

**ФОРМИРОВАНИЕ КОНЦЕПЦИИ ПОСТРОЕНИЯ ИМИТАЦИОННЫХ
МОДЕЛЕЙ ИСПОЛНЯЕМЫХ БИЗНЕС-ПРОЦЕССОВ****Е. Л. Зайцева (Санкт-Петербург)****Введение**

В настоящее время формируется современное направление в построении информационных систем на основе использования BPMS (Business Process Management Systems) – систем управления бизнес-процессами. BPMS, или BPM-системы обеспечивают кратчайший путь от формулировки бизнес-процесса до его исполнения.

Любая информационная система предназначена для поддержки бизнес-процессов организации. Поэтому задача формирования эффективных бизнес-процессов является актуальной как на этапе внедрения, так и при дальнейшем сопровождении информационной системы. Обычно эту задачу приходится решать многократно, реагируя на изменение внешних условий, требований и иных аспектов функционирования организации. Реальный способ точной количественной оценки эффективности сложных бизнес-процессов – имитационное моделирование. Построение имитационных моделей – это трудоемкая работа, которую желательно автоматизировать, для того чтобы иметь возможность оперативно оценивать целесообразность внесения изменений в бизнес-процессы, исполняемые BPM-системой. Таким образом, любая коррекция исполняемого бизнес-процесса может быть тут же подвергнута анализу с помощью программно сгенерированной имитационной модели. Принципиальным моментом является тот факт, что и BPM-система и разрабатываемый программный комплекс используют одно и то же входное представление бизнес-процесса.

Исходя из этого, целью данной работы является формирование концепции по автоматизированному созданию имитационных моделей бизнес-процессов. Необходимо определить наиболее подходящий и современный формат для начального описания процесса и выбрать систему, в которой будет анализироваться автоматически построенная модель. Также надо представить, каким именно образом эта концепция может быть реализована в рамках выбранных средств.

Обзор и выбор форматов для моделирования бизнес-процессов

Для начала необходимо выбрать наиболее подходящий формат для статического описания процесса. Главные критерии выбора – это современность, распространённость, возможность программного анализа. При этом формат должен быть связан с системами исполнения бизнес-процессов.

Основные классические стандарты моделирования бизнес-процессов – это нотации IDEF0, IDEF3, DFD; EPC, eEPC (стандарты ARIS). Первые три стандарта используются в системе моделирования BPWin (AllFusion). Это современное средство моделирования процессов с богатыми возможностями. Однако возможность дальнейшего программного анализа модели несколько ограничена. Начиная ещё с более старых версий, система поддерживает функции генерации отчётов. С помощью таких отчётов, шаблон которых надо предварительно настроить, можно получить сведения о модели и её структуре. Известна работа, в которой на основе модели из BPWin и сгенерированного отчёта, строилась имитационная модель для GPSS¹. Недавно появилась и другая возможность – это импорт модели в формат xml и дальнейший анализ полученного файла. Исследование такого представления модели показало, что оно содержит в себе полную характеристику модели и информацию о её структуре, но способ представле-

¹ Рахалов Д. А. Разработка пакета программ для имитационного моделирования бизнес-процессов.

ния оказался довольно запутанным. Вместе с этим классические стандарты моделирования не имеют прямой связи с системами исполнения бизнес-процессов и поэтому не особо подходят в качестве начального формата для построения имитационной модели.

Некоторое время назад к классической линейке присоединился новый, более современный стандарт – BPMN. Основная цель BPMN – создание стандартной нотации, понятной всем бизнес-пользователям. Бизнес-пользователи включают в себя бизнес-аналитиков, создающих и улучшающих процессы, технических разработчиков, ответственных за реализацию процессов, и менеджеров, следящих за процессами и управляющих ими. Следовательно, BPMN призван служить связующим звеном между фазой дизайна бизнес-процесса и фазой его реализации.

Так как BPMN – это всего лишь нотация, то есть графическое представление процесса, то для работы с ней требуется наличие унифицированного файлового представления. Таким средством является язык XPDЛ.

В итоге, можно утверждать, что нотация BPMN соответствует заявленным критериям, таким как современность, распространённость. Также возможен программный анализ модели при переходе к языку XPDЛ, а XPDЛ, в свою очередь, используется как промежуточный формат для различных систем исполнения и моделирования.

Следующий шаг – это поиск программного продукта, который бы обеспечил комфортное моделирование в нотации BPMN и имел возможности экспорта модели на язык XPDЛ.

Список продуктов, который поддерживает нотацию BPMN, достаточно велик и насчитывает более 50 позиций. Среди всех программ заметно выделяется бесплатная программа под названием BizAgi. В ней представлены почти все возможности BPMN, а также поддерживается импорт и экспорт моделей в различные форматы. Для BPMN существует унифицированное представление в виде xml-документа с помощью специального языка – XPDЛ. Он также обеспечивает связь между системами моделирования процессов и BPMN-системам, нацеленными на исполнение процесса. Преобразование BPMN<->XPDЛ работает в обе стороны. В данной программе элементам модели можно присваивать свои (пользовательские) переменные и указывать их значение. Пользовательские переменные необходимы при анализе бизнес-процесса с целью формирования его имитационной модели, так как задают временные, стоимостные и другие его параметры.

Выбор конечной модели и программы для имитационного моделирования

Выбор парадигмы моделирования

Приведём список основных современных парадигм моделирования:

- 1) объектно-ориентированная. Предполагает написание модели на ООП языке высокого уровня, например на языке Java;
- 2) процедурно-ориентированная. Пример – система Netlogo. Модель описания процессов в системе с помощью процедур;
- 3) транзактно-ориентированная. Есть схема модели, а есть транзакты, или заявки, которые перемещаются по ней. Пример системы – GPSS;
- 4) агентная. Система, которая моделируется не через глобальное задание структуры, а с помощью описания ей составных частей, которые существуют в модели одновременно и совершают действия без глобального контроля. Примеры – системы AnyLogic и RepastJ.

Выбор парадигмы моделирования тесно связан с выбором системы моделирования, поэтому окончательное решение этого вопроса возможно после рассмотрения современных систем моделирования.

Выбор системы моделирования

В рамках общей концепции генератора можно рассматривать «мультигенерацию» имитационных моделей (GPSS, MOBIUS, T-net, Anylogic, Repast). Это означает, что по одному и тому же описанию бизнес-процесса на языке XPDЛ можно построить совершенно разные имитационные модели. Они будут отличаться по парадигмам и системам моделирования. На данной стадии работы предполагается формирование концепции построения имитационной модели для одной из этих систем. Остальные ветви – предмет дальнейшего рассмотрения.

Существует множество систем имитационного моделирования. Одни из самых известных и распространённых – GPSS, Arena, Anylogic. Выбор системы обусловлен такими критериями, как возможность построения модели пользователем с нуля, открытость кода, удобство использования, универсальность, современность.

GPSS – это давно созданная система, постоянно обновляющаяся. Последняя коммерческая версия появилась около двух лет назад. В студенческой версии есть ограничения по числу блоков в модели, хотя эта система и предполагает написание модели в открытом, текстовом виде.

Системы ARENA выдает код модели на языке SIMAN, который очень похож на GPSS. Ограничение состоит в том, с этим кодом нельзя работать – измененный код уже не запустится с помощью ARENA. В ARENA можно экспортировать специально подготовленные IDEF3-модели из системы BPwin.

AnyLogic – это коммерческая система ограниченной доступности (в рамках оценочной версии). Система является java-ориентированной, но код её закрыт.

К классу агентных систем можно отнести не столь распространённую, но современную, объектно-ориентированную и чрезвычайно гибкую систему под названием Repast. Она также основана на языке Java, как и AnyLogic, но обеспечивает пользователю больше свободы. Код модели пишется в открытом виде, компилируется и запускается на исполнение. Система предоставляет несколько классов–библиотек для специфических задач моделирования систем.

Система Repast.J

Repast – библиотека, предоставляющая средства для агентно-ориентированного моделирования систем. Она написана на нескольких языках программирования: Repast.J (Java based), RepastPy (based on the Python Scripting language), Repast.Net (implemented in C# but any .Net language can be used), RepastS (Repast Symphony, Java based). Поскольку одна из библиотек написана на Java, возможен запуск системы на разных ОС.

Средой для создания модели может служить любой текстовый редактор. Затем модель следует откомпилировать программой javac. Для более комфортного проектирования можно использовать специализированные программы, такие как NetBeans, Eclipse, и JBuilder. Следует отметить, что в данных IDE присутствуют средства графической разработки программ, а именно возможна генерация классов из UML диаграммы.

Обычно для моделирования системы создают 3 класса, но классов может быть и больше:

- Класс агента, в котором описываются переменные агента, функции, отвечающие за его поведение, функции для передачи и приема параметров из агента;
- Класс окружения – в нем описывается среда, в которой находится агент, (не обязательно);
- Класс модели – в нем создаются агенты и среда, а также описываются входные параметры модели, графическое представление системы, создается очередь событий.

Структура конечной имитационной модели

Конечная имитационная модель будет построена специально для системы RepastJ. Эта система относится к агентным системам имитационного моделирования. В данном контексте это означает, что в мире модели ее субъекты ведут себя некоторым образом, без участия глобального контроля. Проектируемая модель похожа на событийно-ориентированную систему с сохранением агентной парадигмы. То есть в модели присутствуют приборы-исполнители, которые ведут себя в соответствии с определённым строением мира, то есть самой модели.

В качестве основного компонента модели будет рассматриваться офис (либо компания), в которой работает определённое количество людей. Люди – это те самые приборы-исполнители, свойственные для любой системы массового обслуживания. Каждый исполнитель связан со своими непосредственными предшественниками и последователями, тем самым задаётся структура перемещения заявок от одного прибора к другому. Каждый исполнитель имеет свою очередь ограниченного объёма, обладает временными и стоимостными характеристиками. Таким образом, за каждым человеком закреплена определённая работа. В системе необходимо присутствие класса заявок, которые будут перемещаться по модели, могут быть исполнены в течение некоторого времени, а могут и находиться в очереди на исполнение.

RepastJ хорош тем, что его модели строятся на основе объектно-ориентированного языка Java, тогда как объектно-ориентированная модель имеет много общего с агентной системой. RepastJ предоставляет несколько классов–библиотек для специфических задач моделирования систем, тем самым облегчая создание модели, если она пишется «с нуля».

Заключение

В ходе исследования были сформированы представления о нотациях, языках, системах моделирования бизнес-процессов. В результате были выбраны нотации, языки и системы моделирования, подходящие для дальнейшей работы. Таковыми являются: нотация BPMN для начального описания бизнес-процессов, язык XPDЛ для промежуточного представления процесса, его анализа, загрузки в систему исполнения бизнес-процессов, агентно-ориентированная система моделирования RepastJ. Также известен подход к анализу файлов в формате XPDЛ – функции парсера из семейства JDOM.

Предметом дальнейшей работы будет определение набора программных продуктов для построения имитационных моделей различных парадигм (RepastJ, стохастические сети, аналитические решения). Комплекс будет состоять из нескольких частей. Каждая часть будет отвечать за построение имитационной модели в рамках одной из упомянутых парадигм и будет использовать один и тот же исходный модуль, который осуществляет обработку входного xpdл-описания бизнес-процесса – определяет структуру процесса и его параметры.

Построение модели на основе агентной парадигмы будет выглядеть как сборка конечной системы из заранее описанных классов. Такими классами являются классы работ. После компиляции имитационная модель может быть запущена в системе моделирования RepastJ с целью получения её характеристик.

Аналитическая модель может быть представлена в форме нелинейной неоднородной сети массового обслуживания, рассматриваемой в качестве «исполнителя» бизнес-процесса. Она обеспечивает расчет основных временных и стоимостных характеристик процесса.

Также может быть осуществлён стоимостной анализ системы, который основан на использовании концепции Activity Based Costing. Такой анализ может пригодиться при оптимизации затрат исполнения бизнес-процесса.