

ЭКСТРЕМАЛЬНОЕ ПРОГРАММИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ В АСДУ ЭЛЕКТРИЧЕСКИМИ СЕТЯМИ

АВТОРЫ:

Ю.Я. ЛЮБАРСКИЙ,
Д.Т.Н.,
АО «НТЦ ФСК ЕЭС»

Ю.И. МОРЖИН,
Д.Т.Н.,
АО «НТЦ ФСК ЕЭС»

Н.М. АЛЕКСАНДРОВ,
НПП «ДИНАМИКА»

Автоматизация рассуждений диспетчерского персонала при анализе нештатных технологи-

ческих ситуаций является самым трудоемким этапом создания интеллектуальных систем диспетчерского управления электрическими сетями.

Ключевые слова: автоматизированная система диспетчерского управления (АСДУ); электрические сети; интеллектуальное диспетчерское управление; экстремальное программирование.



Современные умные электрические сети активно используют интеллектуальные агенты — программы, помогающие диспетчерскому персоналу в нештатных ситуациях

ПОСТАНОВКА ЗАДАЧИ

Создание умных электрических сетей связано с необходимостью использования интеллектуальных агентов — программ для оказания помощи диспетчерскому персоналу в нештатных ситуациях [1]. Для этого требуется «преобразование» опыта эксплуатационного персонала в соответствующие программные средства, что в свою очередь связано с весьма трудоемкой разработкой, основанной на тесном сотрудничестве программистов, специалистов-технологов, инженеров по знаниям [2]. Желательно при этом минимизировать объем работы профессиональных программистов, не увеличивая значительно объем работы эксплуатационников.

ЕСТЕСТВЕННЫЙ ЯЗЫК В ЛОГИКЕ ДИСПЕТЧЕРСКОГО УПРАВЛЕНИЯ

Рассмотрим (без ограничения общности) задачи обнаружения и ликвидации неисправностей в качестве основных задач диспетчерского управления в энергетических системах и электрических сетях.

Ликвидация технологического нарушения — это:

- выявление и отделение поврежденного оборудования или участка сети;
- устранение опасности для обслуживающего персонала и оборудования, не затронутого технологическим нарушением;
- предотвращение развития технологического нарушения;
- восстановление в кратчайший срок электроснабжения потреби-

телей и качества электроэнергии (частоты и напряжения);

- создание наиболее надежной послеаварийной схемы сети и отдельных ее частей;
- выяснение состояния отключившегося оборудования и возможности включения его в работу [3].

При выявлении поврежденного оборудования диспетчер использует различные информационные системы типа EMS, содержащие расчетные модели с отображением схем сети и энергетических объектов. В условиях дефицита времени диспетчер может испытывать затруднения с идентификацией повреждений и определения их причин. В ряде случаев требуются «диспетчерские рассуждения» (иногда весьма сложные и чреватые «человеческими ошибками»). Телефонные переговоры диспетчера часто отражают эти затруднения. Задать в трудной ситуации вопрос (или серию вопросов, организованных в «рассуждение») компетентной интеллектуальной информационной системе — может оказаться хорошим выходом. Но на каком языке задавать такие вопросы? Очевидно, что в этом случае не может использоваться тот или иной язык программирования — это за пределами диспетчерской компетенции. Лучшим вариантом является использование естественного (в нашем случае, русского) профессионального языка диспетчера [1].

СИСТЕМА МИМИР И ЕЕ РАЗВИТИЕ

В инструментальной экспертной системе МИМИР [1] развита концепция ограниченного естественного языка и моделей рассуждений. На основе МИМИР был разработан и практически

ИНТЕЛЛЕКТУАЛЬНЫЙ АГЕНТ

В компьютерной науке интеллектуальный агент — это программа, самостоятельно выполняющая задание, указанное пользователем компьютера, в течение длительных промежутков времени. Интеллектуальные агенты используются для содействия оператору или сбора информации. Одним из примеров заданий, выполняемых агентами, может служить задача постоянного поиска и сбора необходимой информации в Интернете. Компьютерные вирусы, боты, поисковые роботы — все это также можно отнести к интеллектуальным агентам. Такие агенты, как и любые прочие, имеют сложный, зачастую реализуемый нейросетями алгоритм, как, к примеру, у поисковой системы Google (экспериментальный поиск по видео). «Интеллектуальность» в этом контексте понимается как возможность обратной связи в соответствии, например, с результатами анализа поисковых запросов и их выдачей.

В искусственном интеллекте под термином интеллектуальный агент понимаются сущности, получающие информацию через систему сенсоров о состоянии управляемых ими процессов и осуществляющие влияние на них через систему актуаторов, при этом их реакция рациональна в том смысле, что процессы, выполняемые ими, содействуют достижению определенных параметров.

применен ряд задач-приложений (в частности, проработка заявок на ремонт оборудования и планирования ремонтов и др.

[4–9]). Вопрос на ограниченном естественном языке стал основным оператором нового языка вопросного программирования. Но все же МИМИР использовался как язык программирования, и МИМИР-программы разрабатывались программистами. В МИМИР-программах нужно было применять характерные методы программирования (резервирование переменных, организация подпрограмм, компиляция программ и др.). Надежды авторов системы МИМИР на применение этого языка самими диспетчерами-эксплуатационниками не оправдались. В статье Ю.Я. Любарского и соавторов [10] рассматривались проблемы создания интеллекту-

ЭКСТРЕМАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Экстремальное программирование (англ. Extreme Programming, XP) — одна из гибких методологий разработки программного обеспечения. Авторы методологии — Кент Бек, Уорд Каннингем, Мартин Фаулер.

Название методологии исходит из идеи применить полезные традиционные методы и практики разработки программного обеспечения, подняв их на новый «экстремальный» уровень. Так, например, практика выполнения ревизии кода, заключающаяся в проверке одним программистом кода, написанного другим программистом, в «экстремальном» варианте представляет собой «парное программирование», когда один программист занимается написанием кода, а его напарник в это же время непрерывно просматривает только что написанный код.

альных агентов на базе вопросного программирования. В частности, предлагалось привлечение к разработке не только инженера по знаниям, но и эксперта-посредника. Все же надо признать, что эта разработка остается весьма трудоемкой.

Отсюда следует, что необходимо, сохраняя естественно-языковые МИМИР-вопросы, отказаться от их включения в традиционные программные формы.

Конечная цель такой «реформы» — исключение из процесса разработки интеллектуальных агентов как профессиональных программистов, так и инженеров по знаниям. Единственным фактором при таком подходе оказывается технолог-эксплуатационник (заказчик), передающий свой опыт интеллектуальному агенту.

ЭКСТРЕМАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Сформулированному выше подходу соответствует новое направление в организации создания программ, называемое экстремальным программированием (XP) [11]. Главным инструментом XP является пользовательская история — небольшое и удобное в работе представление информации. Подобные истории сформулированы на повседневном языке пользователя и содержат небольшие детали, таким образом, оставаясь открытыми для интерпретации. Они помогают читателю понимать, что должна делать система. В XP предполагается, что завершенная пользовательская история представляет собой работающий код. Постоянное тестирование историй — неременное требование XP. При обычном использовании XP пользовательские истории создаются совместно разработчиками

и представителем клиента при доминировании клиента. Наша цель — исключить из процесса представителя разработчика (программиста). При этом программы-рассуждения можно называть просто рассуждениями (или моделями рассуждений). Разумеется, это идеализация, поскольку участие программистов в разработке системы целесообразно только в качестве консультантов.

МИМИР-вопросы «упаковываются» в пользовательскую историю с помощью фраз, генерируемых формальной грамматикой [12].

ПРИМЕР ПОЛЬЗОВАТЕЛЬСКОЙ ИСТОРИИ

Допустим, надо создать пользовательскую историю для ситуации «Срабатывание защиты», чтобы определить сработавшие защиты оборудования (ЛЭП, трансформаторов, систем шин). Далее для каждой сработавшей защиты необходимо найти выключатели, которые должны отключаться этой защитой при срабатывании. Если хотя бы один из выключателей не отключился, следует написать имя соответствующей защиты (с префиксом «срабатывание») и имя не отключившегося выключателя (с префиксом «нет отключения»). Далее нужно определить, сработало ли устройство УРОВ этого выключателя, и, если да, написать «Работа УРОВ».

Теперь допустим, что База знаний разработана, там содержатся топологические модели электрической и релейной защит [3]. При заполнении Базы знаний определены семантические группы понятий, связи между ними. Имена семантических групп и (при

ПРИМЕР ПОЛЬЗОВАТЕЛЬСКОЙ ИСТОРИИ

```

для КАЖДОГО ЭЛЕМЕНТА *1
{«РЗА объект, события срабатывание»
КРОМЕ
  «РЗА УРОВ»
КРОМЕ
  «РЗА АПВ»}
ОТОБРАЗИТЬ
«срабатывание РЗА»
ОТОБРАЗИТЬ
ИМЯ_РЗА *1
для КАЖДОГО ЭЛЕМЕНТА *2
{выкл., вкл., воздействие РЗА *1}
ОТОБРАЗИТЬ
«нет отключения»
ОТОБРАЗИТЬ
ИМЯ_ВЫКЛ. *2
для КАЖДОГО ЭЛЕМЕНТА *3
{«РЗА УРОВ выкл. *2, события срабатывание»}
ОТОБРАЗИТЬ
«работа УРОВ»
ВОЗВРАТ

```

Листинг 1

необходимости) отдельных их элементов составляют Словарь прикладной интеллектуальной системы. В результате система может отвечать на естественно-языковые вопросы, составленные на основе Словаря.

Тогда требуемая пользовательская история может иметь вид, представленный в листинге 1.

Строки, обрабатываемые МИМИР-процессором, записываются строчными буквами, строки «внешних» грамматических конструкций — заглавными.

Следует заметить, что описания получают достаточно краткими (это требование обязательно в методике XP).

В качестве пояснения рассмотрим, как находятся сработавшие защиты оборудования (ЛЭП, трансформаторов, систем шин). Это можно сделать, применив дизъюнктивную конструкцию:

«оборудование ЛЭП»

ИЛИ
«оборудование трансформаторы»
ИЛИ
«оборудование СШ» \$ 1
для КАЖДОГО ЭЛЕМЕНТА *1
«РЗА оборудование \$ 1, события срабатывание»
...

Более экономное решение дает «логическое вычитание» (оператор КРОМЕ) из всех защит энергообъектов — РЗА вида УРОВ и затем РЗА вида АПВ.

Для ввода и выполнения в МИМИР-системах пользовательских историй (ПИ) приведенного выше вида потребуется относительно простое программное обеспечение, состоящее из двух основных элементов:

- ПИ — редактор, «понимающий» формальную грамматику;
- МИМИР-исполнитель.

Работа исполнителя может быть организована как в компилятивном режиме, так и в режиме интерпретатора. Второй вариант

(интерпретация) предпочтительнее, так как обеспечивает простое и быстрое тестирование.

Допустим, что приведенное выше описание применяется в тестовом режиме: короткое замыкание на одной из линий 110 кВ (ВЛ 110_кВ Волконка — Головинка), отказ одного из линейных выключателей с последующей работой УРОВ.

В итоге получим:

Срабатывание РЗА
РЗЛ Волконка ВЛ 110 кВ Волконка — Головинка
Нет отключения
Волконка В-110 ВЛ Головинка
Работа УРОВ
Срабатывание РЗА
РЗЛ Головинка ВЛ 110 кВ Волконка — Головинка

Каждая завершенная пользовательская история получает уникальное имя, включаемое в Словарь системы, так что возможны вложения — вызов одних рассуждений из других. Так может обеспечиваться интеграция ПИ в систему.

Практически вместо записи ПИ выражениями вида (1) для пользователей может оказаться удобнее заполнение (с помощью соответствующего редактора) «блочной» структуры (рис. 1 на с. 44).

ЗАКЛЮЧЕНИЕ

Изложенная концепция сочетания интеллектуальных естественно-языковых систем с методикой экстремального программирования открывает возможность относительно несложной и эффективной разработки интеллектуальных агентов при минимизации участия в этом процессе профессиональных программистов.

ЛИТЕРАТУРА

1. Любарский Ю.Я. Интеллектуальные информационные системы. М.: Наука, 1990.
2. Любарский Ю.Я. Интеллектуальные модели рассуждений в советчике диспетчера электрических сетей // Электрические станции. 2017. № 3. С. 35–39.
3. Любарский Ю.Я., Хренников А.Ю., Александров Н.М. Нештатные ситуации в электрических сетях. Оперативный диспетчерский анализ на основе технологии экспертных систем // Новости электротехники. 2017. № 5. С. 36–38.
4. Любарский Ю.Я., Портной М.Г., Рабинович Р.С., Левиуш А.И., Князева Н.П. Экспертная система оперативного рассмотрения ремонтных заявок для АСДУ энергообъединений // Электричество, 1991, № 2.
5. Типовая инструкция по ликвидации нарушений в работе распределительных электрических сетей 0,38–20 кВ с воздушными линиями электропередачи. ТИ 34-70-058-86.
6. Власюк В.Д., Каковский С.К., Любарский Ю.Я. Автоматизация поиска повреждений в распределительных электрических сетях // Электрические станции. 2015. № 4. С. 29–36.
7. Любарский Ю.Я. Интеллектуальные модели рассуждений в советчике диспетчера электрических сетей // Электрические станции. 2017. № 3. С. 35–39.
8. Анашкин С.В., Каковский С.К., Карташев С.В., Любарский Ю.Я., Мирошкин А.Г. Тренажер для диспетчерского персонала электрических сетей // Электрические станции. 2014. № 2. С. 2–6.
9. Авагимова Ю.С., Дьячков В.А., Любарский Ю.Я., Рубцова Е.В. Принципы построения автоматизированной системы годового планирования ремонтов электросетевого оборудования // Электричество. 2009. № 3.
10. Любарский Ю.Я., Александров Н.М., Хренников А.Ю. Интеллектуальная АСДУ. Особенности мультиагентной архитектуры // Новости электротехники. 2017. № 5 (107) — 6 (108). С. 36–38.
11. Кон М. Пользовательские истории. Гибкая разработка программного обеспечения. М.: Вильямс, 2012.
12. Гладкий А.В. Формальные грамматики и языки. М., 1973.
13. Любарский Ю.Я., Моржин Ю.И., Александров Н.М., Радин П.С. Возможность получения оперативной справки об аварии в энергосистеме // Энергия единой сети. 201. № 3. С. 26–30.

БЛОЧНАЯ СТРУКТУРА ДЛЯ ЗАДАНИЯ ПОЛЬЗОВАТЕЛЬСКОЙ ИСТОРИИ

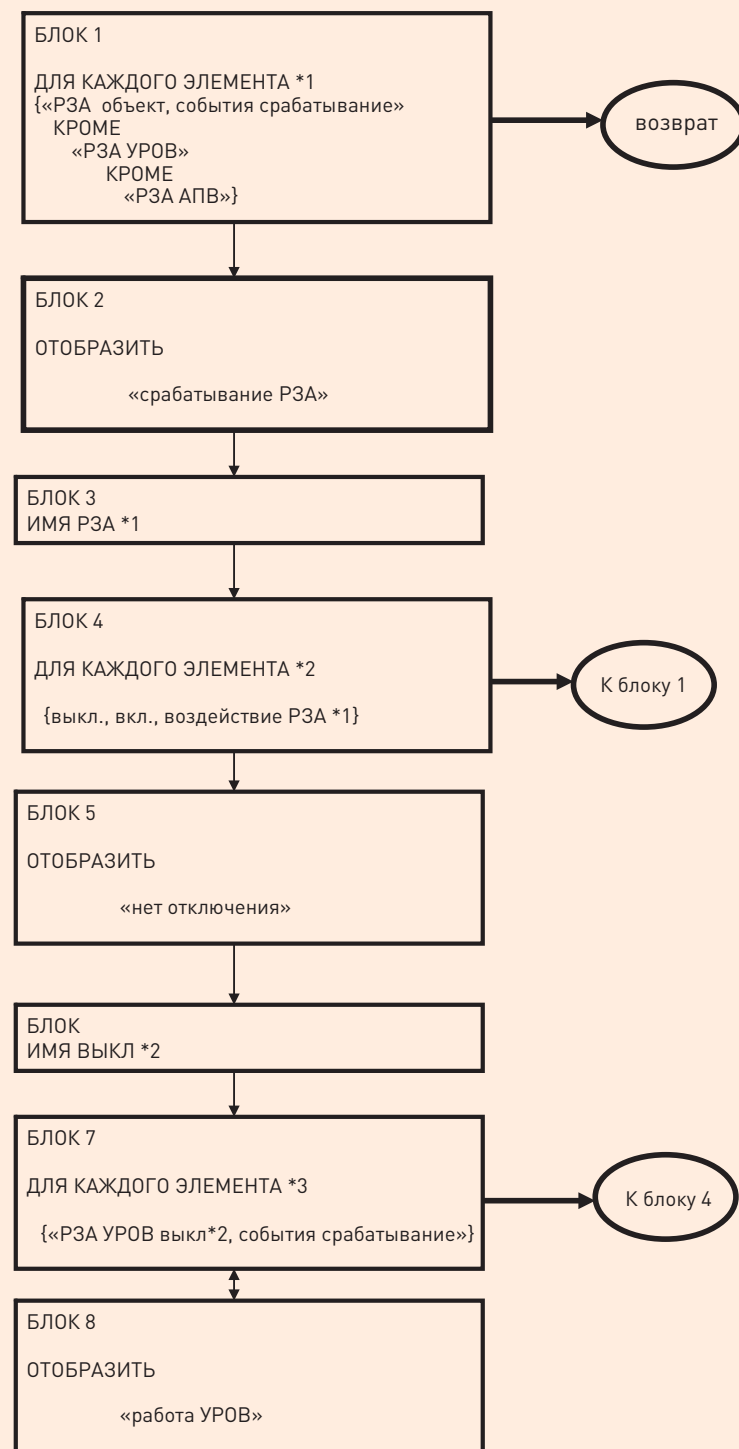


Рис. 1

ноябрь 2018



VIII
Открытый
шахматный турнир
энергетиков
памяти М. М. Ботвинника

2018

ШАХМАТНЫЙ ТУРНИР ЭНЕРГЕТИКОВ

Приглашаем команды энергетиков поддержать нашу добрую традицию и принять участие в ежегодном открытом шахматном турнире!

Состоится личное и командное первенство по правилам ФИДЕ для быстрых шахмат.

НАБИРАЙТЕ ЧЕТЫРЕХ ИГРОКОВ
И РЕГИСТРИРУЙТЕ КОМАНДУ
НА САЙТЕ ТУРНИРА
WWW.TURNIR.NTC-POWER.RU

