УДК 520.24; 681.51

А. Н. Бевзов, А. В. Курочкин, А. А. Лубков, А. Д. Петухов, П. С. Филатов

Институт автоматики и электрометрии СО РАН
пр. Акад. Коптюга, 1, Новосибирск, 630090, Россия

bvz@iae.nsk.su; kurochkin@iae.nsk.su; lubkov@iae.nsk.su
petuhov@iae.nsk.su; pf92@mail.ru

СОЗДАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АСУ БОЛЬШОГО СОЛНЕЧНОГО ВАКУУМНОГО ТЕЛЕСКОПА НА ОСНОВЕ УНИФИЦИРОВАННОГО ПРОЦЕССА РАЗРАБОТКИ

Представлен опыт создания программного обеспечения системы автоматизации Большого солнечного вакуумного телескопа. Основное внимание уделено ключевым стадиям этого процесса. Для достижения возможности повторного использования созданного кода с целью автоматизации других телескопов, а также переносимости кода на другие операционные системы применялась методика объектно-ориентированного программирования, паттернов проектирования и инструментарий Qt.

Ключевые слова: система автоматизации, солнечный телескоп, унифицированный процесс, объектно-ориентированное программирование, паттерны проектирования, Qt.

Введение

Основная задача создания программного обеспечения системы автоматизации Большого солнечного вакуумного телескопа (БСВТ) [1], предназначенного для изучения нестационарных явлений на поверхности Солнца, состоит в том, чтобы обеспечить автоматизированный процесс наблюдения Солнца. Поскольку БСВТ является инструментом проведения научных исследований, в проекте по созданию его системы автоматизации необходимо было предусмотреть возможность внесения изменений в программное обеспечения как по ходу работы над проектом, так и в процессе эксплуатации уже созданной системы. Кроме того, была поставлена задача достижения максимально возможного повторного использования программного кода, который можно было бы применять также на АСУ других телескопов, решающих аналогичные задачи.

Эффективность в решении задач, связанных с необходимостью создания модифицируемого и переносимого кода, была достигнута за счет того, что в ходе работы над проектом использовалась методика унифицированного процесса разработки программного обеспечения [2]. В отличие от более раннего – классического – подхода, основанного на так называемой «водопадной модели» [3], когда сначала почти полностью проектируется вся система, которая затем реализуется в соответствии с принятыми решениями, методика унифицированного процесса предполагает пошаговое, инкрементное и итеративное развитие проекта.

Несмотря на то что упомянутые подходы, равно как и их положительные и отрицательные стороны, хорошо известны, до сих пор нет единого мнения относительно того, какая методика предпочтительнее в том или ином случае. В работе, опираясь на опыт создания программного обеспечения,

Бевзов А. Н., Курочкин А. В., Лубков А. А., Петухов А. Д., Филатов П. С. Создание программного обеспечения АСУ Большого солнечного вакуумного телескопа на основе унифицированного процесса разработки // Вестн. Новосиб. гос. ун-та. Серия: Физика. 2014. Т. 9, вып. 1. С. 87–94.

ISSN 1818-7994. Вестник НГУ. Серия: Физика. 2014. Том 9, выпуск 1

приводятся аргументы, иллюстрирующие предпочтительность использования унифицированного процесса для решения задач автоматизации, контекст которых определяется особенностью создания инструментов для проведения научных исследований.

К основным шагам унифицированного процесса относятся стадии анализа и постановки требований, проектирования, реализации, отладки и тестирования, сопровождения.

Фаза постановки требований и анализа

На начальном этапе к создаваемой АСУ БСВТ был выдвинут ряд требований:

1) выполнить модернизацию системы автоматизации на основе использования современного программного и аппаратного обеспечения, существенно расширив ее возможности;

2) добиться возможности повторного использования разработанного программного кода для систем автоматизации других телескопов [4; 5];

3) предусмотреть возможность управления телескопом и проводить наблюдательные эксперименты в удаленном режиме по Интернету;

4) обеспечить возможность управления оптико-механическими узлами телескопа с помощью переносного пульта управления;

5) добиться возможности альтернативного выбора различных программных и аппаратных средств;

6) предусмотреть возможность вносить изменения в программную и аппаратную архитектуру уже в процессе выполнения проекта.

С учетом перечисленных требований было принято решение вести разработку на основе использования основных положений унифицированного процесса разработки программного обеспечения [2], в основе которого лежит пошаговое, инкрементное и итеративное выполнение работ.

Кроме требований к создаваемой системе автоматизации, был также учтен опыт создания подобных систем для других телескопов. Наиболее интересным и ценным оказался опыт по созданию ATST (Advanced Technology Solar Telescope) [1], поскольку этот проект учитывает общие тенденции развития систем автоматизации для телескопов разного вида – звездных, солнечных, радиотелескопов.

Среди наиболее важных особенностей, которые были учтены при проектировании АСУ БСВТ, можно назвать следующие:

1) выделение подсистем по функциональному принципу, а также их интеллектуализация и децентрализация;

2) использование унифицированных средств связи;

3) возможность иметь свободу выбора различных программных и аппаратных средств для создания систем автоматизации;

4) использование моделирования некоторых узлов для обеспечения отладки и диагностики создаваемых программ.

Варианты использования

На начальном этапе на основе сформулированных требований были созданы варианты использования [2], которые представляют собой описание основных работ, выполняемых на БСВТ: юстировка, калибровка, загрузка и выгрузка программного обеспечения, подготовка к работе, проведение наблюдений и регистрация данных.

Варианты использования дорабатывались на разных стадиях проекта. На начальном этапе это было текстовое описание того, какие действия должны осуществлять различные подсистемы АСУ на языке предметной области. Затем, на стадии проектирования, варианты использования были расширены и переработаны уже с учетом того, какие наиболее важные классы внутри этих подсистем отвечают за те или иные действия при работе телескопа. На более поздней стадии проектирования и ранней стадии реализации варианты использования применялись для того, чтобы определить функциональность основных классов программного обеспечения.

В наиболее важных случаях, для того чтобы понять, как происходит взаимодействие между различными объектами, а также для лучшего понимания того, какую функциональность должен предоставить тот или иной класс, на основе вариантов использования были созданы диаграммы кооперации, взаимодействий и последовательности.

[1] http://atst.nso.edu

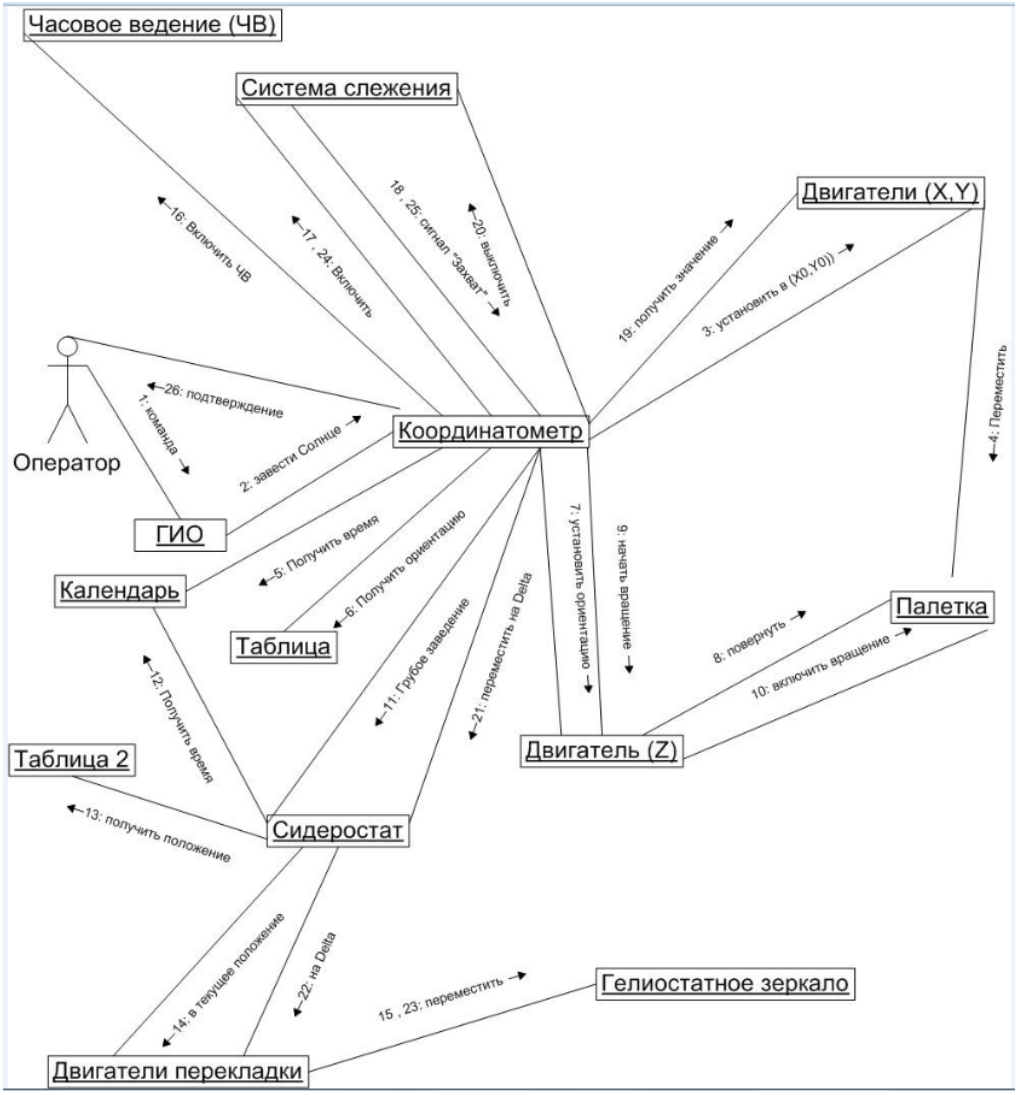

Рис. 1. Диаграмма кооперации при автоматизированном заведении Солнца в трубу телескопа

В качестве примера, иллюстрирующего наличие достаточного большого количества объектов, а также показывающего разнообразие существующих между ними связей, которые должны быть корректно учтены при создании управляющей программы, на рис. 1 представлена диаграмма кооперации, описывающая процесс автоматизированного заведения Солнца в трубу телескопа.

Словарь предметной области

Одним из первых важных шагов, выполненных на стадии формулировки требований и анализа, стало создание словаря предметной области, содержащего определения и толкование ряда понятий, которые используются при описании работы телескопа и, как следствие, при создании программного обеспечения для управления этим телескопом. Наиболее важными оказались следующие группы понятий.

- Понятия уровня предметной области: сидеростат, гелиостатное зеркало, щель спектрографа, координатометр, палетка координатометра, изображение Солнца на палетке координатометра, центр палетки координатометра, начало координат инструментальной системы.
- Понятия аппаратного уровня: компьютер оператора, контроллер, двигатель, датчик, подсистема (как отдельные модули – сидеростат, координатометр, спектрограф и др.).

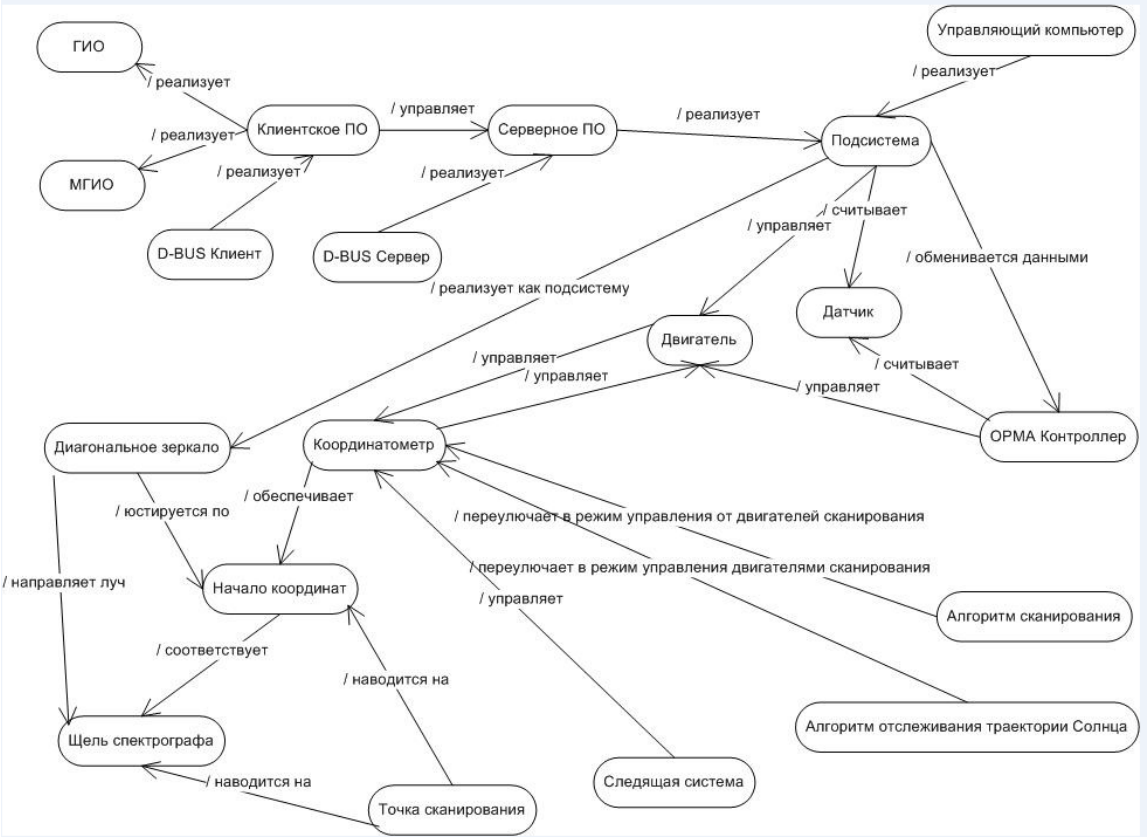

Рис. 2. Основные классы анализа и отношения между ними

- Понятия функционально-алгоритмического уровня: система координат (инструментальная, гелиографическая), точка сканирования, захват точки сканирования, область сканирования, алгоритм сканирования.
- Понятия программного уровня: подсистема (как принцип выделения отдельных частей программного обеспечения), клиентская программа, графический интерфейс оператора (стационарный, мобильный, удаленный), D-BUS клиент, серверное программное обеспечение, D-BUS сервер, конфигурационный файл, имитаторы двигателей, следящей системы, вакуумной системы.

Наличие словаря предметной области позволило разработчикам использовать единый понятийный аппарат и на его основе определить классы анализа и отношения между ними (рис. 2).

Фаза проектирования

Важнейшим шагом на раннем этапе проектирования стало определение программной архитектуры АСУ БСВТ. В этой архитектуре можно выделить два уровня: системный и прикладной.

Системная архитектура определятся выбором следующих инструментов и решений по разработке программного обеспечения:

1) инструментарий Qt, включающий библиотеку Qt в качестве средства прикладного программирования;

2) шина D-BUS в качестве средства обмена данными между различными подсистемами системы автоматизации;

3) язык программирования C++ для разработки основного программного обеспечения АСУ;

4) процесс-ориентированный язык Рефлекс [6] для создания программного обеспечения вакуумной подсистемы.

На создание *прикладной архитектуры* программного обеспечения оказали влияние несколько факторов. Наиболее важными стали основные иерархии классов проектирования, а также отношения между этими классами (рис. 3) в нотации языка UML [7]. Эти иерархии первоначально были получе-

ны на основе анализа выявленных ранее классов, а затем расширены и модифицированы по ходу работы.

Иерархия классов ITCU, включающая интерфейсный класс ITCU, класс CTCU с общими для разных телескопов методами, а также индивидуальные для разных типов телескопов классы реализации CTCU_BSBVT (для вакуумного телескопа) и CTCU_AST (для автоматизированного солнечного телескопа), предназначена для решения задачи начальной загрузки и выгрузки программного обеспечения серверной части АСУ.

Иерархия классов ISubSystem предназначена для решения задач, которые возлагаются на различные подсистемы.

Иерархия классов IDevice предназначена для работы с аппаратными устройствами – датчиками, двигателями и др.

Иерархия классов IConnection предназначена для работы с различными линиями связи, по которым периферийные устройства подключены к управляющему компьютеру телескопа.

Благодаря использованию классических в объектно-ориентированном программировании отношений наследования в перечисленных иерархиях классов удалось решить задачу повторного использования кода.

На этапе проектирования также была определена целесообразность использования паттернов проектирования [8]. Так, например, паттерн «одиночка» удобен для созда-

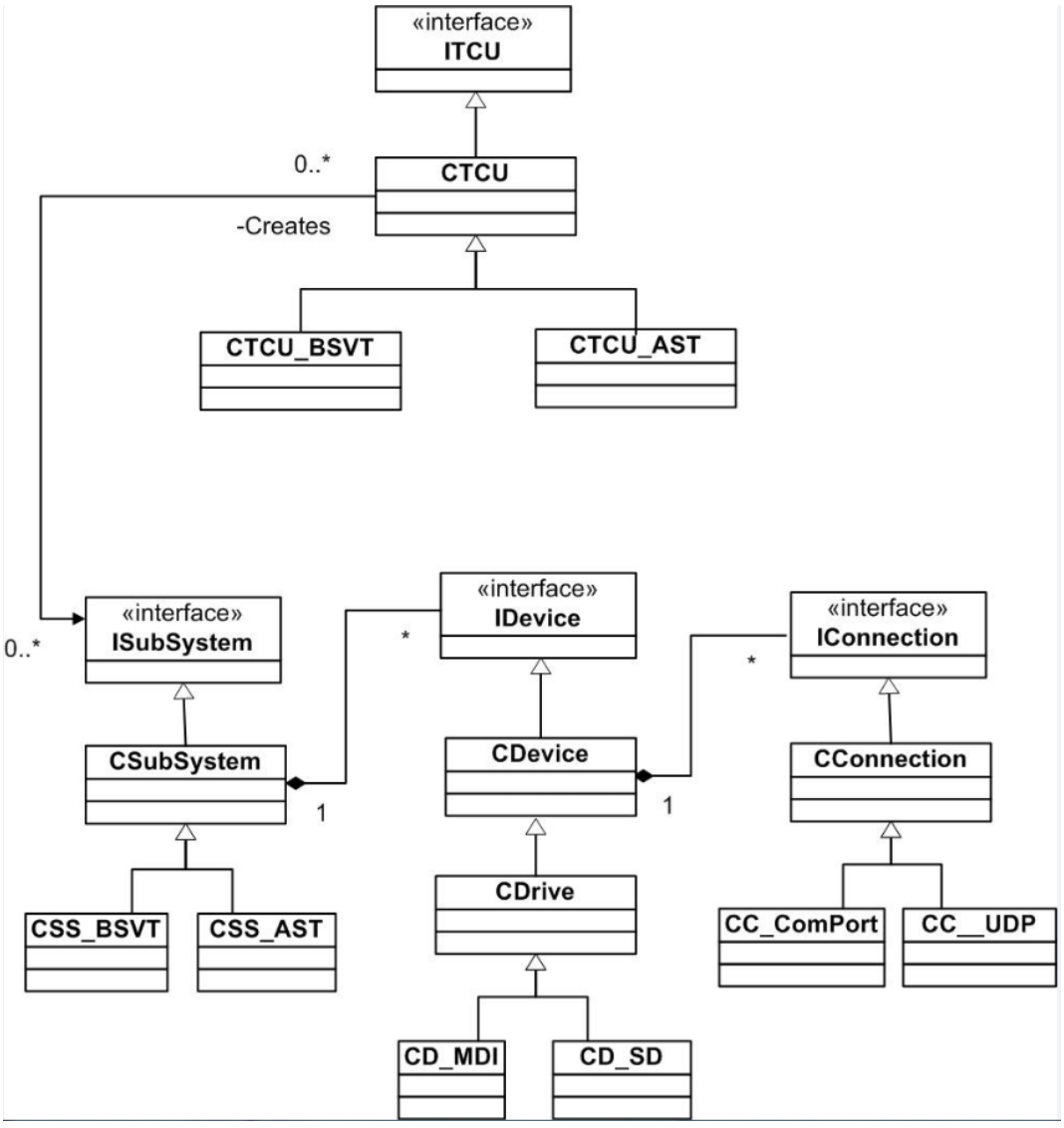

Рис. 3. Диаграмма основных классов

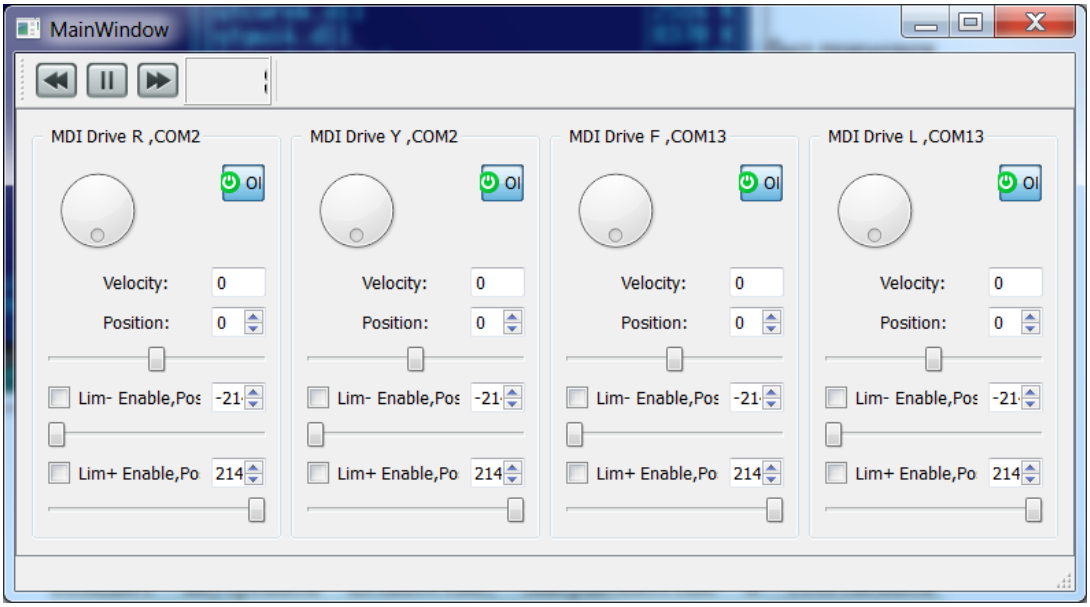

Рис. 4. Скриншот программы имитатора двигателей

ния единственного объекта в иерархии «*IConnection*» в том случае, когда к одной линии связи подключается несколько двигателей. Паттерны проектирования «абстрактная фабрика» и «фабричный метод» удобны при создании одинаковых семейств продуктов для разных типов телескопов. Паттерн проектирования «шаблонный метод» также целесообразен для реализации в чем-то похожих, а чем-то отличающихся алгоритмов на различных видах телескопах. Так, например, процесс сканирования на телескопе АСТ [4], который является целостатом и на котором нет вращения изображения Солнца, отличается от сканирования на телескопе БСВТ, который является сидеростатом и на котором приходится компенсировать вращение изображения Солнца за счет дополнительного поворота палетки координатометра в процессе сканирования.

Фаза реализации

Опыт реализации данного проекта показал, что благодаря выбору упомянутых методов и средств разработки была построена гибкая архитектура программного обеспечения, которая позволила вносить неизбежные в такого рода проектах изменения в программном и аппаратном обеспечении.

Так, например, в ходе реализации проекта полностью было изменено аппаратное и программное обеспечение для подсистемы термоконтроля градиента температур входного иллюминатора. Однако это минимально отразилось на создании общего программного обеспечения АСУ, несмотря на то что проектирование нового варианта системы термоконтроля было осуществлено уже на этапе реализации проекта.

Аналогичные замечания можно сделать по поводу системы вакуумирования, в которой кардинальным образом изменилось аппаратное обеспечение, что, однако, не помешало интегрировать ее в уже созданную архитектуру.

Еще одним существенным изменением, возникшим по ходу проекта, явилась необходимость использовать контроллер между основным компьютером АСУ и двигателями, чего раньше не планировалось. Это привело к необходимости решения сразу двух задач: создание нового программного обеспечения, а также необходимость интеграции созданной программы с уже существующими модулями. Задачу интеграции удалось решить благодаря наличию в архитектуре программного обеспечения унифицированного интерфейса (иерархия «IConnection»), ориентированного на передачу прикладных данных независимо оттого, какое соедине-

ние реально используется для подключения периферийных устройств к компьютеру.

Фаза отладки и тестирования

Одним из важных и успешных решений, принятых еще на самых ранних стадиях работы над проектом, было создание имитаторов двигателей и следящей системы, которые позволили отлаживать основное программное обеспечение АСУ в условиях территориальной удаленности разработчиков от объекта автоматизации в лабораторных условиях при наличии только ограниченного стендового оборудования.

На рис. 4 представлен скриншот программы имитатора двигателей, которая позволяет: 1) задавать и изменять конфигурацию двигателей; 2) визуализировать отработку команд; 3) сохранять и восстанавливать образ системы после выключения питания; 4) моделировать временные характеристики отклика двигателей при отработке команд; 5) ускорять время отработки некоторых команд АСУ БСВТ.

Благодаря этому имитатору, а также имитатору следящей системы у разработчиков была возможность создавать и отлаживать работу основного программного обеспечения АСУ на самых ранних стадиях работы над проектом.

Заключение

В настоящее время проект по созданию программного обеспечения системы автоматизации БСВТ находится в стадии реализации и проведения стендовых испытаний, а полученный опыт позволяет сделать следующие выводы.

1. Использование унифицированного процесса предпочтительно по сравнению с классической «водопадной моделью», так как унифицированный процесс позволяет более гибко и оперативно учитывать изменения, которые неизбежно приходится вносить в программное и аппаратное обеспечение по ходу работы.

2. В условиях необходимости отлаживать созданное программное обеспечение на удалении от объекта автоматизации при отсутствии полной комплектации программного и аппаратного обеспечения, которое будет использоваться на реальном объекте, очень полезным оказывается создание необходимых имитаторов конечных устройств. Затраты, связанные с необходимостью создавать такие имитаторы, оправданы тем, что благодаря им удается отладить большую часть программы в условиях стендовых испытаний.

3. Использование инструментария Qt также оправдало возложенные на него надежды. В настоящее время инструментарий Qt благодаря своей библиотеке классов удобен как в качестве средства, так и в качестве среды разработки. Использование шины D-BUS, а также технологии сигналов и слотов позволило эффективно решать задачи информационного обмена между компьютерами распределенной сети. Инструментарий Qt эффективен как средство для создания межплатформенной переносимости, что, в свою очередь, позволяет расширить возможности выбора наиболее подходящего программного и аппаратного обеспечения с точки зрения ценовой политики и программно-аппаратной совместимости и интеграции различных программных и аппаратных средств.

Список литературы

1. *Степанов В. Е.*, *Банин В. Г.*, *Круглов В. И.*, *Григорьев В. М.*, *Китов А. К.*, *Шамсутдинов М. А.*, *Коровкин А. И.*, *Кузнецов Ю. А.*, *Осак Б. Ф.*, *Трифонов В. Д.* Экспериментальный макет Большого солнечного вакуумного телескопа (ЭМ БСВТ) СибИЗМИР // Новая техника в астрономии. 1979. Т. 6. С. 42–51.

2. *Якобсон А.*, *Буч Г.*, *Рамбо Дж.* Унифицированный процесс разработки программного обеспечения. М., 2002. 496 с.

3. *Royce W. W.* Managing of the Development of Large Software Systems // Proc. of the IEEE WESCON. August, 1970. P. 1–9.

4. *Лубков А. А.*, *Зотов А. А.*, *Котов В. Н.*, *Лылов С. А.* Автоматическое управление солнечным телескопом // Датчики и системы. 2008. № 10. С. 8–12.

5. *Bevzov A. N.*, *Kotov V. N.*, *Lubkov A. A.*, *Lylov C. A.*, *Perebeynos S. V.* Automated Control System for Horizontal Solar Telescope of Sayansk Solar Observatory // Proc. of the IASTED International Conf. on Automation, Control, and Information Technology: Automation, Control, and Applications (ACIT-2010). Novosibirsk, 2010. P. 178–182.

6. *Зюбин В. Е.* Язык Рефлекс. Математическая модель алгоритмов управления // Датчики и системы. 2006. № 5. С. 24–30.

7. *Рамбо Дж.*, *Якобсон А.*, *Буч Г.* UML: Специальный справочник. СПб., 2002. 296 с.

8. *Гамма Э.*, *Хелм Р.*, *Джонсон Р.*, *Влиссидес Дж.* Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб., 2001. 366 с.

Материал поступил в редколлегию 13.09.2013

A. N. Bevzov, **A. V. Kurochkin**, **A. A. Lubkov**, **A. D. Petukhov**, **P. S. Filatov**

SOFTWARE DEVELOPMENT PROCESS FOR AUTOMATION SYSTEM OF LARGE SOLAR VACUUM TELESCOPE

The paper describes experience of software development for automated system for Large Solar Vacuum Telescope. Most attention is given to the work evolution on the different stages of this process. Automated system software has been developed using Object-Oriented Programming, Design Patterns and Qt tools to achieve reusable code, that can be used in other telescopes and be portable to different operating systems.

Keywords: automatic system, solar telescope, unified process, Object-Oriented Programming, Design Patterns, Qt.