

УДК 681.3

Рыбина Г.В.

Московский государственный инженерно-физический институт
(технический университет), кафедра кибернетики

Интегрированные экспертные системы: подходы и принципы глубинной интеграции компонентов

In this paper discussed some questions of design of real-time integrated expert systems for diagnostics and complex technical system control. The models and methods of integration traditional expert systems with simulation modeling systems are described. Software systems «Rational Rose Real Time 6.0» (Rational Software Corp.) and G2 (Gensym Corp.) are used in experiments.

В статье обсуждаются некоторые вопросы построения интегрированных экспертных систем в реальном времени для диагностики и комплексного технического и системного контроля.

Введение

Функционирование современных предприятий в условиях постоянно возрастающей сложности решаемых задач и быстрой смены предъявленных к ним требований во многом определяется возможностями совместного использования новейших информационных и коммуникационных технологий, таких, как сети Internet и Intranet, мощных методов системного анализа и искусственного интеллекта (ИИ), математических и логико-лингвистических моделей и методов исследования, современных методологий проектирования, а также инструментальных средств (ИС) для их реализации.

Следует отметить, что популярность и практическая значимость исследований в ИИ, в частности в области экспертных систем (ЭС), долгое время приносили пользу другим областям науки, особенно таким *пограничным* направлениям, как лингвистика, психология, логика, программирование, медицина, химия, экономика и другим средне- и слабоструктурированным предметным областям (ПО) с богатой эмпирикой, размытой теорией, скрытыми взаимосвязями, с большим количеством *белых пятен*. Все это послужило хорошей основой для создания некоторой *среды* ИИ, готовой к выработке удобных методов и подходов к решению практически любой проблемы за счет глубокого анализа явлений и процессов, происходящих в конкретной ПО.

Накопление опыта и создание технологических приемов в проектировании ЭС для широкого класса задач *открыли дверь* для многих методов и технологий, которые ранее не были приемлемы для ИИ, например классическая математическая статистика, стохастические подходы, модели распознавания, имитационное моделирование и т.п. Возникла новая проблема, связанная с объединением разных подходов, методов и технологий в согласованную и

оптимальную форму с целью использования средств каждой технологии максимальным образом для увеличения точности и качества решения задач.

Таким образом, доминирующие в последнее десятилетие тенденции к интеграции исследований в различных областях привели к необходимости совмещения семантически разнородных объектов, моделей, методологий, концепций и технологий, что неизбежно породило принципиально новые классы систем, такие, как интегрированные интеллектуальные системы, интегрированные экспертные системы, интегрированные информационные системы, интегрированные производственные системы и предприятия и др.

Следовательно, можно констатировать, что сегодня в ИИ происходят два основных процесса – расширение арсенала ИС за счет добавления новых инструментов из других областей знаний (*интеграционный процесс*) и извлечение новых подобластей из хорошо известных (*разделительный процесс*).

Примером *разделительного процесса* является направление, связанное с извлечением знаний из БД (Knowledge Discovery и Data Mining). Добавим, что, скорее всего, в дальнейшем процессы кристаллизации и разделения новых областей сформируют и третью тенденцию.

В фокусе внимания данной работы находятся *интеграционные процессы*, связанные со статическими и динамическими ЭС; в частности, обсуждаются некоторые результаты исследований и разработок по созданию теории и технологии построения интегрированных экспертных систем (ИЭС) для *статических* ПО, полученные автором в рамках выполнения исследовательского проекта АТ-ТЕХНОЛОГИЯ [1-3, 5 и др.].

Однако наибольшее внимание уделяется вопросам создания *динамических* ИЭС, функционирующих в реальном времени (ИЭС РВ), в которых интеграционные проблемы проявляются наиболее ярко.

За последние годы автором накоплен определенный опыт разработки ИЭС РВ на основе инструментальной системы G2 (Gensym Corp.) для задач диагностики и управления [1, 2], таких, как: управление современным электрофизическим комплексом, диагностика сложных технических систем, контроль предстартовой подготовки ракетносителей, радиоэкологический мониторинг территорий, непосредственно прилегающих к атомным электростанциям, экологический мониторинг атмосферного воздуха, прогнозирование радиационных повреждений бортовых микро-ЭВМ и др.

Несмотря на внешнее различие проблемных областей (ПО), предметом исследований являлись сложные технические системы (СТС) дискретного и непрерывно-дискретного типа, для которых использование ИЭС РВ обеспечивает поддержку решения следующих основных задач [1]:

- динамическое моделирование процессов функционирования СТС;
- контроль работы СТС, регистрация отклонений от заданного режима, сигнализация о предаварийных ситуациях и аварийное отключение и др.;
- изучение действий операторов по управлению СТС и обучение персонала;
- удобный графический интерфейс пользователя для наблюдения за изменениями основных параметров СТС и др.

В связи с этим существенно изменяется архитектура ИЭС, предназначенной для решения задач в реальном времени (РВ), т.к. модифицируются практически все базовые компоненты статической ИЭС и добавляются две новые подсистемы (моделирования внешнего мира и сопряжения с реальным оборудованием), а также значительно изменяется технология создания ИЭС РВ, которая определяется используемыми ИС. Однако в целом теория, методология и технология создания ИЭС РВ не вышли из стадии экспериментальных исследований, поэтому данная проблема является весьма актуальной.

В центре внимания данной работы находятся вопросы, связанные с построением одного из важнейших компонентов ИЭС РВ – подсистемы моделирования внешнего мира.

1. Постановка задачи

Вопросы моделирования внешнего мира и различных его состояний в РВ являются кардинальными при построении любой ИЭС РВ. Наиболее востребованными здесь оказались методы и средства *имитационного моделирования* (ИМ), т.к. размерность решаемых задач и неформализуемость СТС, для которых разрабатываются ИЭС РВ, не позволяют использовать строгие математические методы.

С точки зрения взаимопроникновения интеграционных процессов позиции традиционных ЭС и ИМ достаточно близки, поскольку и те и другие основаны на использовании моделей (математических – в случае ИМ и экспертных – в случае ЭС), носящих равноправный характер, объединение которых может обеспечить получение нового качества принимаемых решений и общего результата в целом. Однако проблемы совместного использования ЭС и ИМ решаются только с узких, частных позиций разработки конкретного приложения, не затрагивая проблемы интеграции в общем виде.

В исследованиях автора [3] эти вопросы рассматриваются и решаются в рамках универсальной задачно-ориентированной методологии (ЗОМ), предназначенной для использования в статических и динамических приложениях и представляющей собой совокупность моделей, методов, алгоритмов и процедур для создания прикладных ИЭС.

Проблему интеграции в ЗОМ предлагается рассматривать с точки зрения трех уровней интеграции, для каждого из которых была предложена классификация ИЭС, введены понятия ИЭС с поверхностной и глубинной интеграцией компонентов и показано, что методологии разработки простых ЭС могут применяться только для создания ИЭС с поверхностной интеграцией и совершенно неприменимы для ИЭС с глубинной интеграцией.

В этом случае предлагается применить подход, связанный с усовершенствованием ЭС путем включения нетрадиционных для них функций некоторого компонента N (где N – это СУБД, ППП, система ИМ и т.д.), что является важной концептуальной основой ЗОМ.

В данном случае наибольший интерес представляют вопросы интеграции ЭС с ИМ в рамках ИЭС с глубинной интеграцией компонентов. Модульный принцип построения ИЭС на основе ЗОМ, а также имеющееся сходство между отдельными понятиями в ЭС и ИМ обеспечивают возможность интеграции этих технологий. Кроме того, сложность выбранных СТС делает невозможным применение строгих математических моделей для их описания, поэтому основной путь решения проблемы – построение имитационной модели ($M^{ИМ}$), причем предпочтение было отдано методам интеллектуального ИМ [4].

Рассмотрим предложенный подход к построению имитационных моделей СТС на примере моделирования в среде G2 системы жизнеобеспечения электрофизического комплекса (ЭФК), в данном случае – ускорителя заряженных частиц, являющегося типичным представителем СТС [5].

2. Имитационные модели системы жизнеобеспечения ЭФК

Для упрощения дальнейшего описания структуры $M^{ИМ}$ ограничим количество подсистем в системе жизнеобеспечения одной подсистемой – обеспечения вакуума. С точки зрения ИМ, подсистемы жизнеобеспечения ЭФК представляют собой дискретно-непрерывные системы, теоретико-множественная модель которых, в общем случае, выглядит следующим образом:

$$M_{СЖ}^{ИМ} = \langle M_{ОУ}, M_P, M_{СВ}, V^X, V^U, V^E, V^Y, V^Z, S, F^Y, \rightarrow^u, F^{xeu} \rightarrow^{yz} \rangle$$

где $M_{ОУ}$ – модель объекта управления (ОУ); M_P – модель регулятора; $M_{СВ}$ – модель случайных воздействий; $V^X = \{v_i^x\}, i = (1, m)$ – множество контролируемых неуправляемых входов; $V^U = \{v_j^u\}, j = (1, s)$ – множество контролируемых управляемых входов $M_{ОУ}$; $V^E = \{v_h^e\}, h = (1, k)$ – множество случайных возмущений; $V^Y = \{v_l^y\}, l = (1, r)$ – множество выходных параметров $M_{ОУ}$ (используется в регуляторе); $V^Z = \{v_g^z\}, g = (1, q)$ – множество выходных параметров $M_{ОУ}$; $S = (s_c), c = (1, n)$ – множество возможных (допустимых и нештатных) состояний; $F^Y \rightarrow^u$ – функция генерации управляющего вектора $u(t_{i+1})$ на основе поступившего выходного вектора $y(t_i)$; $F^{xeu} \rightarrow^{yz}$ – функция отображения входа ОУ в его выход.

Если обозначить через I вход $M_{ОУ}(V^X, V^U, V^E)$, а через O выход (V^Y, V^Z) , то $o(t) = F^{xeu} \rightarrow^{yz}(i(\tau), \forall \tau \in [v, t], o^{(k)}(v), k = 0; (n-1), t)$,

где $[o^{(k)}(t)](t = v) = o^{(k)}(v), k = 0; (n-1)$ – начальные условия (НУ). Таким образом, выход в любой момент времени t – есть некоторая функция и от входа, и от НУ. Выход $M_{ОУ}$ имеет размерность $(r + q) \leq n$; таким образом, $O \subseteq S$, а каждое конкретное состояние ОУ описывается множеством выбранных (по разным критериям) свойств: $C^- = \{c_1^-, \dots, c_\lambda^-\}$, где C^- – означенные свойства ОУ.

Таким образом, через множество C^{\sim} можно описать множество состояний $S \subseteq S^D \subseteq S^{HSH}$, где S^D – множество допустимых, а S^{HSH} – множество нештатных состояний.

Такое формализованное представление M^{IM} описывает работу системы в целом, однако остается достаточно абстрактным для дальнейшей реализации. Поэтому для детализации, а также универсальности полученной модели, может быть использовано, например, CASE-средство «Rational Rose RealTime 6.0» (Rational Software Corp.), в котором поддерживается объектно-ориентированная парадигма анализа и проектирования и в качестве языка построения модели используется UML с расширениями для поддержки разработки систем РВ (однако это не является обязательным, и дальнейшее проектирование может осуществляться в среде G2).

В этом случае все элементы оборудования, составляющие M_{OY} и M_P , будут представлены в виде *капсул* (абстрактного представления понятий реального мира, т.е. оборудования ЭФК) с необходимым набором *портов* (абстрактного представления каналов передачи данных реального объекта), через которые приходят сообщения от других элементов (капсул в абстрактном понимании, оборудования в реальном мире).

Так как $O \subseteq S$, то выход (V^y, V^z) M_{OY} полностью описывается атрибутами капсул, представляющих M_{OY} , часть входа (V^u) описывается атрибутами капсул, представляющих M_P , часть (V^e) – капсулами компонента, представляющего собой генератор случайных воздействий, а каждая «стрелка» (\rightarrow) представляется каналом передачи сообщений (информации) – протоколом.

Для каждой капсулы создается диаграмма состояний и переходов, совокупность которых для всех капсул раскрывают (представляют) функции $F^y \rightarrow^u$, $F^{xeu} \rightarrow^{yz}$. Получаем следующее представление M^{IM} в терминах UML-RT:

$$M_{СЖ}^{IM} = \langle C, P, S, T, E, R_c, R_p, A \rangle,$$

где C – множество капсул; P – множество портов; S – множество состояний капсул; T – множество переходов; E – множество событий, наступление которых инициирует переход к другому состоянию; R_c – множество отношений между капсулами; R_p – множество отношений между протоколами; A – механизм отслеживания событий и инициирования соответствующих переходов.

Полученное достаточно универсальное представление M^{IM} позволяет перейти к ее реализации на инструментальном уровне, в данном случае в среде G2, в которой описанная модель не претерпевает существенных изменений: диаграмма классов, разработанная в Rational Rose RealTime, становится аналогичной диаграммой в среде G2; иерархия протоколов переходит в иерархию соединений (connections) и отношений (relations); логика диаграммы переходов и состояний описывается правилами конструкции *whenever*, которые дают возможность реализовать механизм *отслеживания событий*.

Под *событием* понимается нахождение системы в априорно известном состоянии: переменная, параметр или атрибут объекта получает новое значение; при присвоении значения переменной произошла ошибка; переменная

утратила свое значение (значение больше не актуально); создан объект какого-либо класса; объект передвинут (изменил координаты) на рабочем пространстве; объект перешел в активное или неактивное состояние; два объекта стали находиться в определенном отношении; два объекта стали находиться в любом отношении; два объекта были соединены между собой.

Следовательно, M^{IM} в среде G2 можно представить такой совокупностью:

$$M_{СЖ}^{IM} = \langle CL, O, C, E, RL_e, R_{CI}, R_o \rangle,$$

где CL – множество классов системы, O – множество объектов, C – множество соединений между объектами, E – множество модельных событий, RL_E – множество правил машины отслеживания событий, R_{CI} – множество отношений между классами, R_o – множество отношений между объектами.

3. Методика построения имитационных моделей СТС

Таким образом, в рамках ЗОМ была создана следующая конкретная методика построения M^{IM} СТС, ориентированная на использование в среде G2:

1. Проводится анализ ПО, выделяются основные понятия ПО, их характеристики и операции их функционирования.

2. В G2 описываются абстракции данных понятий в виде классов. Характеристики понятий СТС представляются с помощью атрибутов, а операции с помощью методов.

3. С помощью визуального редактора строится схема оборудования реальной СТС на основе программных объектов, экземпляров описанных классов. При этом взаимосвязи между объектами реальной СТС задаются с помощью соединений и отношений.

4. В итоге получается $S = \langle O, R \rangle$, где O – множество объектов схемы, R – множество отношений между ними.

5. Далее строится множество модельных и временных событий – $E = \{e_i\}$. Модельное событие – априорно заданное состояние, т.е. набор означенных атрибутов одного или более объектов. Временное событие – событие с априорно заданным модельным или реальным временем.

6. Строится множество действий $D = \{d_j\}$, связанных с множеством событий E , (действие – это совокупность методов объектов схемы S) и планировщик их совместного функционирования (последовательно, параллельно, с временной задержкой). Длительность действий моделируется в G2 конструкциями типа *wait for t*, где t – время задержки.

7. На основании п.п. 4-5 строится *машина отслеживания событий*, основными функциями которой является сканирование состояний системы, а при переходе системы в одно из состояний, для которого описано событие в множестве E , – инициирование действия, связанного с данным событием. Этот механизм в среде G2 реализуется с помощью правил типа *whenever*.

8. Определяется множество начальных состояний S_{begin} . Инициализация схемы S одним из начальных состояний происходит путем выполнения конструкций типа *initially*.

4. Модель глубинной интеграции компонентов ИЭС РВ

Модель интеграции основных компонентов действующего прототипа ИЭС РВ для управления ЭФК изображена на рис.1, где представлены следующие компоненты: БЗ — база знаний прототипа ИЭС РВ; РП — рабочая память; ПМВМ — подсистема моделирования внешнего мира; БМФ — библиотека математических функций; R1 — множество правил, описывающих экспертные знания; R2 — множество правил отслеживания событий в M^{IM} ; А — множество действий в M^{IM} ; F1 — множество функций, используемых при решении задачи; F2 — множество функций, используемых при выполнении действий А; ГИС — инструментальная геоинформационная система (здесь может быть любая другая система в зависимости от специфики ИЭС РВ).

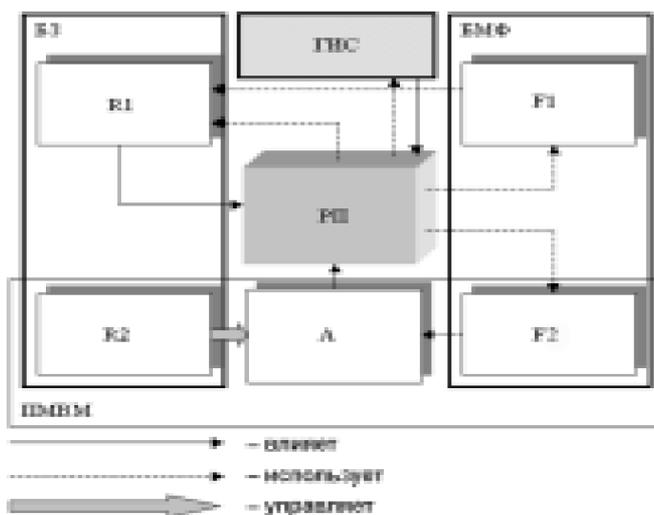


Рис.1. Модель интеграции основных компонентов ИЭС РВ

Данную модель можно пояснить следующим образом. В РП представлены выделенные объекты ПО. Совокупности атрибутов данных объектов описывают состояния системы. Правила типа R1 описывают экспертные знания, необходимые для решения поставленной задачи. В результате вывода на правилах происходит изменение состояния системы, т.е., в общем случае, меняются атрибуты объектов РП в соответствии с целью решаемой задачи. При этом процесс вывода решения на правилах типа R1 использует свойства объектов РП, а также функций типа F1 в БМФ. Функции типа F1 оперируют в процессе своей работы свойствами объектов РП.

Для имитации процессов внешнего мира используется *машина отслеживания событий*, представленных с помощью правил типа R2, которые сканируют модельные (априорно заданные состояния системы) и временные события и вызывают действия A. Действия A имеют заданную временную протяженность, а результатом их выполнения являются новые состояния системы. При выполнении действий A используются, в общем случае, математические функции типа F2 из БМФ. Функции типа F2 оперируют объектами РП. Кроме этого, в приведенном примере для работы с картографической информацией используется ГИС, с помощью которой осуществляется ввод изменяющихся данных в РП, причем эти данные должны быть синхронизированы.

Заключение

Таким образом, $M^{ИМ}$ любой СТС в среде G2 — это объектная модель, т.е. множество взаимосвязанных как каналами передачи данных, так и логическими схемами программных объектов, имитирующих динамику поведения реальной СТС, описанной с помощью этих объектов. Собственно процесс имитации в реальном времени поддерживается планировщиком G2, который координирует процессы обработки модельных и временных событий, что существенно облегчает разработку $M^{ИМ}$.

Следует отметить также, что при необходимости для моделирования сложных непрерывных процессов, описываемых, в частности, дифференциальными уравнениями второго или большего порядка, может быть дополнительно использована система MATLAB или аналогичные системы, а интеграция с G2, будет производиться на основе средств интерфейса GSI (G2 Standard Interface).

Исследования выполнялись в лаборатории «Системы искусственного интеллекта» кафедры кибернетики МИФИ при поддержке РФФИ РАН (грант №00-01-00679).

Литература

1. Рыбина Г.В. Особенности и принципы построения интегрированных экспертных систем для диагностики сложных технических систем // Приборы и системы управления. – 1998. – №9. – С.12-16.
2. Рыбина Г.В., Рыбин В.М. Динамические интегрированные экспертные системы реального времени: анализ опыта исследований и разработок // Приборы и системы управления. – 1999. – №8. – С.4-8.
3. Рыбина Г.В. Задачно-ориентированная методология автоматизированного построения интегрированных экспертных систем для статических проблемных областей // Известия РАН. Теория и системы управления. – 1997. №5. – С.129-137.
4. Емельянов В.В., Ясиновский С.И. Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов. // Языки РДО. – М.: АНВИК, 1998. – С. 427.
5. Рыбина Г. В. Использование методов имитационного моделирования при создании интегрированных экспертных систем реального времени // Известия РАН. Теория и системы управления. – 2000. – №5.