

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СЛОЖНЫХ ДИНАМИЧЕСКИХ СИСТЕМ В ИНТЕГРИРОВАННОЙ ИНСТРУМЕНТАЛЬНОЙ СРЕДЕ MATLAB + SIMULINK + STATEFLOW

А. Д. Десятов, А. А. Сирота

Воронежский государственный университет

В статье рассматриваются возможности интегрированной инструментальной среды, построенной на основе Matlab, Simulink и Stateflow и применение ее для моделирования службы поддержки на предприятии по оказанию информационных услуг. Разработан подход для динамического автоматизированного конструирования моделей и предложен способ изменения структуры моделей в режиме симуляции.

Современные ЭВМ, вычислительные комплексы и сети являются мощными средствами исследования сложных систем с использованием технологий имитационного моделирования.

Общими тенденциями, реализуемыми при разработке современных инструментальных средств и систем имитационного моделирования сложных динамических систем, являются:

- обеспечение визуального конструирования;
- совмещение средств и систем имитационного моделирования с интеллектуальными системами поддержки принятия решений (экспертные системы, САПР и т. п.);
- обеспечение возможностей анимации при отображении процессов функционирования моделируемых систем;
- интеграция с CASE-технологиями, конструирование многоуровневых моделей систем в рамках методологии структурного системного анализа [5].

Одним из представителей современных инструментальных средств для проведения имитационного моделирования является система Matlab. При этом имеет смысл говорить не о языке Matlab, а о некоей интегрированной среде, в которую входят Simulink — инструмент, обеспечивающий визуальное конструирование, для моделирования, имитации и анализа динамических систем, Stateflow, как средство для моделирования поведения, вызванного событиями, а также компоненты, написанные с использованием универсальных языков программирования высокого уровня и подключаемые в модель посредством Matlab Application Programming Interface (Matlab API).

Для исследования возможностей проведения имитационного моделирования в интегрированной среде Matlab + Simulink + Stateflow в качестве примера системы в данной работе будет рассмотрена задача моделирования службы поддержки на предприятии по оказанию информационных услуг. Основной целью проведенного исследования при этом является анализ возможностей автоматизированного создания моделей многоуровневых и многокомпонентных систем с изменяющейся во времени структурой.

Служба поддержки (Service Desk) — это некоторая диспетчерская служба, которая в полной мере ответственна перед клиентом или пользователем за предоставление согласованных с ним сервисов, является центром приема всех жалоб и предложений, осуществляет контроль текущего состояния сервисов и имеет полномочия по выдаче нарядов на устранение возможных сбоев, а также на контроль процесса устранения неисправностей. В самом общем виде данная служба может осуществлять перечисленные функции в отношении всех сервисов (а не только информационно-технологических), предоставляемых организацией и ее отдельными подразделениями [1].

Помимо службы поддержки (Service Desk) на предприятии существуют другие группы специалистов, обладающие специфическими знаниями, людскими или временными ресурсами, необходимыми для разрешения инцидентов. В процессе оказания услуг они могут занимать место второго или третьего уровня поддержки, в то время как служба поддержки является первым уровнем. Таким образом, необходимо построить имитационную модель системы, структура которой представлена на рис. 1, а

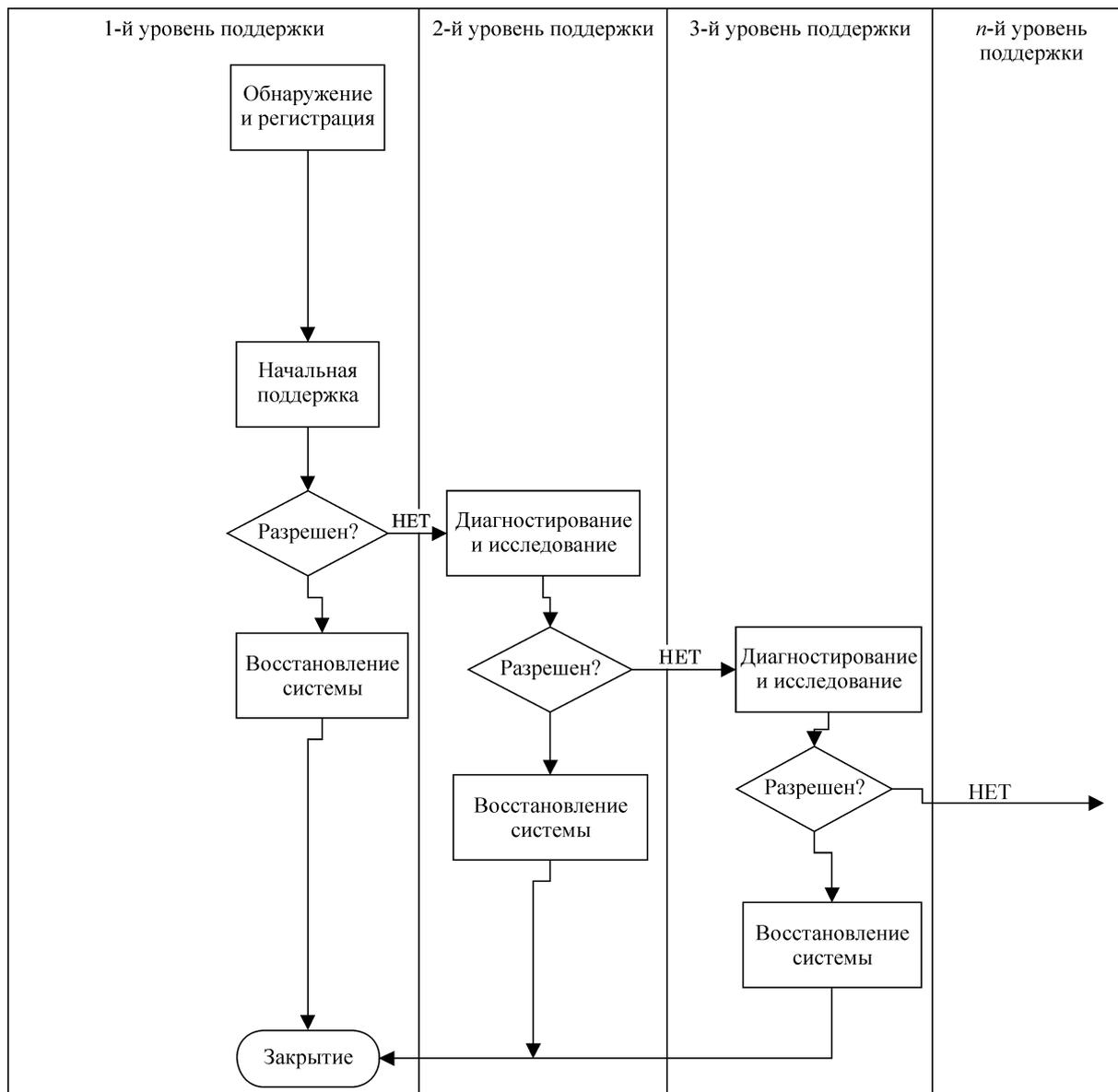


Рис. 1. Блок-схема работы многоуровневой службы поддержки

также провести ее испытания применительно к реальным условиям функционирования.

Служба поддержки на предприятии ответственна за отслеживание процесса разрешения всех зарегистрированных инцидентов. Инциденты, которые не могут быть решены службой поддержки самостоятельно, назначаются соответствующим группам специалистов, при этом устранение проблемы или поиск временного решения должны быть произведены как можно быстрее, чтобы потери пользователей сервиса были минимальными. После устранения причины нарушения нормального режима системы и восстановления функционирования сервиса инцидент закрывают.

Моделирование подобной системы было выполнено в среде MATLAB и оформлено в виде m-файла управляющей программы (УП), содержащей обращение к mdl-файлу Simulink-модели (S-модели) с вложенной Stateflow моделью (SF-диаграммой). Роль УП состоит в задании варьируемых и неварьируемых факторов, проведении стратегического и тактического планирования модельного эксперимента, реализации процедур статистической обработки результатов моделирования. Роль S-модели заключается в обеспечении динамики процесса функционирования SF-модели и регистрации его результатов на основе использования стандартных элементов подсистемы Simulink. На-

конец, роль SF-модели (Stateflow-модели) состоит, собственно, в воспроизведении визуальной модели службы поддержки как системы массового обслуживания (СМО) с использованием формализма гибридных автоматов (карт состояний Харела) [5].

Помимо указанных компонентов в модель интегрировались компоненты, написанные на языке Java и использованные для реализации механизма классификации заявок, поступающих от пользователей сервиса. Благодаря интеграции Matlab и Java компоненты органично расширяют модель системы, и так как Java является объектно-ориентированным языком, то автоматически и модель получает всю мощь объектно-ориентированного подхода. Хотя Matlab и поддерживает ООП на уровне языка, интеграция с Java делает возможным использование реальных компонентов, используемых для автоматизации работы службы.

При построении модели службы поддержки необходимо учитывать, что полученная модель должна соответствовать организационной структуре предприятия, сервисную службу которого требуется смоделировать. Т. е. ставится задача воспроизвести многоуровневую систему поддержки с произвольным количеством элементов обслуживания (людских ресурсов) на каждом уровне. Учитывая, что невозможно предусмотреть и создать все возможные способы построения службы, то задача сводится к генерации требуемой модели на основе некоторых базовых компонентов, в которых определяются основные функциональные блоки и устанавливаются отношения между блоками.

Базовая модель содержит следующие блоки:

1. PointOfContact — отвечает за регистрацию всех заявок, поступающих на обработку.

2. ServiceLevel — инкапсулирует в себе процесс функционирования отдельно взятого уровня поддержки. Данный блок является основой при создании многоуровневых моделей.

3. DiagnosisDispatcher — распределяет заявки, поступающие в систему по уровням обслуживания с целью диагностирования проблемы

4. ResolutionDispatcher — распределяет заявки по уровням обслуживания и инициирует процесс разрешения проблемы

5. ControlBlock — отвечает за сбор информации (статистики), поступающей от различных элементов моделируемой системы

Общий вид компонентов базовой модели, реализованных в Stateflow, представлен на рис. 2.

Для решения задачи генерации многоуровневой модели предлагается использовать Stateflow Application Programming Interface (Stateflow API) — инструмент, позволяющий создавать и изменять Stateflow диаграммы посредством команд пакета Matlab. Stateflow API предоставляет возможность работы с основными графическими и неграфическими объектами диаграмм: State, Transition, Junction и т.д. Каждый из этих объектов имеет определенные методы и свойства, путем изменения которых происходит редактирование объектов диаграммы. Таким образом, становится возможной автоматизация процесса создания сложных моделей систем в Stateflow. Для этого необходимо написать процедуру генерации (script-файл на языке Matlab), оперирующую объектами диаграмм при помощи Stateflow API.

Процедура генерации получает на вход базовую модель службы поддержки, содержащую основные функциональные блоки. В результате выполнения script-файла, на выходе процедуры формируется многоуровневая Stateflow-модель, соответствующая структуре имитируемой службы поддержки. Следует отметить, что процедура генерации реализуется до запуска модели и результатом процедуры является модель, над которой впоследствии проводятся испытания. Кроме того, имеет смысл вынести информацию для конфигурирования получаемой динамической модели (количество уровней поддержки, количество блоков обработки на уровне) в отдельный файл (файл model.properties). В этом случае все данные, необходимые для создания требуемой модели, локализуются в одном месте. Перед запуском процедуры генерации настройка параметров модели происходит путем редактирования текстового документа model.properties.

Общая схема процесса создания модели представлена на рис. 3.

В основе процедуры генерации лежит метод, позволяющий создавать копии существующих объектов диаграммы. В этом плане Stateflow API предоставляет эффективный способ копирования — копирование сгруппированных состояний (Copying by Grouping). При копировании сгруппированного состояния, копируется и все дочерние состояния и переходы.

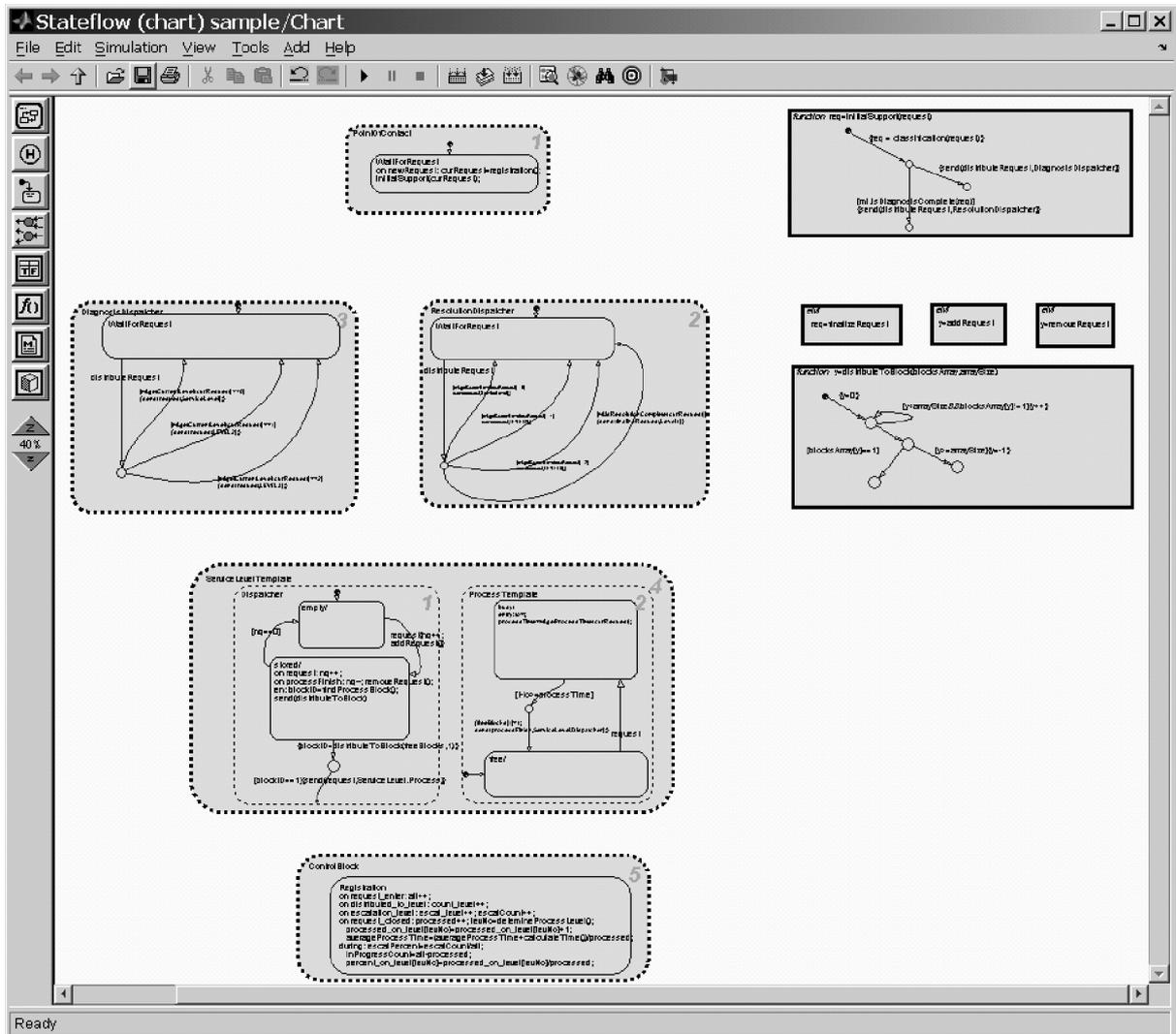


Рис. 2. Элементы базовой модели

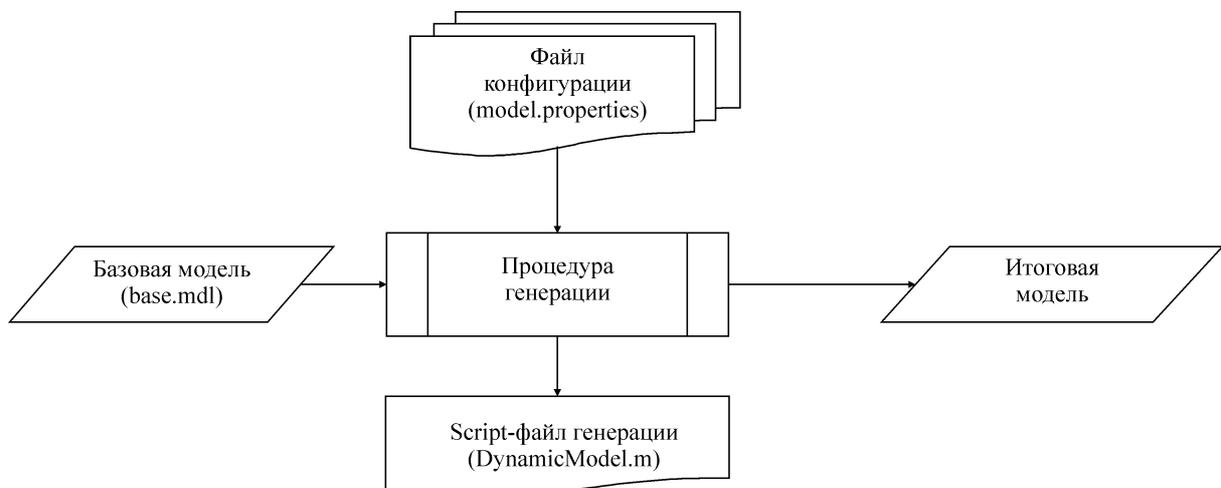


Рис. 3. Схема процесса создания многоуровневой модели

Ниже приведен пример копирования уровня обслуживания ServiceLevel — создание дополнительно уровня обслуживания на основе существующего:

```
% получаем значение свойства IsGrouped
% для состояния ServiceLevel
prevGrouping = ServiceLevel.IsGrouped
% если объект не был сгруппирован,
% сгруппируем его
if (prevGrouping == 0)
ServiceLevel.IsGrouped = 1
end
% получаем указатель на буфер
% (копирование происходит через буфер)
cb = sfclipboard
% копируем сгруппированный объект в буфер
cb.copy(ServiceLevel)
% вставляем объект из буфера на диаграмму
cb.pasteTo(chart)
% устанавливаем значение свойства IsGrouped
% в предыдущее значение
ServiceLevel.IsGrouped = prevGrouping
```

Тот же самый подход копирования сгруппированных состояний применяется и для создания необходимого числа блоков обработки на каждом уровне поддержки.

Многоуровневая модель, получаемая в результате запуска подобной процедуры генерации, представлена на рис. 4.

После того, как требуемая модель сформирована, становится возможным проведение имитационного моделирования. В ходе имитационного моделирования возможна оценка нескольких категорий показателей.

В качестве основных количественных показателей рассматриваются: среднее время обслуживания инцидентов, количество обработанных инцидентов (пропускная способность), количество заявок, время обработки которых превысило установленные нормы (количество эскалаций) и т. п. При исследовании основных количественных показателей, система поддержки рассматривается в качестве “черного ящика”, т.е. на данном уровне абстракции не раскрывается внутренняя структура системы. На вход подается поток событий, соответствующий моментам поступления инцидентов в систему, и на основании указанных показателей делается вывод о соответствии сервисной службы соглашению о предоставляемых услугах (SLA): соответствии среднего времени обработки заявок времени, указанно-

му в документе; наличие разрешенного процента эскалаций и т. п.

Показатели эффективности функционирования уровней обслуживания (например, процент заявок, разрешаемых на каждом уровне, количество эскалаций на уровне, коэффициент загрузки) раскрывают внутреннее устройство “черного ящика”, предоставляя информацию о способе обработки заявок в службе поддержки. Такого рода показатели помогают выявить узкие места в системе, наиболее загруженные уровни поддержки и т. п.

Рассмотрим для примера результаты оценки одного из показателей эффективности — среднего времени ожидания заявки в очереди.

На рис. 5 время ожидания заявки рассматривается для систем с различным количеством блоков обработки: с 3, с 5, и 7 блоками обработки (блоки Process модели, представленной на рис. 4). Время ожидания, установленное в SLA, равняется 30 минутам и нормальный режим функционирования службы поддержки подразумевает соблюдение данного ограничения. Соответственно, на основании графика можно сделать вывод, что система с 3 блоками не подходит, поскольку время ожидания превышает допустимые пределы, однако система из 5 блоков является оптимальной: хотя на незначительном по продолжительности периоде и произошло превышение времени ожидания, но в целом блоки обработки задействованы наиболее эффективно. Использование же 7 блоков обработки не является эффективным с экономической точки зрения.

Помимо динамического конструирования модели посредством Stateflow API, важное практическое значение имеет также возможность перестраивать структуру модели в режиме симуляции. Наличие данного механизма позволяет исследовать поведение моделируемой системы при добавлении/удалении элементов, а именно:

- изменение показателей эффективности при изменении структуры;
- скорость перехода системы в стабильный режим работы;
- загрузку блоков обработки, подключаемых при пиковой нагрузке.

Знание данных аспектов функционирования службы поддержки позволит предоставить информацию, необходимую для оптимального планирования ресурсов.

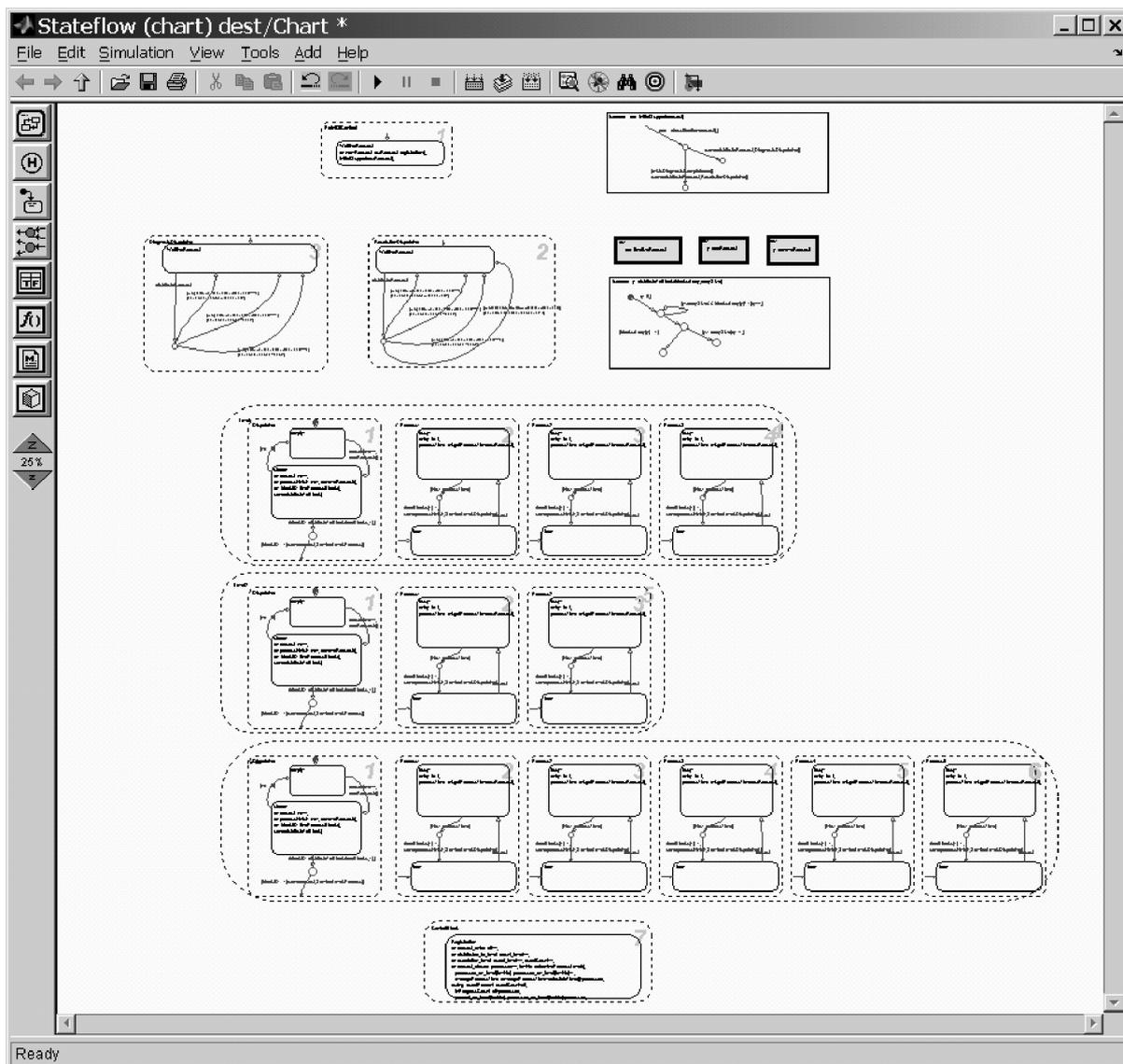


Рис. 4. Модель многоуровневой службы поддержки

Строго говоря, Matlab на уровне языка не поддерживает изменение структуры модели в режиме симуляции. Файл помощи говорит об этом следующее: «Невозможно вносить такие изменения в структуру модели, как удаление, добавление линий или блоков, во время симуляции. Если есть необходимость в проведении такого рода преобразований, следует остановить симуляцию, внести изменений и затем запустить модель снова». Однако данное ограничение можно обойти путем создания УП, назначением которой является управление режимом симуляции. Эти возможности предоставляет Simulink API, имеющий необходимый набор методов, позволяющих динамически запускать/останавливать модель.

Например, запуск модели производится из УП следующим образом: `set_param('<mdl_name>', 'SimulationCommand', 'start')`, где '<mdl_name>' — имя модели 'SimulationCommand' — имя изменяемого параметра.

'start' — новое значение параметра.

Соответственно, для останова модели необходимо выполнить команду:

`set_param('<mdl_name>', 'SimulationCommand', 'stop')`.

Однако нерешенной остается проблема сохранения данных — результатов работы модели при останове симуляции, а также их восстановления при запуске после изменения структуры. Для этих целей при реализации вполне естест-

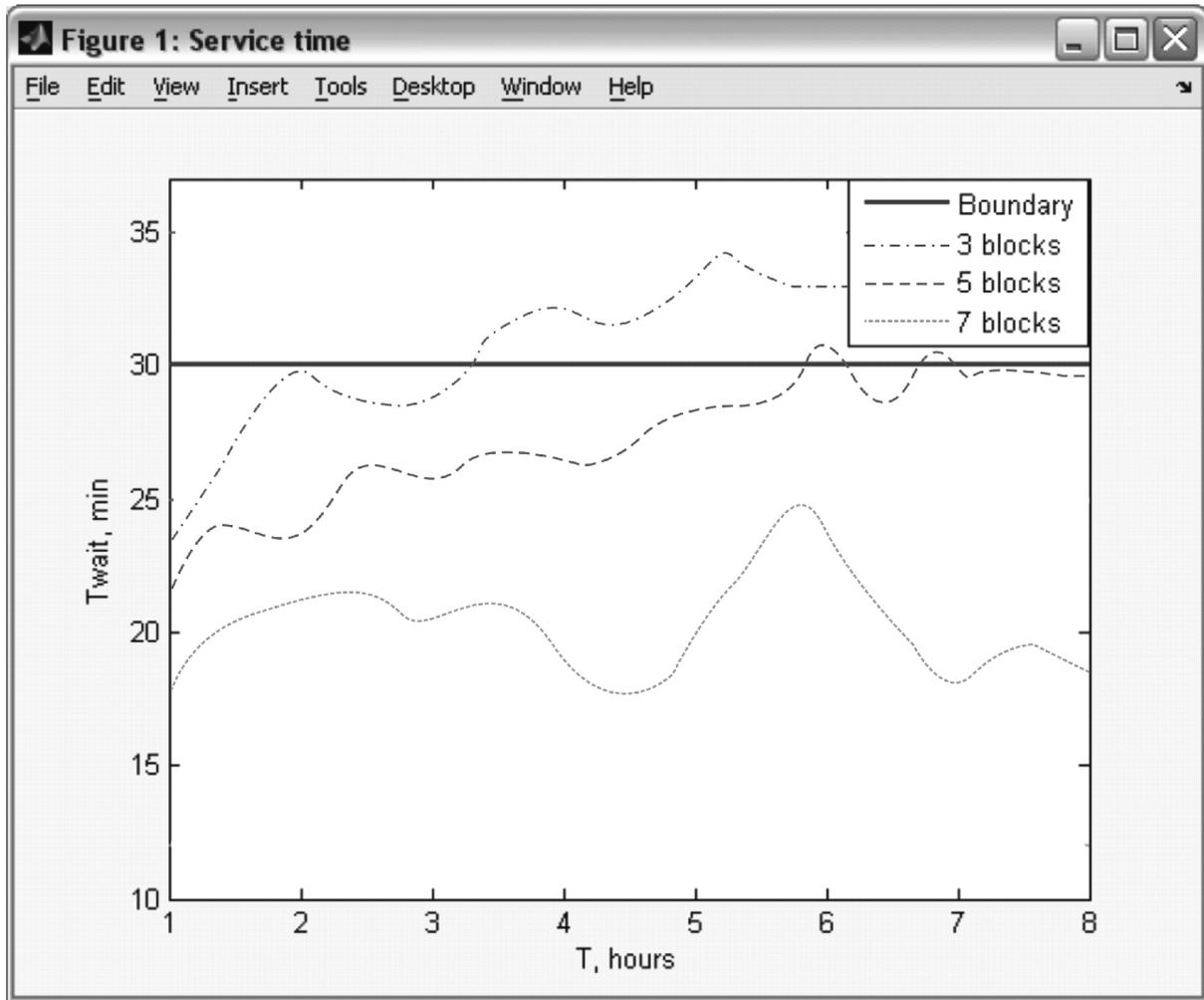


Рис. 5. Среднее время ожидания заявки в очереди

венно использовать Matlab Workspace — хранилище, содержащее множество переменных, созданных в процессе работы программы. Объекты Stateflow диаграммы при этом необходимо настроить в свою очередь таким образом, чтобы инициализация значений переменных, а также сохранение их значений при окончании симуляции происходили с использованием Workspace. Кроме того, использование Workspace обеспечивает сохранение результатов работы модели во внешней памяти и не ограничивает время жизни переменных рамками сессии Matlab. Механизм загрузки/выгрузки данных во внешнюю память предоставляется средой Matlab. Таким образом, оказывается принципиально возможно сохранить состояние перед формированием управляющего воздействия с последующим восстановлением исходного состояния в будущем, что существенно облегчает отладку модели. Общая схема взаимодействия

компонентов системы, реализующих данный подход, представлена на рис. 6.

Реализация вышеперечисленных вариантов организации и проведения имитационного моделирования отличается лишь местом вызова УП: при первом и третьем варианте УП вызывается из пользовательского интерфейса; при программно-управляемой симуляции — перед формированием управляющего воздействия из алгоритма принятия решений.

По результатам статьи были получены следующие выводы:

1. Были проанализированы инструментальные средства для проведения моделирования и выбор был остановлен на интегрированной среде MATLAB + Simulink + Stateflow, в связи с возможностями интеграции среды с компонентами, написанными при помощи современных языков программирования (Java, C++ и т. д.)

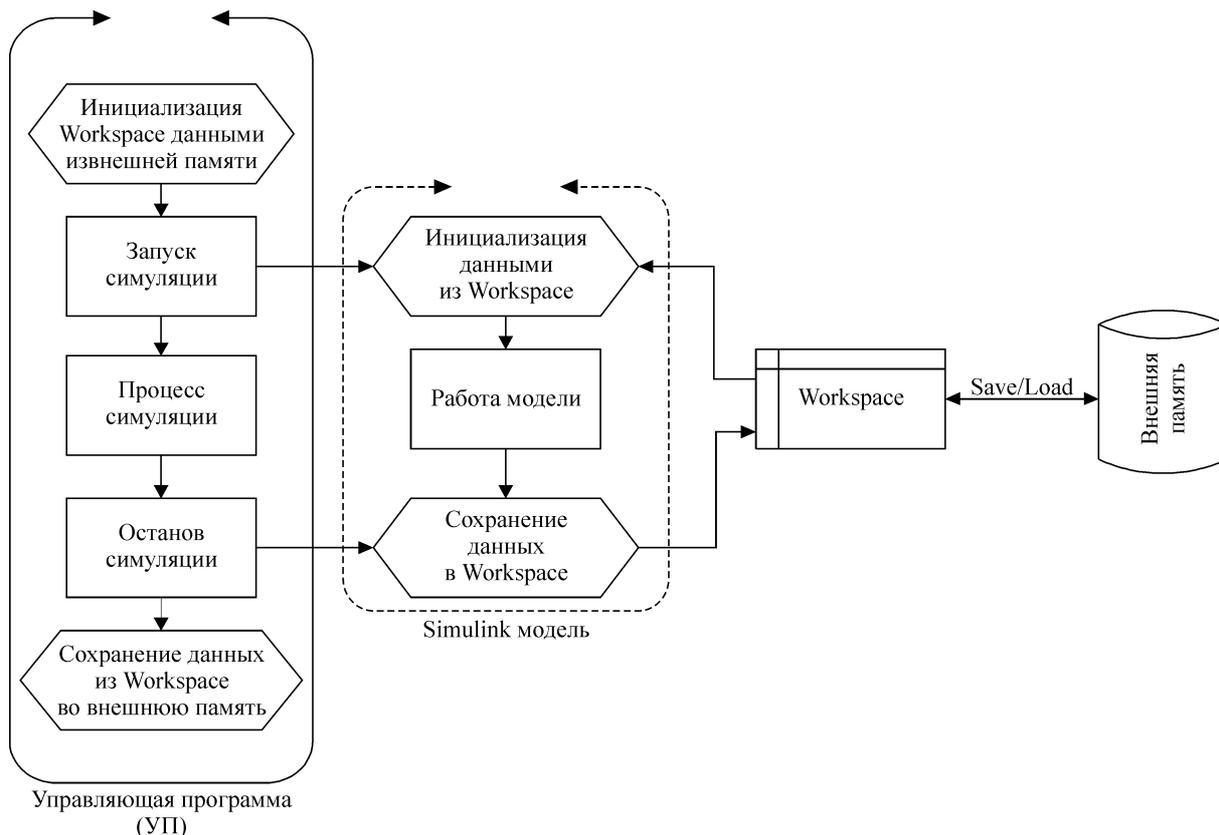


Рис. 6. Иллюстрация предлагаемого принципа изменения структуры модели

2. Была разработана концептуальная и функциональная модель службы поддержки на предприятии. Были выделены основные виды деятельности в рамках процесса поддержки функционирования сервиса.

3. В интегрированной среде MATLAB+ Simulink + Stateflow была создана имитационная модель, соответствующая функциональному описанию процесса. Кроме того, был найден механизм, основанный на Stateflow API, позволяющий расширить базовую модель системы с одним уровнем поддержки и одним блоком обработки до модели, позволяющей адекватно описать многоуровневую службу поддержки на предприятии. Также был предложен механизм изменения структуры модели во время проведения испытаний.

СПИСОК ЛИТЕРАТУРЫ

1. *Алехин З.* Service Desk — цели, возможности, реализации. / «Открытые системы». 2001, № 5—6.
2. *Бенькович Е.С.* Практическое моделирование динамических систем. / Е. С. Бенькович, Ю. Б. Колесов, Ю. Б. Сениченков. — СПб.: БХВ — Петербург, 2002. — 464 с.
3. *Гультияев А.А.* Визуальное моделирование в среде MATLAB. / А. А. Гультияев. — СПб: Питер, 2000. — 432 с.
4. *Лазарев Ю.Н.* Моделирование процессов и систем в MATLAB. / Ю. Н. Лазарев. — СПб.: БХВ (Петербург, Киев), 2005. — 512 с.
5. *Сирота А.А.* Компьютерное моделирование и оценка эффективности сложных систем. / А. А. Сирота. — М.: Техносфера, 2006. — 280 с.
6. *Советов Б.Я.* Моделирование систем. / Б. Я. Советов, С. А. Яковлев. — М.: Высшая школа, 2001. — 275 с.