык *YAWL* предоставляет средства для описания причин возникновения исключительных ситуаций и конкретных механизмов их обработки. Пример описания процесса на языке *YAWL* представлен на рис.4.

Однако существенным недостатком данного языка является тот факт, что он способен описывать только реактивное поведение. Этого явно недостаточно для спецификации поведения, в частности, сценарного поведения современных информационных систем, которые в значительной мере ориентируются на реализацию «интеллектуального» поведения. Такое поведение предполагает возможность выбора сценария поведения, а также варианта продолжения сценария, дальнейшее исполнение которого невозможно по тем или иным причинам. Такими причинами могут быть непредвиденное состояние внешней среды, нештатное завершение некоторого действия или невозможность его завершения и др.

Язык «*Graph-based Workflow Language*» («*Язык потоков работ на основе графов*») имеет несколько интересных идей для представления операционной семантики, в частности, зависимостей между данными, а также оценке «*мертвых*» путей.

Представленный ниже пример иллюстрирует обработку запроса пользователем банковского кредита в банке [7]. В модели (рис. 5) действия представлены прямоугольниками, потоки управления– сплошными стрелками, а потоки данных – пунктирными стрелками. Процесс начинается со сбора кредитной информации. На следующем шаге оценивается риск выдачи кредита. Затем выполняется либо

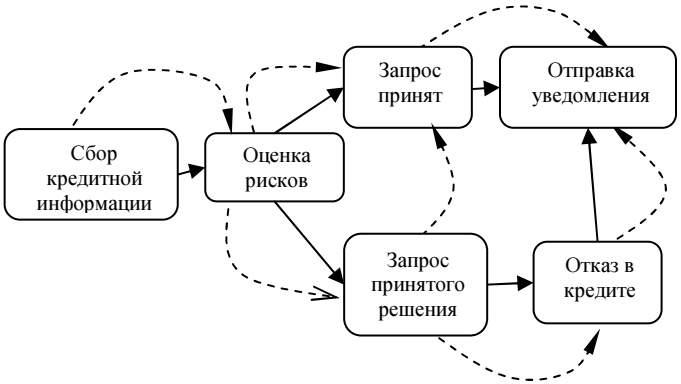


Рис. 5. Пример процесса запроса кредита на языке потоков работ на основе графов.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

действие «*запрос п*ринят», либо «*запрос на принятие решения*». Это решение может быть как положительным, так и отрицательным, но с обязательным уведомлением клиента. Каждая компонента процесса ассоциирована с данными определенного типа. Она имеет входные параметры, но они могут и отсутствовать. Каждый выход компоненты процесса служит входом для следующей компоненты в соответствии с потоком данных. Сами потоки данных на схеме, как правило, не показываются.

Каждая вершина потока управления имеет связанное с ним условие. Оно рассчитывается после того, как предыдущее действие завершилось. Вершина «сигнализирует», когда отвечающее ей действие завершено, инициируя выполнение следующего действия.

Операционная семантика представляется сигналами от вершин, и эти сигналы могут принимать значение ««*истина*» или «*ложь»* с запуском последующих действий, которые выбираются в соответствии со значением сигнала.

Когда все входные вершины некоторой вершины выработали соответствующие сигналы, очередная вершина оценивает значения сигналов (условий, которые определяют возможность начала действия, соответствующего вершине) и запускает выполнение следующего действия, если это разрешено условиями (если все «входные вершины» генерируют сигнал «*истина*»). Необходимые данные или ссылки на них передаются с потоком управления. Язык позволяет использовать иерархическое описание процессов. Наряду с дополнительными выразительными возможностями, важное достоинство этого языка состоит еще и в том, что он имеет программную реализацию.

В заключение данного краткого обзора рассмотрим графические примитивы, используемые для формального описания бизнес–процессов, которые разработаны международной организацией *Object Management Group* (OMG), одной их ведущих организаций в области стандартизации процессов проектирования и разработки информационных систем и их программной реализации. Для этого множества примитивов на английском языке используется название *Business Process Modeling Notations* (*BPMN*).

Стандартные примитивы разделяются на три основных категории, каждая из которых включает некоторое подмножество примитивов. К таким категориям относятся объекты потоков, артефакты, соединительные объекты, организационные пулы (swimlanes). Соответствующие обозначения показаны на рис. 6.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

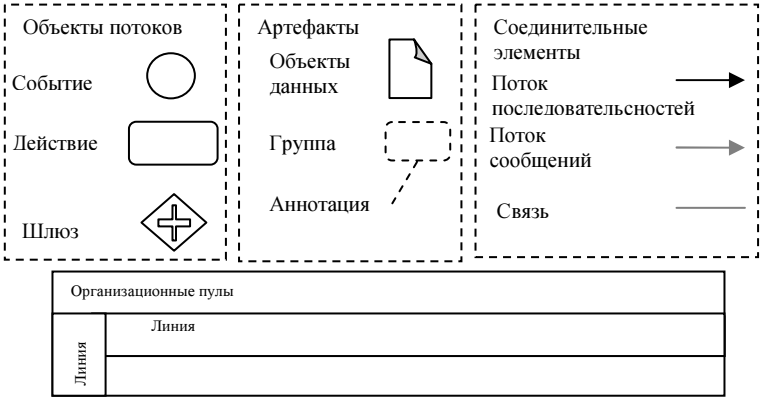


Рис. 6. Категории элементов.

1) Объекты потоков (*«flow objects»*) – это строительные блоки (бизнес–) процесса. Они представляют события (информирующие о достижении некоторых состояний), атомарное или сложное действие (работа, которая должна быть выполнена), шлюзы, (используются для представления паттернов потока управления).
2) Артефакты *(«artefacts»*) представляют объекты данных, группы и аннотации. Объекты данных задаются своими именами, и используются в качестве атрибутов. Группы – это формальное описание нескольких артефактов, объединенных вместе. Текстовые аннотации используются для комментирования и объяснения того или иного аспекта процесса или его составляющих.
3) Соединительные элементы (*«connecting objects*») связывают остальные категории элементов. Потоки последовательностей указывают порядок следования объектов. Поток сообщений представляет обмен информацией между узлами.
4) Организационные пулы (*«swimlane»*) представляют организационный аспект структуры процесса, указывая кем или каким подразделением выполняется определенная часть процесса.

С деталями стандартного набора графических примитивов, их графическим представлением, а также их формальной семантикой можно познакомиться на веб сайте OMG [4].

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Отметим основные достоинства множества графических примитивов, предложенных OMG:

1) Введенное множество примитивов объединяет в себе достоинства нотаций, предложенных в других языках описания процессов.
2) Оно предоставляет широкие возможности для формального представления компонент сложных процессов.
3) Множество примитивов позволяет хорошо представлять модели распределенных (бизнес–) процессов.
4) Введенные примитивы наглядны, а потому обеспечивают хорошее понимание модели, представленной в их терминах.

Однако *BPMN* нотации не предоставляют средств описания потоков данных. В ней имеются ограничения на обмен сообщениями, в частности, сообщения могут пересылаться только между соседними каналами организационных пулов. Определенные проблемы могут возникнуть при описании UML диаграмм, представляющих различные уровни спецификации. Наконец, *BPMM* предоставляет средства только для описания реактивных систем, что ранее отмечалось в качестве недостатка всех ранее рассмотренных языков. Это ограничивает возможности их использования для описания, диагностики и управления исключительными и нештатными ситуациями.

3. Пример приложения, где необходимо использование сценарного управления[1]**.** Прежде чем рассматривать существо предлагаемого языка и модели прогнозирования поведения сложных систем и управления ими на основе знаний о сценариях, рассмотрим содержательный пример, поясняющих суть решаемой проблемы. Приведенный пример используется далее для пояснения и демонстрации содержания развиваемого подхода. Данный пример описывает небольшой фрагмент ракетного стартового заправочного комплекса. Выбор этого примера для демонстрации обусловлен тем фактом, что нештатные ситуации, возникающие при работе заправочного комплекса, могут быть источником серьезных задержек общего процесса подготовки и пуска космических объектов, а также аварий. Поэтому проблема оценки ситуации, складывающейся в процессе заправки, ее прогнозирования и принятия решений с целью

[1] Исходная постановка данной задачи, информация по особенностям системы заправки в целом, по реализуемым в ней процессам, потенциальным нештатным ситуациям, а также по структуре описываемого далее фрагмента заправочного комплекса была предоставлена авторам Б.К.Гранкиным и используется здесь с его разрешения.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

своевременной и эффективной обработки нештатных ситуаций является очень важной и ответственной.

Процесс заправки ракеты на стартовом комплексе является одним из наиболее сложных, ответственных и потенциально опасных в общем процессе подготовки и пуска космических объектов. Задержки, которые возникают в процессе заправки по тем или иным причинам, в частности, вследствие возникновения нештатных ситуаций, являются главными причинами отклонений от графика подготовки ракеты к пуску и даже отмены пуска, когда возникает необходимость слива топлива.

В описываемом процессе подготовки задействованы:

1) Технический персонал комплекса;
2) Компоненты ракетного топлива двух типов, горючее и окислитель;
3) Система передачи топлива (система труб, резервуаров, насосов, датчики уровня топлива и др.);
4) Емкости наземных хранилищ топлива и баки ракеты;
5) Пульт управления процессом заправки.

Процесс заправки ракеты представляет собой сложный многошаговый процесс, в котором взаимодействуют люди, технические объекты и системы. Ввиду структурной сложности системы заправки, а также сложности технологии заправки даже при работе в штатном режиме, необходим мониторинг и компьютерная поддержка процессов управления системой передачи топлива. Это необходимо для контроля последовательности действий персонала и координации этих действий, для контроля последовательности включения и/или выключения тех или иных управляющих элементов системы, контроля промежуточных состояний и др. Кроме того, персонал нуждается в оперативной помощи для обнаружении и диагностики отказов, для обнаружения нештатных ситуаций и, в особенности, в рекомендациях по обработке нештатных ситуаций для возвращении системы в штатный режим функционирования. Например, в случае выхода из строя некоторых элементов системы, когда невозможно продолжать заправку по стандартной схеме, могут существовать альтернативные схемы заправки, использующие другие компоненты системы. Поэтому персонал нуждается в соответствующих подсказках и рекомендациях.

Однако в существующих системах такая возможность отсутствует. Чтобы это стало возможным, параллельно с реально функционирующей системой заправки необходимо иметь ее

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

адекватную математическую модель, представленную на некотором формальном языке, обладающем необходимыми выразительными возможностями. Необходимо также иметь также интеллектуальные средства, способные использовать эту модель для решения перечисленных выше задач мониторинга, диагностики и выработки рекомендаций. По существу, математическая модель должна быть средством представления знаний о системе заправки и возможных (допустимых) сценариях ее использования, а интеллектуальные средства должны играть роль механизмов использования этих знаний. Опишем содержательно модель заправочной станции, которая далее будет использована при построении ее формальной модели функционирования и использования.

Рассмотрим упрощенную модель фрагмента системы передачи топлива от заправочной станции в баки ракеты. В общем случае, система передачи топлива состоит из четырех основных элементов (рис.8):

1) *Емкость*. Она представляет собой хранилище топлива или бак заправляемой ракеты. В каждом хранилище имеется датчик уровня топлива. Емкость может быть соединена с N трубопроводами.
2) *Кран*. Он соединяет два трубопровода и может находиться в двух состояниях - «*кран открыт*» и «*кран закрыт*». В состоянии «*кран открыт*» жидкость (газ) может проходить через кран, и не проходить через него, если он закрыт. На каждом трубопроводе, идущем от емкости, обязательно установлен хотя бы один кран;
3) *Насос*. Это гидравлическая машина, которая служит для перемещения и создания напора (избыточного давления) жидкостей или газов. Насос соединятся только с двумя трубопроводами. Насос может находиться в трех состояниях: «*насос включен вперед*», «*насос включен назад*» и «*насос выключен*». В состояниях «*насос включен вперед*» или «*насос включен назад*» он перекачивает топливо в заданном направлении. В состоянии «*насос выключен*» топливо через него не проходит, т.е. насос в этом случае играет роль закрытого крана.
4) *Трубопровод*. Он состоит из соединительных труб между элементами системы для передачи топлива.

Процесс заправки – это процесс наполнения одной емкости содержимым другой емкости. Модель процесса заправки состоит

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

минимум из двух емкостей, двух кранов и одного насоса. В общем случае, процесс заправки может включать в себя N емкостей, K кранов ($K \geq N$) и M насосов. Поясним на примере понятия штатной и нештатной ситуаций. Рассмотрим схему фрагмента заправочной станции, представленную на рис. 8.

Пусть задача состоит в том, чтобы наполнить емкость $E3$ содержимым (жидкостью или газом) емкости $E1$. Для этого необходимо выполнить последовательность следующих действий:

1) Открывается кран $K1$;
2) Открывается кран $K5$;
3) Включается насос H.

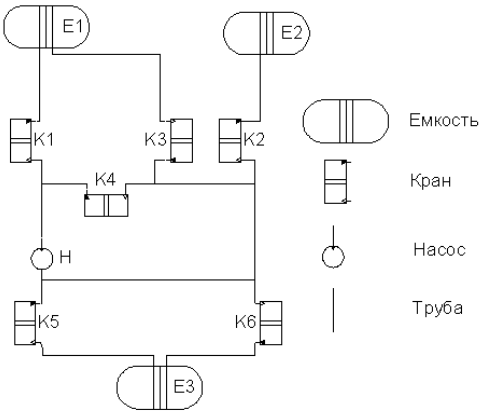


Рис. 8. Схема фрагмента заправочной станции.

По прошествии заданного интервала времени T (с некоторым допуском) после начала заправки уровень топлива в емкости $E3$ должен достигнуть требуемого уровня Y. Если это условие выполняется, то полагается, что процесс заправки прошел «*штатно*». В противном случае ситуация называется «*нештатной*».

Одной из самых ответственных задач мониторинга работы системы заправки является обнаружение самого факта возникновения нештатной ситуации, а также ее диагностика, т.е. определение типа нештатной ситуации, а также причин ее возникновения. В приведенном примере (рис.8) причины нештатной ситуации могут быть такими:

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

1) Неправильно работает датчик топлива, установленный в емкости $E1$ (была предпринята попытка заполнить бак топливом из емкости, где топлива недостаточно);
2) Неправильно работает датчик топлива, установленный в емкости $E3$ (емкость заполнена, но датчик этого факта не фиксирует);
3) Неисправен либо кран $K1$, либо кран $K2$, либо оба вместе, а потому при перекачке топливо через них не поступает;
4) Неисправен насос H (он не включается, а потому эквивалентен закрытому крану).

При возникновении нештатной ситуации автоматическая система мониторинга процесса заправки ответственна за обнаружение и диагностику, а также должна предложить варианты перевода системы в штатный режим функционирования. Эти варианты должны быть представлены в качестве рекомендации оператору станции. Именно оператор ответственен за окончательный выбор. Например, при разрыве одного из трубопроводов система должна предложить альтернативные пути перекачки топлива в заданную емкость в обход поврежденного участка и предложить их оператору станции, который инициирует начало выполнения решения, выбранного им.

Данный пример хорошо представляет особенности задач, которые рассматриваются в данной работе. Как видно из описания примера, основная особенность таких задач состоит в том, что сценарий выполнения процесса не является заранее определенным, поэтому его необходимо формировать в процессе исполнения в зависимости от достигнутых результатов, а в общем случае – и от состояния внешней среды. При этом в соответствии с логикой работы системы отдельные действия могут быть частично упорядоченными. Возможность выполнения одних действий зависит от успешности выполнения других действий, а общий сценарий выполнения процесса допускает вариативность. Например, может существовать несколько вариантов исполнения процессов, имеющих одну и ту же цель или приводящих к одному и тому же результату, но при этом требующих выполнения различных предусловий. Особенность рассматриваемых задач состоит также и в том, что в процессе исполнения сценария допускается возникновение нештатных ситуаций, а потому модель процесса должна предоставлять средства обнаружения факта нештатной ситуации, а также средства ее диагностики, что поможет выбрать сценарий возвращения системы в штатный режим.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Моделирование систем такого класса, а также гибкое управление в них возможны только в том случае, если система обладает «интеллектом». Иначе говоря, она должна не просто реагировать предопределенным образом на события в самой системе и на события внешнего мира, но обладает способностью обнаруживать непредвиденные ситуации, строить новые сценарии достижения цели в процессе функционирования. Следовательно, система должна обладать адекватными знаниями, которые делают возможным оценивание ситуации и построение вариантов достижения цели.

Это может быть достигнуто использованием адекватных языков для представления сценарных знаний, которые содержат в себе механизмы работы со знаниями. К сожалению, существующие языки формальной спецификации процессов, в частности, те, что были рассмотрены в разделе 2, не обладают требуемыми выразительными возможностями. Они не в состоянии обеспечить представление требуемых свойств приложений того класса, который был на примере очерчен в данном разделе. К таким «нереализуемым» свойствам относятся:

- Возможность описания знаний о вариантах достижения целей и механизмов вывода альтернативных сценариев достижения одной и той же цели, а также альтернативных вариантов продолжения уже начатых сценариев, приводящих к той же конечной цели;
- Учет временных ограничений на допустимую длительность действий;
- Учет предусловий, при выполнении которых действие может быть начато и оказаться потенциально успешным;
- Синхронизацию процессов во времени;
- Поддержку обмена информацией между разными сущностями, вовлеченными в распределенный процесс исполнения сценария;
- Обнаружение и диагностику нештатных ситуаций, а также вывод на основе знаний возможных вариантов обработки нештатных ситуаций с целью возвращения процесса в штатный режим функционирования.

Рассматриваемый далее язык описания моделей систем, ориентированный на сценарное представление знаний о реализуемых ими процессах и предоставляющий механизмы оценивания и прогнозирования ситуаций, а также варианты выбора альтернативных путей достижения цели системы, позволяет частично преодолеть

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

перечисленные выше недостатки существующих языков формального описания частично упорядоченных процессов и потоков работ.

4. Язык сценарных знаний.

4.1. Элементы языка спецификации сценарных знаний. Анализируя недостатки существующих языков спецификации взаимосвязанных процессов (см. раздел 2), нетрудно заметить, что ни один из них не обладает выразительными возможностями, необходимыми для спецификации проблем преставления и управления процессами в приложениях того типа, которое описано в разделе 3. Перечислим еще раз основные выявленные недостатки этих языков, которые ограничивают их использование в рассматриваемом классе задач:

1) Отсутствие явного представления потоков данных (*WN*);
2) Слабые возможности по описанию временных аспектов, в частности, временной синхронизации процессов (*WN*);
3) Отсутствие возможности выбора альтернативного сценария поведения, если такая необходимость возникла по некоторым причинам в реальном времени; неспособность описывать «интеллектуальное» поведение (*YAWL*, *OMG*);
4) Отсутствие возможности спецификации циклического поведения (*Graph-based Workflow Language*);
5) Ограничения на возможности обмена сообщениями (*OMG*);
6) Игнорирование того факта, что любое действие (переход в новое состояние) имеет продолжительность во времени (большинство языков).

Хотя перечисленные недостатки касаются разных языком, но важен тот факт, что среди них нет языка, свободного от всех перечисленных недостатков.

Отметим ключевые особенности разработанного языка, который далее кратко называется *SKBL* (*Scenario Knowledge Base Language*):

- Использование понятий атомарных и вложенных сценариев;
- Возможность явного описания предусловий, задающих допустимость исполнения действий (временные условия, условия по доступности данных и др.);
- Учет продолжительности исполнения действий;
- Возможность диагностики нештатных ситуаций и их обработки.

Рассмотрим далее основные компоненты языка.

4.1.1. Состояние и работа. Ключевыми понятиями разработанного языка спецификации процессов являются понятия «*состояние*» и «*действие*» («*работа*»). В графической нотации языка

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

эти понятия представляются узлами–примитивами типа овала и прямоугольника соответственно. Связи между узлами обозначаются стрелками. Как и в сети Петри, стрелки соединяют узлы только разных типов.

Формально «*состояние*» описывается в виде набора данных об объектах. Оно может соответствовать состоянию простого или сложного объекта, состоянию физической и/или виртуальной сущности.

Табл. 1. **Типы узлов.**

Графическая нотация	Название
Имя	Действие
Имя	Состояние

Примерами состояний для приложения, представленного на рис. 8, являются состояния «*Кран открыт*», «*Емкость заполнена*» и т.п. Состояния могут также интерпретироваться как конечные или промежуточные цели, которые должны быть достигнуты[1].

Вложенный сценарий может быть также представлен в терминах простых действий и/или вложенных сценариев и т.д. Например, действие «*Перекачать топливо из емкости E1 в емкость E3*» является сложным, т.к. оно включает в себя более простые действия типа *включение–выключение* некоторых кранов и продолжительный процесс перекачки топлива. В этом примере простыми действиями являются операции «*Открыть кран*», «*Включить насос вперед*» и др. В табл. 1 представлены графические нотации для узлов типа «*состояние*» и «*действие*», используемые в языке *SKBL*.

Табл. 2. **Активная сущность**

Графическая нотация	Название
Имя	Активная сущность

4.1.2.Активная сущность. Активная сущность представляет собой интерфейс обращения к другому процессу или некоторому объекту, который явно не участвует в процессе. Например, в роли активной сущности может выступать оператор станции заправки, который передает процессу параметры наполнения емкости или

[1] Действительно, целью работы системы является достижение некоторого состояния, которому могут предшествовать промежуточные состояния, которые могут рассматриваться как промежуточные цели системы.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

просто выполняет некоторое действие. Активная сущность обозначается шестиугольником (табл. 2).

Табл. 3. **Типы событий**

Графическая нотация	Тип	Описание
 Имя	Событие «*Активная сущность*»	Событие, инициируемое активной сущностью
 Имя	Событие «*Таймер*»	Указывает на время наступления события
 Имя	Событие «*Нештатная ситуация*»	Указывает на возникновение нештатной ситуации
 Имя	Событие «*Инициация работы*»	Передает работе сигнал на начало выполнения
 Имя	Событие «*Работа завершен*а»	Оповещает об успешном завершении элемента работы
 Имя	Событие «*Состояние достигнуто*»	Событие о достижении состояния
 Имя	Событие «*Ресурс освобожден*»	Событие извещает об освобождении некого ресурса
 Имя	Событие «*Узел-процесс завершен*»	Событие извещает о завершении группы элементов «Узел–процесс»
 Имя	Событие «*Узел-состояние завершен*»	Событие извещает о завершении группы элементов «*Узел–состояние*»

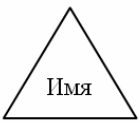
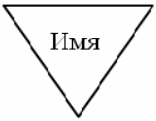
4.1.3. События и элементы условия. Взаимосвязь между действиями и состояниями реализуется с помощью событий, передаваемых ими друг другу, и условных элементов. События – это сообщения о наступлении «чего-либо». Они являются элементами

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

управления. События имеют атрибуты, с помощью которых они передают данные (например, время начала выполнения действия). Типы используемых событий представлены в табл. 3.

Каждое действие и состояние имеет также элемент условия, которое используется для проверки условия начала выполнения действия или наступления состояния. Например, действие «*Проверить уровень топлива в емкости Е3*» должно начаться, например, не раньше, чем через 30 секунд, после того как наступило состояние «*Кран К2 открыт*». Типы условных элементов представлены в табл. 4. Условие представляет собой логическое выражение на множестве событий. В случае выполнения условия, оно инициирует событие, запускающее выполнение процесса или достижение состояния.

Табл. 4. **Условные элементы**

Графическая нотация	Тип	Описание
	Постусловие	Содержит выражение логического типа. Выполняет проверку данных после выполнения работы или достижения состояния. Генерирует события: «*Нештатная ситуация*», «*Узел-процесс завершен*», «*Узел-состояние завершен*».
	Предусловие	Содержит выражение логического типа. Выполняет проверку данных перед выполнением работы. Генерирует событие о начале выполнения действия «*Инициация действия*».

Пример описания фрагмента сценария приведен на рис. 9.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

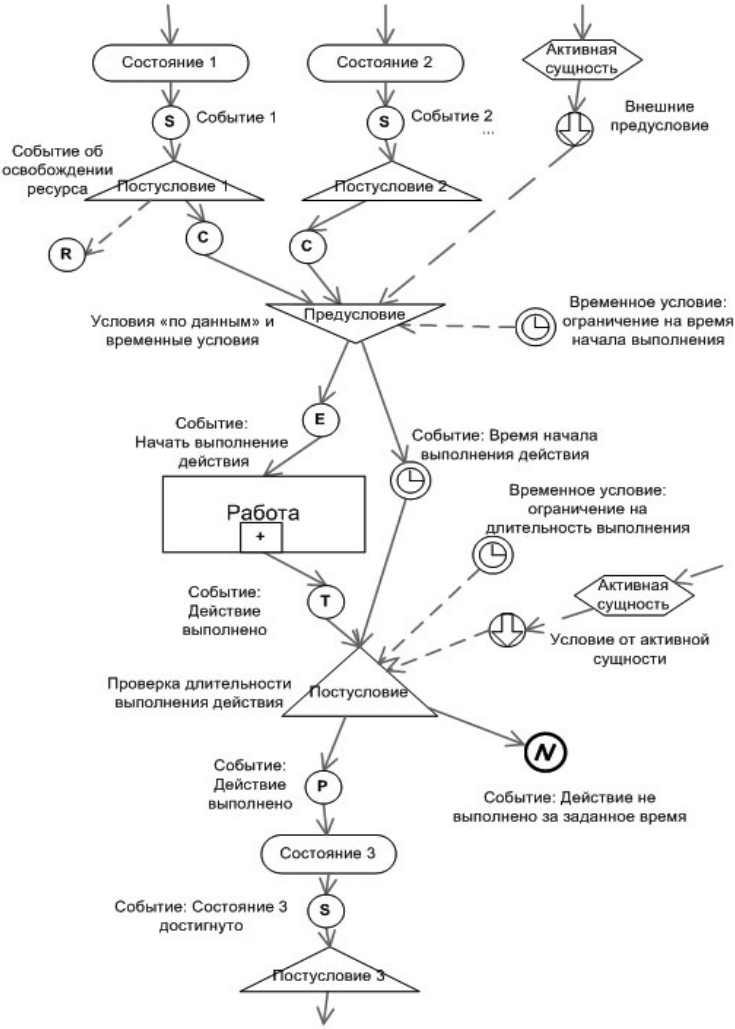


Рис. 9. Фрагмент сценария.

Опишем семантику выполнения сценария. При наступлении состояний 1, 2 генерируются соответствующие события, которые поступают на связанные с состояниями постусловия (1 и 2). Постусловия, в соответствии с заданными логическими выражениями,

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

генерируют события, которые передаются на вход предусловий действий. Предусловие оценивает временные ограничения (возможность начать действие немедленно), и в случае его выполнения генерирует событие на блок «Работа» о начале выполнения соответствующего действия. Одновременно с этим постусловие действия получает на вход событие, сообщающее о времени начала работы и временных ограничениям на его выполнение. Постусловие проверяет наложенные на действия ограничения, и если временные рамки выполнения были нарушены (превышен таймаут), то генерируется событие о нештатной ситуации, в противном случае – событие об успешном выполнении действия, которое передается на следующее состояние (состояние 3).

Для упрощения графического представления группы элементов, описывающих работу узла (включая все необходимые события и условия), а также для структуризации его программной реализации используется их агрегирование в абстрактный элемент *узел-процесс*. Аналогично группируются элементы, связанные с состоянием, в *узел-состояние*. Описание групп элементов представлено в табл. 5.

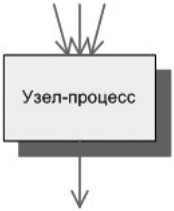
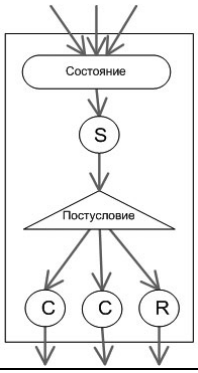
Узел-процесс объединяет предусловия, действие или вложенный сценарий, постусловие. Предусловие на основании заданных правил генерирует событие типа «*Инициация работы*» и событие «*Таймер*» о времени начала выполнения работы, которое передается в постусловие узла-процесса. По окончании выполнения элемент «*Работа*» генерирует событие «*Работа завершена*» и передает его в постусловие. Постусловие генерирует событие «*Узел-процесс завершен*», если время получения события «*Работа завершена*», меньше, чем время, указанное в событии «*Таймер*».

Узел-состояние объединяет состояние и постусловие. При достижении «*Состояния*» генерируется событие «*Состояние достигнуто*». Это событие передается в «*Постусловие*», которое на основании внутренних правил генерирует события типа «*Узел-состояние завершен*», «*Ресурс освобожден*».

4.1.4. Правила соединения элементов. Элементы «*работа*», «*состояние*», «*активная сущность*», а также элементы условий не связываются друг с другом непосредственно. Связь между этими элементами осуществляется через соответствующие события. Отметим, что элемент «*узел-процесс*» может иметь на входе несколько элементов типа «*узел-состояние*», но в качестве выхода может иметь только один узел «узел-состояние». В свою очередь, *состояние* может иметь много как входных, так и выходных *действий*.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Табл. 5. **Обозначения группы элементов**

Группа элементов	Обозначения	Название
		Узел-процесс
		Узел-состояние

4.2. Формальная основа языка спецификации сценария. Формальная модель языка предназначена для описания синтаксической структуры множества сценариев, которые допустимы в приложении. Под синтаксической структурой сценария понимается его агрегированное описание, в котором принимают участие только «узлы–действия» и «узлы–состояния». Синтаксическая структура

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

сценария задает частичный порядок на множестве пар <действие–состояние> и <состояние–действие>, определяя тем самым все множество возможных сценариев, допустимых в приложении. Поэтому формально структура описания сценария задает отношение частичного порядка на множестве узлов двух типов, в терминах которых сценарий описывается.

Синтаксическая структура сценария задается множеством троек, которые ставятся в соответствие действиям:

$$AM_r = < \{X_i\}_{i_r=1}^{n_r},\ A_r,\ X_{j_r}, > , \qquad (1)$$

где A_r – действие, $\{X_i\}_{i_r=1}^{n_r}$ множество «входных» состояний AM_r для действия, т.е. состояний, которые должны быть достигнуты для того, чтобы выполнение действия стало возможным[1], X_{j_r} – «выходное» состояние AM_r, т.е. то состояние, которое будет достигнуто при успешном выполнении действия. Все множество возможных действий обозначим символом AM, так что $A_r \in AM$. Таким образом описывается «*атомарное*» действие.

«*Сложное*» действие приложения описывается, в свою очередь, («*вложенным*») сценарием, который представляет собой тройку:

$$SM_r = < \{X_i\}_{i_r=1}^{n_r},\ S_r,\ X_{j_r}, > , \qquad (2)$$

где SM_r - сценарий, $\{X_i\}_{i_r=1}^{n_r}$ множество «входных» состояний сценария SM_r, X_{j_r} выходное состояние сценария. Обозначим символом $SM_r \in \{SM_k\} = SM$ множество всех вложенных сценариев (действий) моделируемого процесса.

На множестве введенных понятий определим $\boldsymbol{WM} = \boldsymbol{AM} \cup \boldsymbol{SM}$, т.е. множество всех моделей действий и сценариев описываемого процесса ($\boldsymbol{WM}$ означает *Work Model*, т.е. «*Модель работ*»).

Обозначим символом $\boldsymbol{X}$ множество всех состояний (промежуточных и конечных целей), $\boldsymbol{X} = \bigcup_{r=1}^{S} \bigcup_{i_r}^{n_r} \{X_i\}$.

На введенном ранее множестве действий $\boldsymbol{WM}$ и множестве состояний $\boldsymbol{X}$ введем две структуры (два отношения частичного

[1] Конечно, достижение множества "входных" состояний является только необходимым условие его выполнения, но недостаточным, поскольку для этого еще необходимо выполнение ряда предусловий.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

порядка). Первое отношение $-\boldsymbol{U}_{AX}$ (отношение непосредственного следования) определяется на декартовом произведении $\boldsymbol{WM} \times \boldsymbol{X}$. В нем для каждого успешного действия $A_r \in \boldsymbol{WM}$ «*следующим по порядку*» является один и только один элемент из множества $\boldsymbol{X}$, который является результатом действия (его «выходом», «достигнутой целью»).

Второе отношение, $\boldsymbol{V}_{XA}$, определяется на декартовом произведении множеств $\boldsymbol{X} \times \boldsymbol{WM}$. В нем определяется отношение непосредственного предшествования, в котором присутствуют пары $< X_i, A_j >$ в том и только том случае, когда состояние X_i является входом для действия A_j.

В таблицах 6, 7 дан абстрактный пример представления матриц смежности для каждого из этих отношений. Обратим внимание, что в матрицах используются не единичные элементы для обозначения пары отношения, а имена элементов, а именно, имена действий в отношении $\boldsymbol{U}_{AX}$ и имена состояний в отношении $\boldsymbol{V}_{XA}$. Это необходимо в последующем для построения более простых алгоритмов вывода необходимых «фактов» на основе данной структуры и содержания запроса.

Табл. 6. **Пример матрицы смежности для отношения $\boldsymbol{U}_{AX}$**

A_i является входным действием для X_i		Состояния			
		X_1	X_2	X_3	X_4
Действия	A_1	X_1			
	A_2		X_2		
	A_3			X_3	
	A_4			X_3	
	A_5				X_4

Процедуру определения множества действий, которые являются входами для данного состояния, можно задать с помощью табл. 10.

Таким образом, в целом модель сценариев некоторого приложения может быть представлена в виде системы с отношениями:

$$\boldsymbol{SM\ SKB} = < WM, \boldsymbol{X}, \boldsymbol{U}_{AX}, \boldsymbol{V}_{XA} >, \qquad (3)$$

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

где, как и ранее, WM – множество действий (атомарных или сложных), $\boldsymbol{X}$ – множество состояний; $\boldsymbol{U}_{AX}$ – отношение непосредственного предшествования, $\boldsymbol{V}_{XA}$ – отношение непосредственного следования на множестве состояний.

Табл. 7. **Пример матрицы смежности для отношения $\boldsymbol{V}_{XA}$**

X_i является входным состоянием для A_i		Состояния			
		A_1	A_2	A_3	A_4
Состояния	X_1	A_1			
	X_2		A_2		
	X_3			A_3	
	X_4			A_3	
	X_5				A_4

Сделаем одно важное замечание. Поскольку введенные отношения полностью задают порядок на множествах действий и состояний, то существует простая процедура автоматического восстановления графического образа синтаксической структуры сценариев, задаваемых четверкой (3). Покажем это на примере.

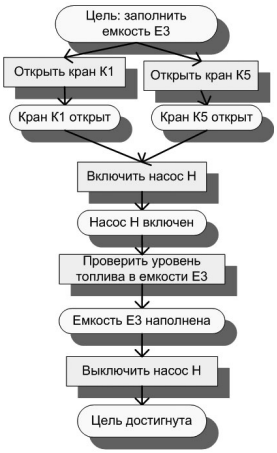


Рис. 10. Синтаксическая компонента сценария заполнения емкости Е3.

4.2. Пример. Спецификация сценарной базы знаний для фрагмента заправочного комплекса. Рассмотрим пример работы фрагмента заправочной станции, схема которого представлена на рис. 8. Пусть необходимо заполнить емкость $E3$ топливом из емкости $E1$. Синтаксическое представление сценария заполнения емкости $E3$, с использованием кранов $K1$ и $K5$, представлено на рис. 10.

Анализируя сценарий, можно видеть, что для заполнения емкости $E3$ топливом из емкости $E1$, необходимо открыть краны $K1$, $K5$, включить насос Н. При этом краны могут открываться параллельно, а

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Рис. 11. Описание фрагмента сценария в графической нотации языка *SKBL*.

насос – только после завершения обоих действий. Во время наполнения емкости *E3* необходимо контролировать уровень топлива в емкости.

Опишем в терминах языка *SKBL* содержание процесса взаимодействия двух узлов, а именно *узлов-состояний* «*Цель: заполнить емкость E3*« и «*Кран K1 открыт*», и *узла-процесса* «*Открыть кран K1*« (рис.11).

Как видно, в процессе открытия крана *K1* может возникнуть две нештатные ситуации:

1.Кран не был открыт за требуемое время;

2. Кран открыт, но топливо не поступает.

Если возникла нештатная ситуация, то первое, что должна обнаружить модель заправки, это сам факт наличия нештатной ситуации. Это делается средствами самой модели по получении соответствующей информации, например, о нештатной задержке процесса окончания заправки емкости. После этого должна быть включена система диагностики нештатной ситуации и после обнаружения причины должен быть запущен соответствующий сценарий возвращения системы в штатный режим работы. Он может, например, заключаться в поиске и использовании других возможных способов заполнения

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

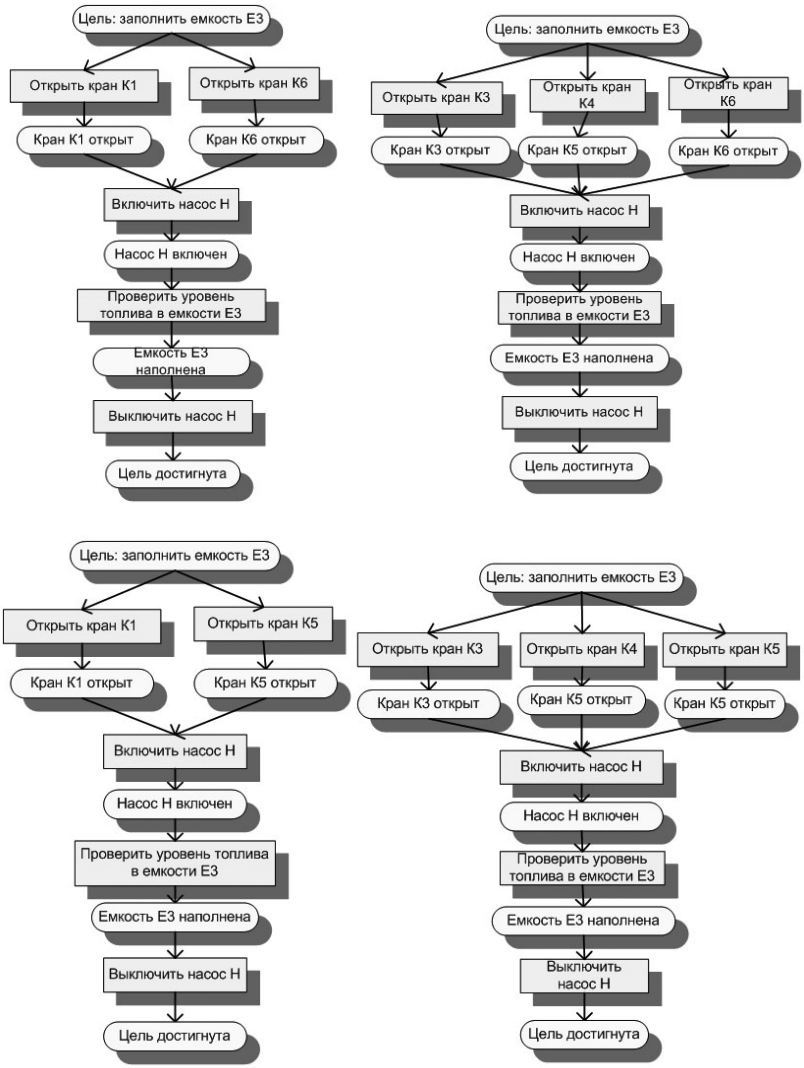


Рис. 12. Множество возможных сценариев заправки емкости $E3$.

емкости $E3$. Оказывается, что сценарная модель знаний такую возможность предоставляет. Покажем это, анализируя структуру

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

фрагмента заправочного комплекса, представленного на рис. 8. Анализ показывает, что эта структура содержит в себе возможность заправки емкости $E3$ еще по нескольким вариантам, каждому из которых соответствует некоторый сценарий. Всего может быть составлено четыре сценария, представленные на рис. 12.

Можно показать, что все эти сценарии, будучи описанными в терминах синтаксической структуры (3) и последующего восстановления ее графического образа представляются одной структурой сценарной базы знаний. Она представлена на рис. 13.

Этот пример достаточно ясно демонстрирует основную идею разработанного языка спецификации процессов, а именно возможность погружения в единую модель множества вариантов реализации сценария. Например, если в процессе выполнения сценария одна из операций не завершилась успешно, то система имеет

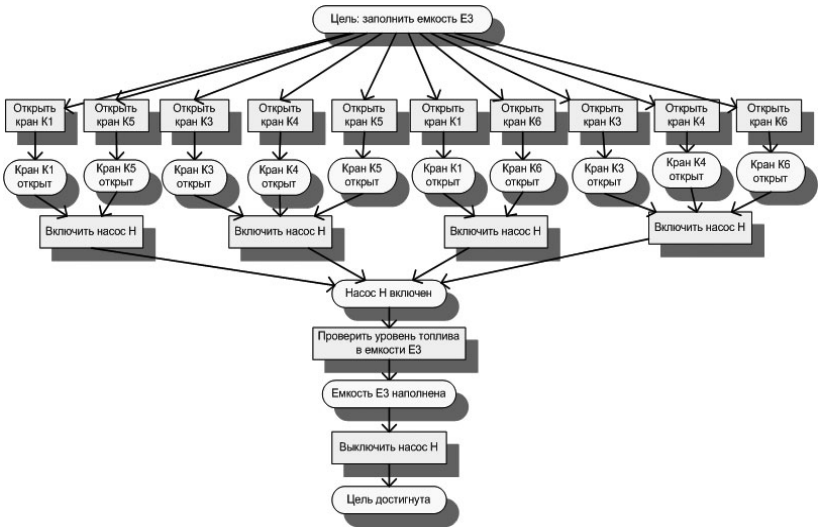


Рис. 13. Сценарная база знаний о заправке емкости $E3$.

необходимую информацию для того, чтобы перейти в состояние «*Нештатная ситуация*». Сообщение об этом может быть подано на пульт управления оператору. Ему также может быть сообщена подробная информация о нештатной ситуации, например, о типе

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

нештатной ситуации, результатах диагностики причин[1] и об альтернативных вариантах достижения цели. Оператор, таким образом, может получить всю необходимую информацию и рекомендации по последующим действиям.

Заметим, что в других приложениях выбор варианта продолжения сценария может делаться автоматически, например, путем установления некоторой системы предпочтений на множестве возможных вариантов продолжения, которые могут быть построены с помощью соответствующего механизма вывода, который сводится к поиску по запросу к структуре, описывающей синтаксическую модель сценарной базы знаний.

5. Заключение. Анализ существующих языков моделирования (см. раздел 2), показывает, что они не в состоянии специфицировать интеллектуальное поведение, поскольку все возможные варианты и ветви развития процесса, а также варианты управления должны быть перечислены заранее. Однако решение задач, которые выдвигаются современными приложениями, в особенности, в части обнаружения и обработки нештатных ситуаций невозможно без использования моделей интеллектуального поведения.

Основной результат данной работы – это язык спецификации сценарного поведения, который позволяет формально описывать поведение системы в случае отклонений от штатного режима, когда выбранный сценарий либо не может быть завершен успешно, либо в системе возникает нештатная ситуация. Эти возможности языка обеспечены использованием модели представления сценарных знаний, которая позволяет в реальном времени оценивать текущую ситуацию, прогнозировать ее развитие и выбирать новое продолжение решения задачи в зависимости от достигнутых состояний и других атрибутов процесса, например, наличия необходимых ресурсов. Для разработанного языка предложена графическая нотация, которая принимает во внимание рекомендации OMG. Для решения задачи динамического выбора продолжения исполнения сценария разработаны механизмы вывода, использующие свойства отношения частичного порядка, Однако в данной работе они не представлены, поскольку описание этих механизмов потребовало бы много места.

Ближайшей задачей дальнейших исследований по данному направлению является программная реализация языка, которая

[1] Модель диагностики является внешней компонентой по отношению к сценарной базе знаний, а потому здесь не рассматривается.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

позволит специфицировать поведение системы в графическом стиле и автоматически генерировать программный код в терминах языка высокого уровня (типа C++ или JAVA). Второй задачей является детальное исследование выразительных возможностей языка и эффективности используемых механизмов вывода на практических задачах из различных предметных областей.

Благодарности. Авторы работы выражают благодарность проф. Б.К.Гранкину, который сформулировал ряд интересных типовых задач, возникающих при подготовке и пуске ракет на стартовом комплексе. Это позволило сформулировать требованию к языку и его проверку пока на простых примерах из указанной области.

Литература

1. *Aalst W., Hofstede A.* YAWL: Yet Another Workflow Language, Information Systems, 30(4), 245-275, 2005.
2. *Aalst W.* The application of Petri Nets to Workflow management. Journal of Circuits, System and Computers, 8(1) 21-66, 1998.
3. *Gorodetsky V., Kotenko I.* Scenarios Knowledge Base: A Framework for Proactive Coordination of Coalition Operations. (Eds. M.Pechoucek and A.Tate) Proceedings of the Third International Conference on Knowledge Systems for Coalition.
4. *Object Management Group.* Business Process Modeling Specification. http://www.bpmn.org/.
5. *Peterson L.* Petri Net. Theory and the Modeling of Systems, Prentice Hall, 1981.
6. *Russel N., Holfstede A., Aalst W., Mulyar N.* Workflow Control Flow patterns: A Revised View. Technical Report, BPM Center Report BPM-06-22, http://is.tm.tue.nl/staff/wvdaalst/BPMcenter/reports/2006/BPM-06-22.pdf .
7. *Scheer A.* ARIS-Business Process Framework, Springer, 1999.
8. *Weske M.* Business Process Management: Concepts, Languages, Architectures. Springer, 2007, 368 pp.
9. YAWL data http://www.workflowpatterns.com/patterns/data/index.php.
10. YAWL exception http://www.workflowpatterns.com/patterns/exception/index.php.
11. YAWL patterns http://www.workflowpatterns.com/patterns/control/index.php.
12. YAWL resource http://www.workflowpatterns.com/patterns/resource/index.php .
13. Yet Another Workflow Language. http://www.yawl-system.com/.

Троцкий Денис Васильевич — аспирант лаборатории интеллектуальных систем Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН). Область научных интересов: сценарное управление, командная работа агентов, инструментальные средства многоагентных систем. Число научных публикаций — 1. Адрес электронной почты: denis.trotsky@iias.spb.su; СПИИРАН, 14-я линия В.О., д. 39, г. Санкт-Петербург, 199178, РФ; р.т. +7(812) 323-3570, факс +7(812)328-4450. Научный руководитель — В.И. Городецкий.

Trotsky Denis Vasilevich — PhD student of Intelligent System Laboratory of The St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research interests: scenario management, team work of agents, multi-agent software tools. The number of publications — 1. E-mail address: denis.trotsky@iias.spb.su;

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

SPIIRAS, 39, 14-th Liniya V.O., St. Petersburg, 199178, Russia; office phone +7(812)323-3570, fax +7(812)328-4450.

Городецкий Владимир Иванович — главный научный сотрудник лаборатории интеллектуальных систем Санкт-Петербургского института информатики и автоматизации РАН. Окончил Ленинградскую военно-воздушную инженерную академию им. А.Ф.Можайского в 1960 году и математико-механический факультет Ленинградского государственного университета в 1970 году. Доктор технических наук, профессор, Заслуженный деятель науки Российской Федерации. Опубликовал более 250 работ, 9 монографий и учебных пособий. Область научных интересов: теория и технология многоагентных систем, инструментальные средства для проектирования, программирования и развертывания многоагентных систем, многоагентные приложения в области защиты компьютерных сетей, объединения данных из гетерогенных источников, оценки ситуаций, управления воздушным движением, методы и средства интеллектуальной обработки данных и извлечения знаний, методы распределенного обнаружения знаний в данных, peer–to–peer системы, распределенное принятие решений. Адрес электронной почты: gor@iias.spb.su, СПИИРАН, 14-я линия В.О., д. 39, г. Санкт-Петербург, 199178, РФ; р.т. +7(812)323-3570, факс +7(812)328-4450.

Gorodetsky Vladimir Ivanovich — Prof. of Computer Science, Chief Scientist of The Intelligent Systems Laboratory of The St. Petersburg Institute for Informatics and Automation of the Russian Academy of Science. Graduated from the Military Air Force Engineer Academy in St. Petersburg (1960) and Mathematical and Mechanical Department of the St. Petersburg State University (1970), received his Ph.D. degree (1967) and Doctor of Technical Sciences degree (1973) in the area «Space Vehicle Optimal Control». Main publications (more 250) are related to the areas of Optimal control system theory, Space flight mechanics, Applied statistics, Artificial intelligence, in particular, Distributed intelligence, Knowledge-based planning and scheduling, Pattern and knowledge discovery in databases, P2P data mining and knowledge discovery, Data and information fusion, Decision combining, Distributed classification, Multi-agent system technology and software tools, P2P agent systems and P2P Agent platform, and, Service-oriented multi-agent systems, Multi-agent applications, in particular, Computer network security, Air traffic control, Project management, Digital image steganography. E-mail address gor@iias.spb.su, SPIIRAS, 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812)323-3570, fax +7(812)328-4450.

Труды СПИИРАН. 2009. Вып. 8. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2009. Issue 8. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru