

**ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ВНЕШНИХ СИСТЕМ  
ПРИ ПРОЕКТИРОВАНИИ КОМПЛЕКСОВ ВООРУЖЕНИЯ,  
ВОЕННОЙ И СПЕЦИАЛЬНОЙ ТЕХНИКИ**

**А.В. Журавлев, К.А. Аксенов (Екатеринбург)**

Развитие комплексов вооружения, военной и специальной техники невозможно без создания современных испытательных стендов для отработки архитектурных, алгоритмических и программно-аппаратных решений. Для обеспечения гибкости проектирования военной техники важно иметь возможность на ранних стадиях проверять правильность решений, закладываемых в конструкторскую и программную документацию программно-аппаратных комплексов. Возможность полноценной проверки работы изделия на стадии опытного образца или макета системы могут обеспечить испытательные стенды (комплексы) отработки аппаратуры и программ. Основой комплекса отработки аппаратуры и программ является имитационное оборудование, обеспечивающее моделирование работы внешних по отношению к объекту контроля систем [1, 2]. Имитационное оборудование позволяет проверять объект контроля, как в штатных режимах функционирования, так и с внесением различных сбоев и неисправностей для отработки алгоритмов резервирования (проверка принципа одной возможной неисправности) [2].

На сегодняшний день на предприятиях, занимающихся созданием вооружения, военной и специальной техники нет единых подходов и платформ для создания имитационного оборудования, в связи с трудоемкостью создания полноценного испытательного оборудования, поддерживающего проведения штатного режима работы и внесения имитаций, отработка на предприятиях может проходить по ускоренной схеме без полной проверки алгоритмов. Подобный подход в итоге приводит к выявлению замечаний на более поздних этапах создания систем, при стыковке систем производства разных предприятий на внешних позициях (испытательные полигоны головного предприятия, полигоны МО РФ). Устранение замечаний на поздних этапах является дорогостоящим, кроме того негативно сказывается на имидже предприятия-изготовителя. Одним из основных современных трендов и одной из важнейших задач при проектировании является смещение времени выявления замечаний на более ранние этапы создания систем [4].

Для решения задачи унификации программного обеспечения испытательного оборудования проведены исследования и получены результаты, характеризующиеся научной новизной:

- создана модель универсального имитирующего комплекса, обеспечивающего моделирование внешних систем в части информационного взаимодействия на испытательных стендах;
- разработан метод моделирования работы группы внешних систем на основе модели универсального имитирующего комплекса.

**Модель универсального имитирующего комплекса**

Универсальный имитирующий комплекс предназначен для моделирования внешних систем при отработке опытных образцов или макетов новых видов вооружения, военной и специальной техники на испытательных стендах в штатных режимах работы и с внесением неисправностей. Универсальный имитирующий комплекс должен обеспечивать взаимодействие по различным интерфейсам и иметь возможность подключения функциональных модулей для поддержки протокола и логики работы конкретной внешней системы [5].

Модель универсального имитирующего комплекса представлена на рисунке 1 и состоит из семи основных компонентов:

- подсистемы загрузки конфигурации;
- подсистемы управления;
- подсистемы представления задач;
- подсистемы контроля состояний;
- подсистемы синхронизации;
- подсистемы диагностики;
- подсистемы работы с интерфейсами.

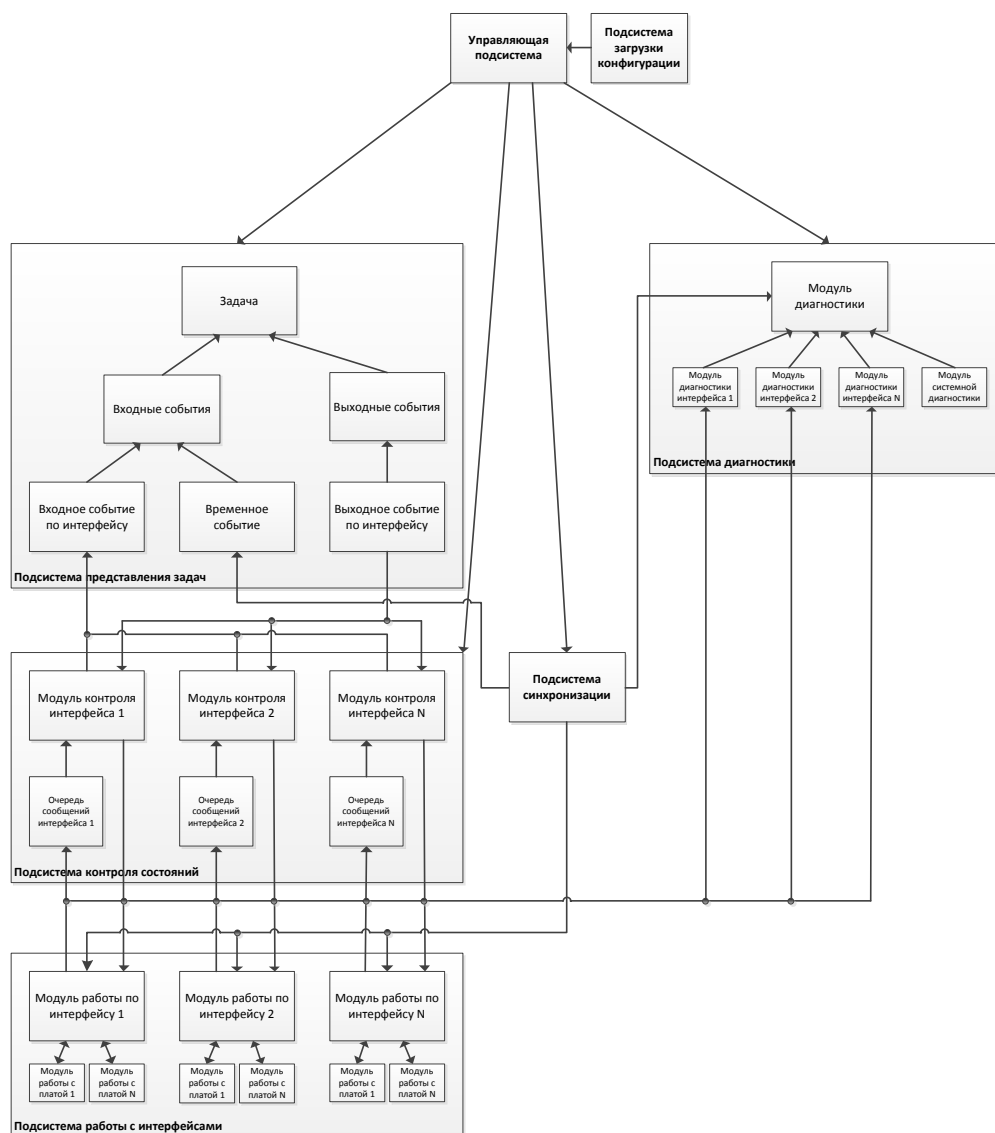


Рис.1 – Модель универсального имитирующего комплекса

**Подсистема загрузки конфигурации** предназначена для загрузки и представления информации в структурах данных имитатора. Подсистема загрузки принимает на вход два файла конфигурации, рекомендуется использовать две локальные базы данных формата sqlite. Выбран формат баз данных (БД) sqlite, так как представление и выборка данных осуществляется наиболее доступным и быстрым способом, также БД в данном формате легко представить в табличном виде для

---

пользователя. Файл описания аппаратной части (первый конфигурационный файл) имитирующего комплекса содержит описание используемых портов ввода – вывода, команд, используемых в ходе режима, а также формат представления сохраняемой диагностической информации. Файл исходных данных (второй конфигурационный файл) содержит описание циклограммы проводимого режима (задачи, события и команды). Файл описания аппаратной части, составляется для каждой имитируемой системы, а коррекция производится только в случае внесения изменений в кабельную сеть или коррекции исходных данных по взаимодействию систем. Конфигурационный файл режима создается для каждого режима. Для сопоставления файла описания аппаратной части с файлом исходных данных используется таблица команд, содержащаяся в обоих файлах.

**Подсистема управления** предназначена для загрузки и инициализации модулей системы. Данная подсистема производит управление всеми подсистемами на этапе запуска и останова программы.

**Подсистема представления задач** предназначена для представления задач имитируемой системы в виде виртуальных событий. Основной задачей подсистемы является абстрагирование аппаратной реализации событий. Схема модели события представлена на рисунке 2. Виртуальное событие (Event) содержит входные воздействия (ActionIn) и выходные воздействия (ActionOut). Виртуальное событие имеет обязательное временное входное воздействие (ActionInTime), используемое для активации события в заданное время. Входными воздействиями могут являться как шаблонные послышки по кодовым интерфейсам, так и релейные команды. Имеется возможность отправлять послышки по кодовым и релейным интерфейсам взаимодействия (выходные воздействия). Особенностью реализации событий является возможность обмена служебной информацией с другими событиями для реализации сложной логики работы системы. Для реализации интеллектуальных событий предусмотрена система функциональных модулей (плагинов), позволяющая определить функцию-обработчик события, входного или выходного воздействия. Система плагинов, позволяет подменить обработчик без перекомпиляции всей программы, нужно лишь указать файл с новым обработчиком в конфигурационном файле аппаратной части.

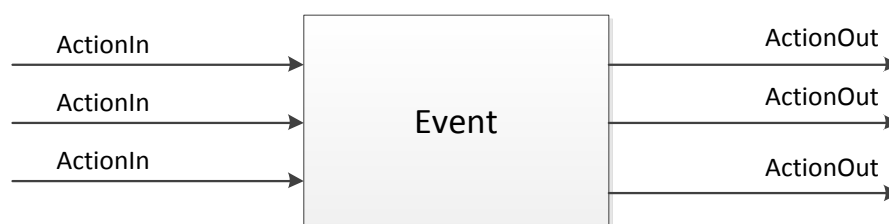


Рис.2 – Модель представления события системы

**Подсистема контроля состояний** предназначена для загрузки структур данных и инициализации остальных подсистем, является контроллером всей программы. Получив список всех устройств, используемых системой, производит настройку и инициализацию, создает потоки контроля интерфейсов, осуществляющих контроль входных воздействий и выдачу выходных воздействий. Каждый отдельный поток контролирует свой интерфейс. Обмен информацией между потоками и подсистемой работы с интерфейсами осуществляется через очередь сообщений.

**Подсистема синхронизации** предназначена для синхронизации всех имитируемых систем и штатной аппаратуры, также может использоваться для

---

повышения точности измерения временных интервалов за счет получения меток времени от высокоточного генератора.

**Подсистема диагностики** реализует ведение журнала событий, происходящих в каждом из интерфейсов, а также системного журнала программы для отражения событий самой программы. Отчеты имитатора по окончанию режима поступают на вход системы анализа результатов испытаний для последующего контроля правильности прохождения режима.

**Подсистема работы с интерфейсами** предназначена для реализации единого интерфейса взаимодействия с платами расширения. По команде от подсистемы контроля состояний производится запуск потоков ожидания приема сообщений, по приему сообщения производится выдача принятого сообщения в очередь сообщений для его последующей обработки в подсистеме контроля состояний.

### **Архитектурные особенности модели**

Общая архитектура модели базируется на основе модулей, каждый, из которых имеет законченный интерфейс для возможности использования в другом проекте. В разработке рекомендуется применять такие парадигмы программирования как паттерн «одиночка» (singleton) и «фабрика объектов» (abstractfactory). Паттерн «одиночка» используется как защита от создания нескольких одинаковых объектов (например, инициализация одной интерфейсной платой несколькими программными модулями), и для доступа к системной диагностике в любом месте программного кода. Фабрика объектов необходима для логического объединения структур данных унаследованных от базового типа и имеющих различную реализацию. Например, контейнер типа ActionIn имеет различные реализации для разных типов интерфейсов, такие как ActionInRS422 и ActionInNet.

### **Общий принцип работы**

Универсальный имитирующий комплекс выполняет обмен информацией по одному или нескольким интерфейсам. При поступлении с объекта контроля, сигнал поступает в модуль взаимодействия с платами, затем передается в очередь соответствующего интерфейса, действующую по принципу буфера FIFO, после чего попадает в интерфейсный модуль. Модуль обрабатывает сигнал и выдает соответствующую команду, состоящую из события и времени, в блок контроля, который, в свою очередь, отправляет её в виртуальное событие (Event). Событие содержит входные воздействия (ActionIn), имеющие 4 вида состояний: ON, OFF, Sleep и Delete. Активация входного воздействия выполняется с помощью обязательного временного входного воздействия (ActionInTime) в заданное время. После активации, ActionIn принимает состояние OFF, которое переходит в ON после поступления сигнала из блока контроля. Таким образом, блок контроля перебирает все ActionIn входы событий до тех пор, пока не найдется соответствующий событию блок Event.

Для реализации сложной логики работы системы существует возможность событий обмениваться служебной информацией с другими событиями, в этом случае, после выхода из блока Event, сигнал, через блок контроля поступает в следующий блок виртуального события. Данный механизм является основой для метода моделирования работы группы внешних систем.

После обработки события блок выдает в качестве выходного воздействия посылку для кодового или релейного взаимодействия, которая через блок контроля поступает в соответствующий интерфейсный модуль и передается в модуль взаимодействия с платами, который отправляет сигнал в соответствующую плату. Дополнительным преимуществом при предложенном построении является то, что имитирующий комплекс выполняет функцию регистратора: все команды, поступающие

и отправленные в модули взаимодействия с платами, регистрируются в файле диагностики.

Общий принцип работы модели универсального имитирующего комплекса изображен на рисунке 3.

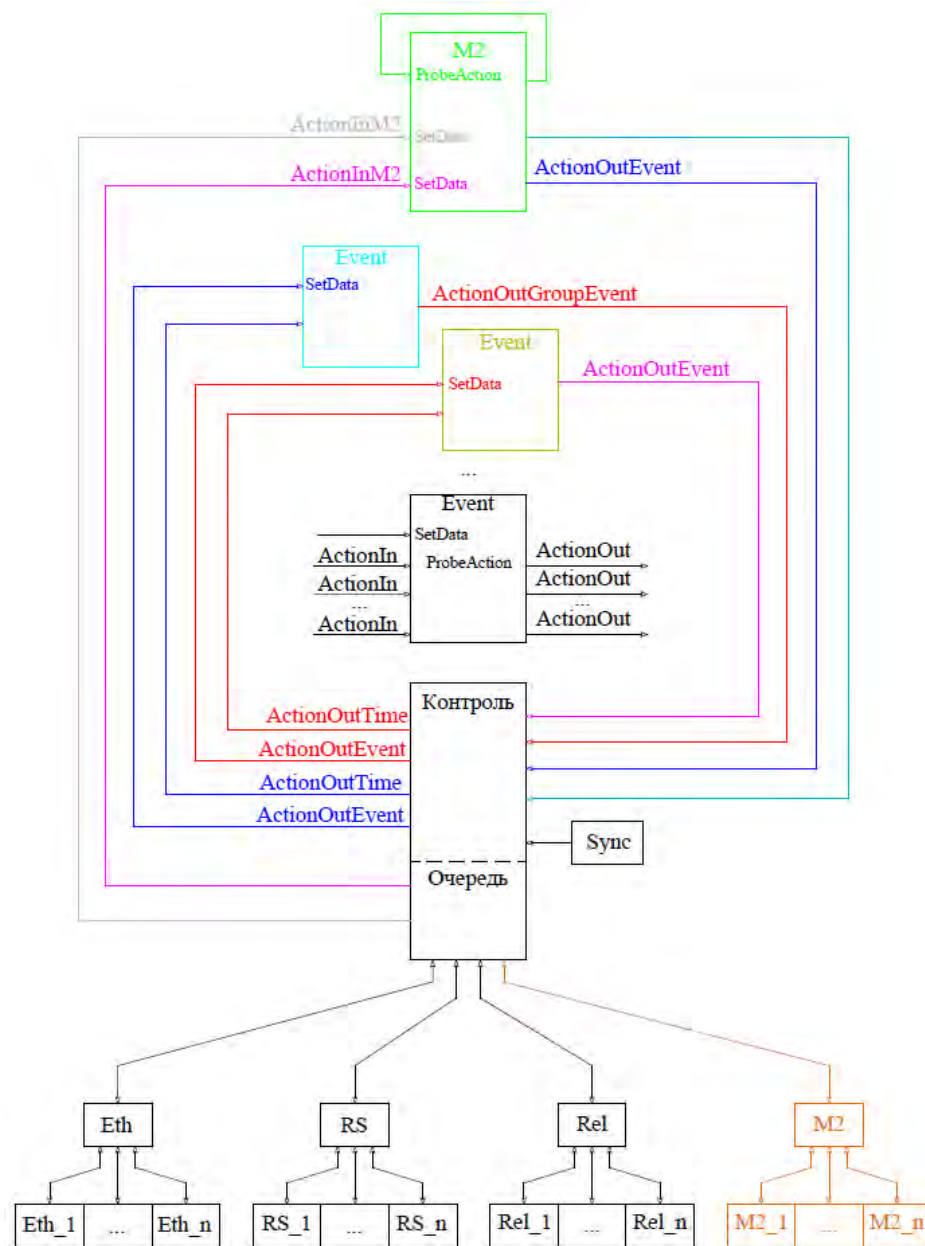


Рис.3 – Общий принцип работы модели универсального имитирующего комплекса

### Метод моделирования работы группы внешних систем

Модель универсального имитирующего комплекса может быть применена и для моделирования группы (каскада) систем. Рассмотрим типовую ситуацию для систем управления изделиями ракетно-космической техники, где в качестве объекта контроля выступает наземная цифровая вычислительная система (НЦВС), и имитируются системы: прибор преобразования команд и сигналов (ППКС) и бортовая аппаратура системы управления (БАСУ). При этом в реальной системе информационное

взаимодействие НЦВС – ППКС осуществляется по протоколу ГОСТ-52070 (M2), ППКС – БАСУ – релейные команды и сигналы (Rel). Схема приведена на рисунке 4.



Рис.4 – Группа (каскад) систем

Метод моделирования группы внешних систем заключается в том, что:

- выделяется одно событие (Event), отвечающее за информационное взаимодействие с объектом контроля (в примере, по протоколу ГОСТ-52070) данное событие обеспечивает моделирование системы, взаимодействующей напрямую с объектом контроля;

- выделяется необходимое количество событий двух типов для моделирования команд и сигналов соответственно (количество определяется по перечню команд и сигналов) во вторую имитируемую систему (в примере - БАСУ);

- на базе модели универсального имитирующего комплекса осуществляется их связь через выходные воздействия в событие (ActionOut) и группа задач (ActionOutEvent), при этом воздействия типа группа задач отвечают за реализацию логических связей и описывают алгоритм функционирования второй имитируемой системы (в примере - БАСУ) в заданном режиме работы. Алгоритм функционирования задается в конфигурационном файле верхнего уровня, а связь выходных воздействия в события в файле нижнего уровне (аппаратная часть), тем самым имитируя работу первой системы по преобразованию информации из одного протокола (как правило, последовательного) в другой (как правило, параллельный).

Принципиальная схема работы метода приведена на рисунке 5.



Рис.5 – Принципиальная схема метода моделирования группы внешних систем

### Имитационное моделирование и модельно-ориентированный подход

Дополнительным преимуществом созданной модели является то, что в качестве функциональных модулей (плагинов) могут выступать как модули собственной разработки, так и модули, сгенерированные из моделей, реализованных в MatLab Simulink (и State Flow) с адаптером собственной разработки для подключения. Данная функция позволяет отрабатывать алгоритмическую часть сначала на модели, а затем повторять необходимый объем испытаний на реальной аппаратуре испытательного стенда.

---

Возможность подключения кода, сгенерированного из моделей MatLab Simulink (и State Flow), позволяет реализовать подход модельно-ориентированного проектирования при создании комплексов вооружения, военной и специальной техники и интегрировать предложенное в рамках работы решение в существующие передовые процессы разработки.

### **Результаты внедрения**

На основе модели универсального имитирующего комплекса разработано одноименное программное обеспечение. Это программное обеспечение внедрено в программно-аппаратные комплексы для испытательных стендов систем управления научно-производственного объединения автоматики им. академика Н.А. Семихатова (Екатеринбург).

На сегодняшний день на базе программного обеспечения универсального имитирующего комплекса разработано и введено в эксплуатацию более 250 единиц имитационного оборудования для испытательных стендов систем управления изделиями военной и специальной техники. За счет унификации при использовании данного подхода трудозатраты на создание перечня систем удалось сократить на 70% и обеспечить параллельную разработку испытательного оборудования для 10 испытательных стендов количеством разработчиков, которое ранее применялось для запуска трех стендов.

Предложенные в работе модели универсального имитирующего комплекса и метод моделирования работы группы внешних систем внедрены в учебный процесс ФГАОУ ВПО «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина» по дисциплине «Проверочно-пусковая аппаратура систем управления ракетными комплексами», изучаемой магистрантами по направлению «Бортовая и наземная аппаратура ракетных комплексов».

### **Заключение**

Создание универсального имитирующего комплекса на базе рассмотренной модели в значительной степени ускоряет задачу разработки и запуска испытательных стендов на этапе отработки комплексов вооружения, военной и специальной техники. Применение предложенной модели при создании имитационного оборудования обеспечивает унификацию подходов и расширяет функционал испытательной базы для обеспечения отработки, как в штатном режиме работы, так и при внесении сбоев и неисправностей.

Метод моделирования группы внешних систем позволяет создавать модели каскадов внешних систем, при этом разделяя реализацию работы систем по конфигурационным файлом. Данный метод особенно актуален для отработки подсистем систем управления изделиями ракетно-космической техники (наземной, бортовой, контрольно-испытательной), так как они в большинстве случаев имеют в своем составе вычислительную часть (ЦВС) и периферийные приборы, которые сопрягаются в свою очередь с другой подсистемой, например, (ЦВС и периферийная аппаратура из состава НАСУ сопрягаются с БАСУ).

Сохранение в архитектуре модели динамического компонента – функциональных модулей (плагинов), обеспечивает гибкость решения и возможность широкого распространения для различных классов задач, таких как имитационное оборудование, аппаратура автономных проверок, аппаратура для разовых испытаний, в том числе на позициях головной организации и МО РФ.

---

## Литература

1. **Микрин Б.А.** Бортовые комплексы управления космическими аппаратами и проектирование их программного обеспечения // изд. МГТУ им. Н.Э. Баумана, 2003. – С.94-128.
2. **Дудин Н.В., Хохряков В.А., Щепочкин И.Н.** Процесс отработки программного обеспечения ФГУП научно-производственного объединения автоматики имени академика Н.А. Семихатова// Актуальные проблемы ракетно-космической техники («III Козловские чтения»). Сборник трудов – Самара: ФГУП «ГНПРКЦ «ЦСКБ-Прогресс», 2013. – С.218-221.
3. **Капитонов В.А., Муштакова И.В.** Комплексная оценка влияния степени резервирования системы управления на основные характеристики РН// Актуальные проблемы ракетно-космической техники («III Козловские чтения»). Сборник трудов – Самара: ФГУП «ГНПРКЦ «ЦСКБ-Прогресс», 2013. – С.83-87.
4. **Журавлев А.В., Шашмурин И.В.** Оптимизация процесса экспериментальной отработки на испытательных стендах систем управления //10-ый Всероссийский межотраслевой молодежный конкурс научно-технических работ и проектов: аннотации конкурсных работ. – Москва: Московский авиационный институт (национальный исследовательский университет), 2018. – С.121-124.
5. **Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес.** Приемы объектно-ориентированного проектирования. Паттерны проектирования // Библиотека программиста. – Санкт-Петербург: Изд. «Питер», 2001. – С.368.