

Инновационный подход к разработке программного обеспечения систем боевого управления

к.т.н. **В. Н. Волобуев**, к.т.н. **А. Ф. Гаврилов**, **А. В. Калачинский**

С появлением первых систем боевого управления встал вопрос о создании надежного программного обеспечения (ПО). Традиционно сложившийся подход к разработке ПО включает в себя несколько основных стадий, связанных между собой лишь через программную документацию (рис. 1).

Опыт проектирования, разработки, сдачи и сопровождения систем, созданных при участии авторов, показал, что существенным недостатком традиционного подхода является отсутствие наглядных способов проверки соответствия между описанием программного обеспечения и его фактической реализацией. Это влечет за собой усложнение процесса отладки ПО, его интеграции, а также сопровождения и авторского надзора. Также значительно усложнена совместная разработка отдельных задач и преемственность исходного кода программ из-за высокой зависимости его от индивидуальных подходов исполнителей к реализации алгоритмов.

Основным подходом для устранения описанных недостатков, широко внедряемым авторами, является использование при разработке ПО автоматного подхода [1 – 3], в котором автоматы задаются графами переходов. Пример построения графа переходов автомата и описание используемых переменных показаны на рис. 2.

Экспериментальное внедрение автоматного подхода было осуществлено авторами при реализации протокола информационного взаимодействия в конце 2003 года. При этом в терминах автоматного программирования [4, 5] были описаны два алгоритма управления – алгоритм информационного взаимодействия с объектом и алгоритм управления объектом.

Опыт экспериментального использования предлагаемого подхода показал следующие его преимущества по сравнению с традиционным:

- быстрая реализация (два месяца от идеи внедрения до отладки на комплексном стенде);
- подробное технологическое документирование работы ПО (необходимое документирование автоматически добавляется в исходный текст программы средствами разработки);
- простота изменения поведения работы системы за счет изменения графа переходов автомата;
- быстрая интеграция (одна неделя);
- наглядность программной документации;
- однозначность соответствия программной документации и разработанного ПО.

Основным достоинством применения автоматного подхода является реальное соответствие структур алгоритмов и их программной реализации на всех стадиях создания ПО – от проектирования до испытаний.

Автомат	
Имя:	0
Название:	Организация связи и прием данных

x0	Питание включено
x1	Чтение данных прошло
x2	Получен признак отказа 1
x3	Связь с узлом установлена
x4	Команда смена основности
x5	Команда на ручное переключение резерв
x6	Выработан признак отказа 2
x7	Выработан признак неисправности связи
x8	Получен запрос отказа 2
x9	Установлена связь с предыдущим узлом
x10	Количество переключений < 2
x11	Количество потерь связи > 1
x12	Питание изделий включено
x13	Признак начала работы с объектом
xt1	Превышено время ожидания 10 сек.
xt0	Превышено время ожидания 60 сек.

z1	Отправить заявку на соединение
z2	Отправить заявку на отключение
z3	Снять признак смены основности
z4	Снять признак ручного переключения
z5	Установить признак связи с узлом
z6	Снять признак связи с узлом
z7	Соединиться с другим узлом
z8	Увеличить счетчик переключений
z9	Увеличить счетчик потерь связи
z10	Повторить выдачу структур
z11	Выдать команду «Авария»
z12	Привести данные в исходное состояние
z13	Снять команду «запрос отказа»
z14	Снять признак отказ 2
z15	Установить Признак завершения режима
zat0	Вкл. таймер ожидания вып-я операции
zdt0	Выкл. таймер ожидания вып-я операции
zam0	Выдать сообщ. 'Нет связи'
zdm0	Удалить сообщ. 'Нет связи'
zam1	Выдать сообщ. 'Прибор неисправен'
zam2	Выдать сообщ. 'Отказ 2'
zam3	Выдать сообщ. 'Связь неисправна'
zam4	Выдать сообщ. 'Отказ 1'
zdm	Удалить все сообщения

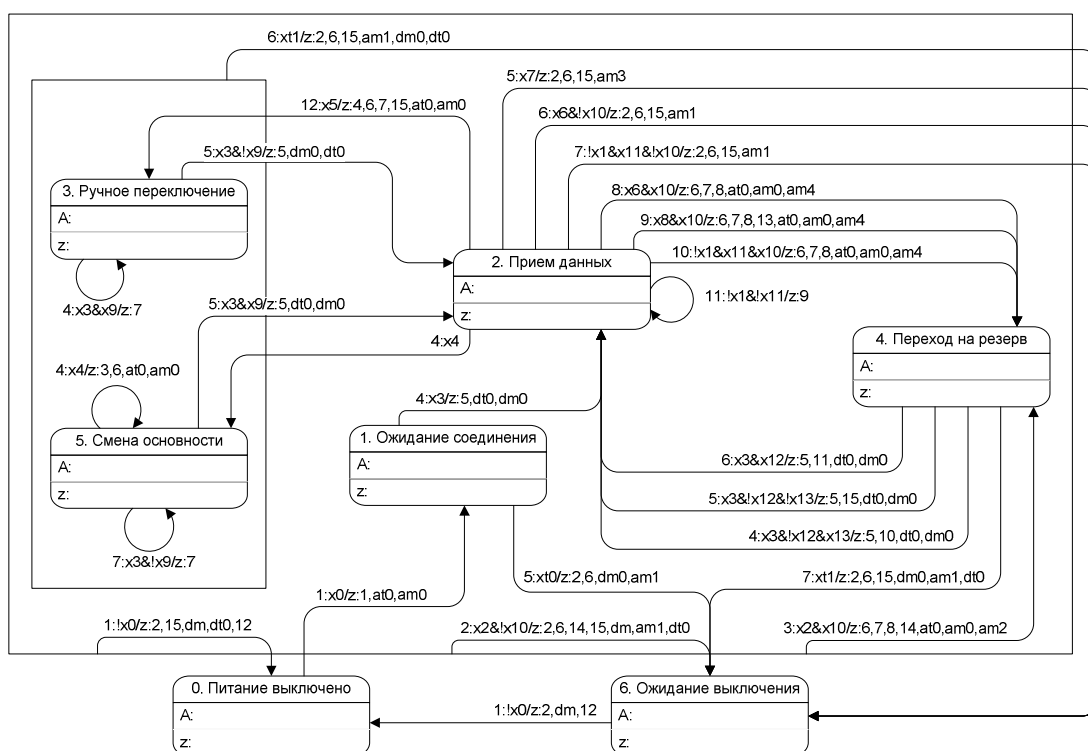


Рис. 2. Граф переходов автомата и описание используемых переменных

Применение автоматного подхода привело к необходимости использования, при разработке ПО, стадии проектирования автоматов реализации управления поведением системы (рис. 3).

Стадия проектирования автоматов необходима для создания на основе алгоритмов электронного документа, содержащего графы переходов автоматов, реализующих управление поведением системы. На основе этого документа с использованием специального программного обеспечения [6] производится автоматическое преобразование графа переходов в исходный текст ПО. Такая организация работы позволяет достичь однозначного соответствия между описанием ПО и его практической реализацией.

В настоящее время автоматное программирование развивается в трех основных направлениях: логическое управление, программирование с явным выделением состояний и объектно-ориентированное программирование с явным выделением состояний, создаются технологии и программные продукты по автоматизированному построению и верификации автоматных программ [7].

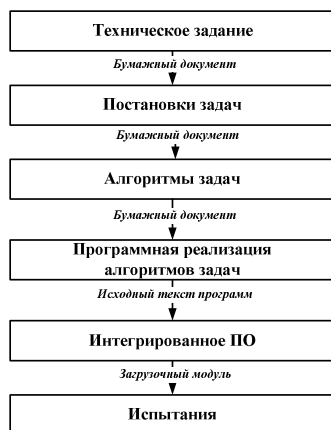


Рис. 1. Основные стадии разработки ПО при использовании традиционного подхода

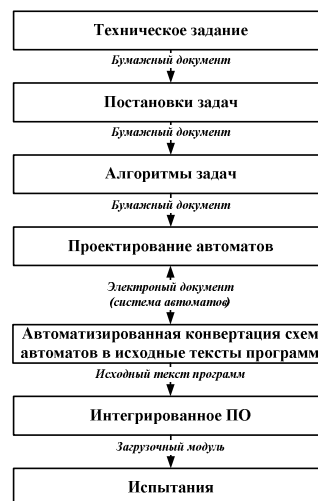


Рис. 3. Основные стадии разработки ПО с использованием автоматного подхода

Перечислим основные достоинства автоматного подхода при разработке ПО.

1. Использование графов переходов в качестве языка спецификаций алгоритмов делает обозримым весьма сложное поведение программы и позволяет легко вносить изменения как в спецификацию, так и в ее реализацию. На практике, автоматная программа часто работает правильно с **первого раза**, что является следствием самого подхода к проектированию, когда программный код является вторичным по отношению к наглядному графу переходов.

2. Подробное документирование проекта создания программного обеспечения позволяет при необходимости вносить в него изменения через длительный срок после выпуска, в том числе, специалистами, не участвовавшими в проектировании.

3. Программа, построенная на основе автоматного подхода, четко разделяется на две части – системонезависимую и системозависимую. При проектировании системонезависимой части программы детали реализации входных и выходных воздействий скрыты. Они раскрываются только при реализации системозависимой части (входные переменные и выходные воздействия). Стадии проектирования и реализации системонезависимой части программы полностью разделены.

4. Для кодирования любого числа состояний автомата используется только одна внутренняя переменная, что обеспечивает возможность наблюдения за поведением автомата путем "слежения" за изменениями значений только этой переменной.

5. Автоматическое ведение протокола в терминах спецификации обеспечивает возможность сертификации программы. При этом демонстрируется соответствие функционирования программы "поведению" системы графу переходов для рассматриваемых событий при выбранных значениях входных переменных.

По мнению [8], целесообразно использовать при создании ПО стадию проектирования графов автоматов. Для этого необходимо разработать:

- правила представления автоматов;
- методы верификации автоматных алгоритмов [9];
- программное обеспечение конвертации автоматов, разработанных на основе указанных выше правил, в исходные тексты программ;
- соответствующие корректировки к стандартам предприятия в части разработки ПО с использованием автоматного подхода.

Рассмотренный в статье инновационный подход к созданию ПО позволяет повысить качество его разработки, а также заметно снизить временные затраты на его отладку, интеграцию и сопровождение.

Литература

1. Шалыто А. А. SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб, Наука, 1998.
2. Шалыто А. А., Поликарпова Н. И. Автоматное программирование. СПб.: Питер. 2010.
3. Волобуев В. Н., Гаврилов А. Ф., Новоселов С. Е., Ремизов А. О. Особенности разработки сложных программных комплексов // Системы управления и обработки информации. 2009. Вып.17.
4. Шалыто А. А., Туккель Н. И. Автоматный подход к созданию программного обеспечения для систем логического управления и «реактивных» систем // Системы управления и обработки информации. 2000. Вып. 2.
5. Шалыто А. А., Туккель Н. И. Программирование с явным выделением состояний // Мир ПК. 2001. Вып. 8, 9.
6. Головешин А. Конвертор «Visio2Switch». <http://is.ifmo.ru/progeny/visio2switch>.
7. Научно-технический вестник СПбГУ ИТМО. Вып.53. Автоматное программирование. 2008.
8. Гаврилов А. Ф., Калачинский А. В., Новоселов С. Е., Михайлов В. О. Перспективы развития технологий разработки программного обеспечения АИУС // Системы управления и обработки информации. 2004. Вып.8.
9. Вельдер С. Э., Шалыто А. А. Методы верификации моделей автоматных программ // Научно-технический вестник СПбГУ ИТМО. Вып.53. Автоматное программирование. 2008.