

ИНТЕГРИРОВАННАЯ МОДЕЛЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

к.т.н. П.Ф. Буданов, д.т.н., проф. Ю.Г. Даник,
к.ф.-м.н. С.Е. Кальной, к.т.н. С.А. Олизаренко

Предложена интегрированная модель программного обеспечения на концептуальном и логическом уровнях представления и реализованная с использованием универсального языка объектно-ориентированного моделирования UML в Enterprise Architect v 2.50.

Постановка проблемы и анализ литературы. В [1] была рассмотрена задача разработки системы автоматизированного управления и контроля воздушной среды (САУК ВС) закрытых и открытых объектов, обладающей высокой степенью быстродействия, надежности и многофункциональности. В связи с этим был разработан алгоритм функционирования САУК ВС [2]. Известные программно-аппаратные реализации не позволяют реализовать алгоритм функционирования САУК ВС для проведения контроля и управления параметрами воздушной среды в режиме реального времени. Для этого необходимо использовать методические рекомендации [3] по интегрированному моделированию [4 – 6] программного обеспечения САУК ВС.

Целью статьи является разработка пакета программного обеспечения, который позволяет проводить расчеты быстродействия чувствительных элементов, составляющих основу САУК ВС.

Изложение основного материала. В рамках создания программного обеспечения (ПО) САУК ВС разработана интегрированная модель программного обеспечения (ИМ ПО) на концептуальном и логическом уровнях представления. Интегрированная модель ПО САУК представляет собой результат выполнения объектно-ориентированного анализа согласно методических рекомендаций по моделированию ПО [4], представленных в [5 – 6]. Интегрированная модель ПО САУК реализована с использованием универсального языка объектно-ориентированного моделирования UML в Enterprise Architect v 2.50.

Данная модель является основой для последующего проектирования ПО САУК с целью дальнейшей автоматической кодогенерации.

На рис. 1 представлен фрагмент ИМ ПО САУК, включающий: диа-

грамму прецедентов (случаев использования); потоки событий, описывающие прецеденты «Контроль параметров воздушной среды», «Установка режима чистой вентиляции», «Анализ концентрации опасных газов», «Анализ содержания кислорода и углекислого газа»; по три сценария (первичный и два вторичных) для соответствующих прецедентов; диаграммы последовательностей сообщений объектов для первичных сценариев прецедентов «Контроль параметров воздушной среды» и «Установка режима чистой вентиляции»; диаграмму классов – абстракций сущностей предметной области; диаграммы переходов состояний для классов «Исполнительное устройство» и «Система воздухообеспечения»; диаграмму деятельности (соотносимой в нашем представлении с соответствующей операцией) «Изменить режим работы» для класса «Система воздухообеспечения».

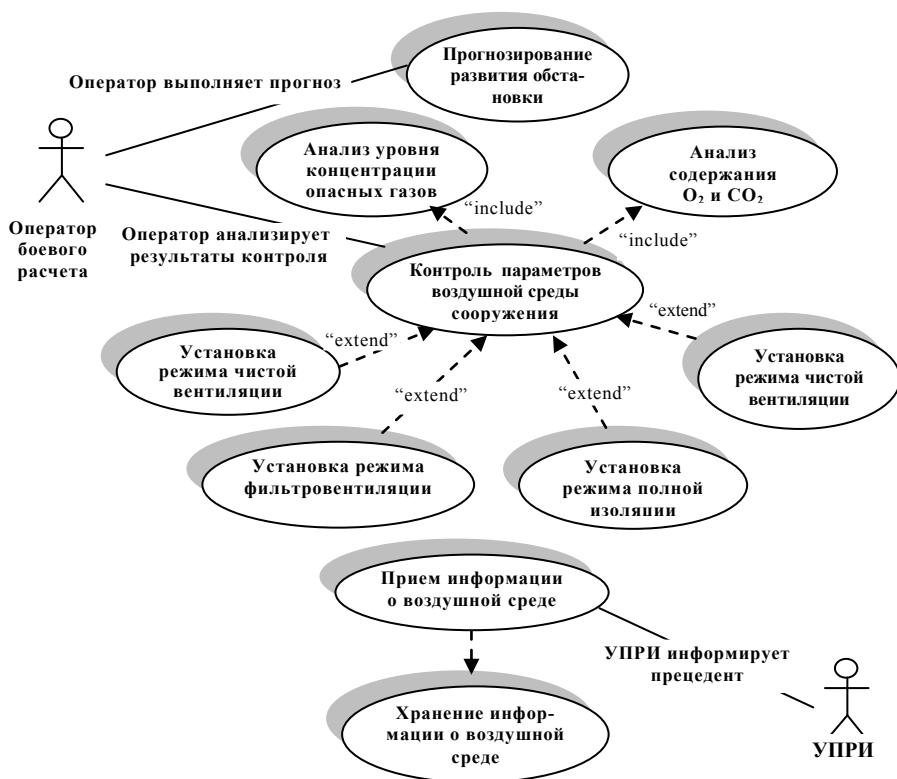


Рис. 1. Диаграмма модели прецедентов

Диаграмма прецедентов (рис. 1) представляет внешний вид разрабатываемого ПО САУК. Другими словами, поведение элементов программного обеспечения описывается с помощью модели прецедентов, отображающей:

системные прецеденты (“Прием информации о воздушной среде”; “Контроль параметров воздушной среды”);

системное окружение (“оператор боевого расчета”; “УПРИ”, “связи между прецедентами и внешними действующими лицами”; “оператор выполняет прогноз”, “оператор анализирует результаты контроля”).

Выполнение прецедентов отображается с помощью соответствующих потоков событий. Поток событий для прецедентов – это последовательность событий, необходимых для обеспечения требуемого поведения – поток событий для прецедента “Контроль параметров воздушной среды сооружения”, поток событий для прецедента “Установка режима чистой вентиляции” и т.д.

Для описания того, как реализуются прецеденты, взаимодействуя между группами объектов, используются сценарии. Сценарий представляет собой одиночный проход по потоку событий – сценарии для прецедента “Контроль параметров воздушной среды сооружения”, сценарии для прецедента “Установка режима чистой вентиляции” и т.д.

Таким образом, модель программного обеспечения САУК воздушной среды объектов закрытого и открытого типов на концептуальном логическом уровне представления информации, полученной в результате объективно-ориентированного анализа предметной области, включает следующие **потоки**:

- 1) поток событий для прецедента “Контроль параметров воздушной среды сооружения”;
- 2) поток событий для прецедента “Установка режима чистой вентиляции”;
- 3) поток событий для прецедента “Анализ уровня концентрации опасных газов”;
- 4) поток событий для прецедента “Анализ содержания кислорода и углекислого газа”;

а также **сценарии**:

- 1) сценарии для прецедента “Контроль параметров воздушной среды сооружения”;
- 2) сценарии для прецедента “Установка режима чистой вентиляции”;
- 3) сценарии для прецедента “Анализ уровня концентрации опасных газов”;
- 4) сценарии для прецедента “Анализ содержания кислорода и углекислого газа”.

Отображаются сценарии с помощью диаграмм последовательностей сообщений для первичных сценариев “Контроль параметров воздушной среды сооружения” и “Установка режима чистой вентиляции”.

Сценарии и диаграммы последовательностей сообщений помогают выделить объекты и их взаимодействия в предметной области. Например, в результате анализа первичного сценария “Контроль параметров воздушной среды сооружения” и построения соответствующей диаграммы последовательностей сообщений выделены следующие объекты предметной области: САУК ВС (О1), ЗФВА (О3), КДР (О4) и т.д.

В результате абстрагирования объектов, полученных при построении диаграмм последовательностей сообщений, получаем множество классов (класс – это описание группы объектов с общими свойствами, поведением и отношениями с другими объектами) интегрированной модели. По мере того как добавляются новые классы, их тестовое представление становится неудобным. Диаграммы классов помогают графически представить классы в модели (рис. 2).



Рис. 2. Диаграмма модели классов

Прецеденты и сценарии применяются для описания поведения нескольких элементов программного обеспечения, т.е. взаимодействия объектов – экземпляров класса. Иногда требуется рассмотреть поведение внутри самого класса.

Диаграмма состояний показывает состояния класса, события, которые вызывают переход из одного состояния в другое, действия, являющиеся результатом смены состояний. Так, например, класс “Система воздухообеспечения” (рис. 3) может находиться в пяти состояниях (включая начальное и конечное состояния): состояние создания, состояние включения и т.д. Перевод из состояния в состояние производится шестью событиями: создание экземпляра, включение системы и т.д.,

выполнением пяти действий: создать, включить систему, изменить режим работы и т.д.

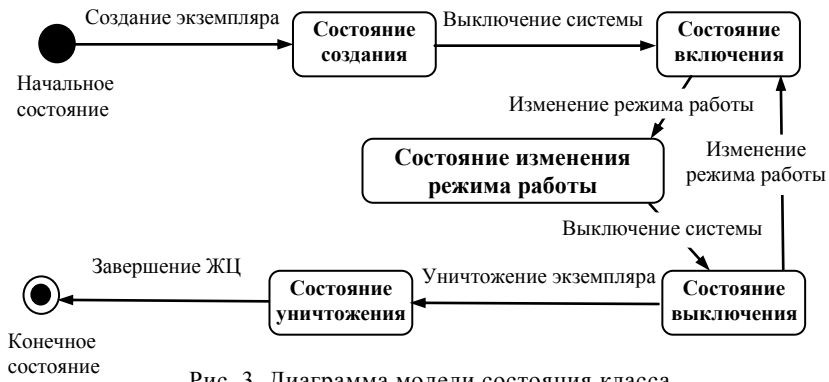


Рис. 3. Диаграмма модели состояния класса

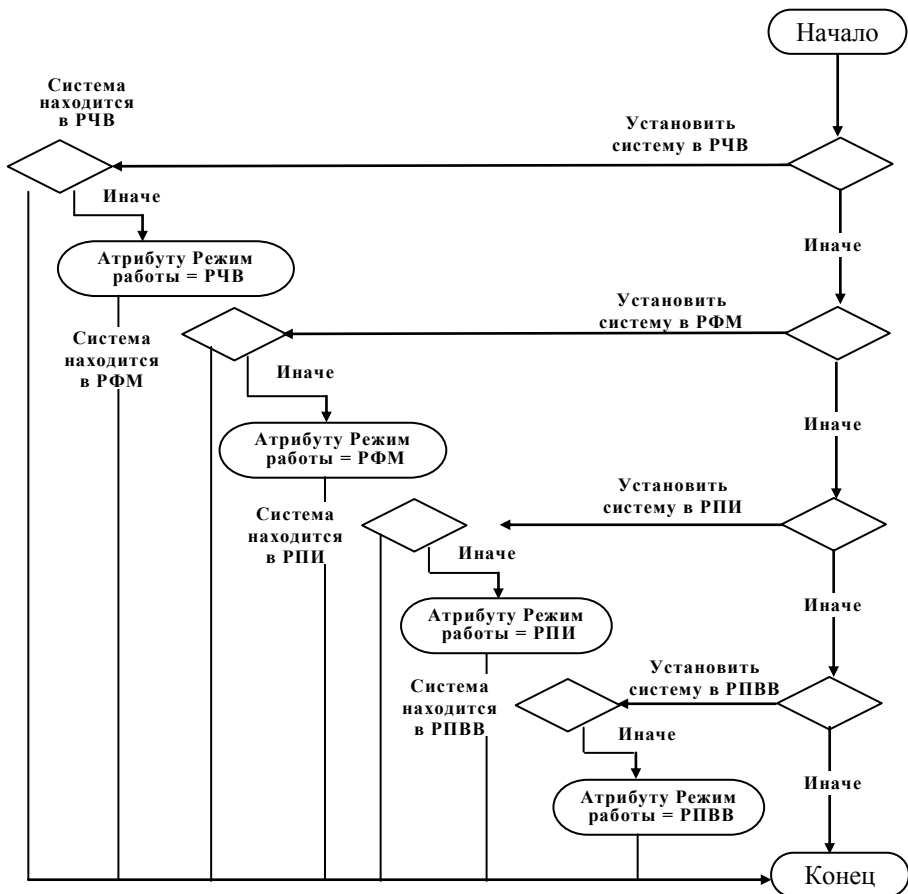


Рис. 4. Диаграмма модели деятельности “Изменить режим работы” для класса “Система воздухообеспечения”

Для описания действий (или точнее деятельностей), выполняемых в заданном состоянии, используются диаграммы деятельностей. Диаграмма деятельностей позволяет отразить внутреннюю реализацию деятельности вплоть до представления неделимых операторов. Так для класса “Система воздухообеспечения” деятельность “Изменить режим работы” может быть задана диаграммой, представленной на рис. 4.

Выводы. На основе универсального языка объектно-ориентированного моделирования UML в Enterprise Architect v 2.50 был реализован алгоритм функционирования системы воздухообеспечения объектов закрытого типа при использовании в качестве сенсоров разработанных фрактальных датчиков контроля воздушной среды.

ЛИТЕРАТУРА

1. Батурін Н.Г., Буданов П.Ф., Даник Ю.Г., Калугін В.Д. Автоматизована система управління параметрами повітряного середовища // Системи обробки інформації. – Х.: НАНУ, ПАНМ, ХВУ. – 2001. – Вип. 6(16). – С. 206 – 210.
2. Буданов П.Ф., Голубничий Д.Ю., Даник Ю.Г., Калугін В.Д. Структура системи управління параметрами воздушной среды объектов // Системи обробки інформації. – Х.: НАНУ, ПАНМ, ХВУ. – 2001. – Вип. 1(17). – С. 177 – 186.
3. Бродин В.Б., Шагурин И.И. Микроконтроллеры. Архитектура программирования, интерфейс. – М.: ЭКОМ, 1999. – 400 с.
4. Кватрани Т. Rational Rose 2000 и UML. Визуальное моделирование. – М.: ДМК, 2001. – 176 с.
5. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. – М.: Финансы и Статистика, – 2000. – 347 с.
6. Грачов В.М., Олизаренко С.А. Разработка модели объектно-ориентированного анализа и проектирования сложных программных систем // Системи обробки інформації. – Х.: НАНУ, ПАНМ, ХВУ. – 2001. – Вип. 5(21). – С. 166 – 171.

Поступила 12.03.2004

БУДАНОВ Павел Феофанович, преподаватель кафедры ХВУ. В 1981 году окончил Ростовское высшее военное командно-инженерное училище. Область научных интересов – применение автоматизированных систем управления.

ДАНИК Юрий Григорьевич, доктор технических наук, профессор, главный научный сотрудник научного центра при ХВУ. В 1987 году окончил Житомирское высшее военное училище радиоэлектроники. Область научных интересов – обработка информации.

КАЛЬНОЙ Сергей Евгеньевич – кандидат физико-математических наук, доцент, доцент ХВУ. Область научных интересов – физика низких температур, теория дифракции.

ОЛИЗАРЕНКО Сергей Анатольевич, кандидат технических наук, начальник лаборатории научного центра при ХВУ. В 1993 году окончил Харьковское ВВКИУ РВ. Область научных интересов – разработка программного обеспечения АСУ.
