



Автоматный метод проектирования и создания параллельных реагирующих систем

Ирина Афанасьева
САО РАН,
Университет ИТМО

29.09.2017 СПИИРАН

Представлены результаты диссертационного исследования на соискание ученой степени кандидата технических наук по специальности 05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.

Научный руководитель: Новиков Федор Александрович, д.т.н., профессор СПбПУ Петра Великого

Предлагается метод, нацеленный на достижение более высоких показателей **надежности и производительности** при проектировании архитектуры и реализации **асинхронных реагирующих систем**

Приводится **графический язык** описания кооперативного взаимодействия автоматных объектов

Изложены **результаты применения** метода при разработке программного обеспечения для управления высокоточным научным оборудованием **в наблюдательной астрономии**

Актуальность исследования

3

Высокопроизводительные **ответственные** встроенные информационные и управляющие системы

— в науке, в медицине, в космосе ... —

требуют безотказного миллисекундного отклика при терабайтах данных

Рост требований к программному обеспечению →

Совершенствование методов проектирования →

Изменение архитектуры программного обеспечения

Для систем со сложным поведением **автоматный подход** обладает наибольшей эффективностью.

В автоматном программировании остаются недостаточно проработанными вопросы

асинхронности и параллельности,

поэтому тема исследования является **актуальной**.

Цель исследования

- Разработка технологии и сопутствующих инструментальных средств для полного описания и автоматизации процесса создания параллельных реагирующих программных систем

Задачи

- Создание **метода разработки архитектуры** реагирующих программных систем в форме кооперативного взаимодействия автоматных объектов
- Проектирование и реализация **языка** описания автоматных объектов (**ЧАО**)
- Разработка **инструментальных средств** для поддержки созданного метода

Потоки управления и данных (control & data flow)

- Явный протокол, неявные состояния. Эти модели подчеркивают порядок выполнения действий
- (блок-схема, императивные языки программирования, DFD)

Последовательность сообщений (sequence of messages)

- Только протокол, индуктивное описание. Эти модели подчеркивают знание источника и места назначения данных
- (диаграммы взаимодействия UML, MSC, BPMN)

Явное выделение состояний (state machines)

- Явные состояния, неявный протокол. Эти модели фокусируются на реакциях системы на события
- (конечный автомат, машина Тьюринга, сеть Петри, SDL)

Трансформационные и реагирующие системы

6

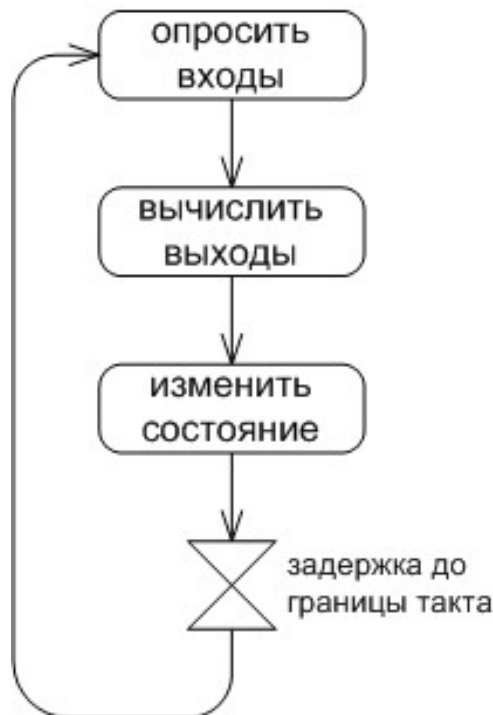
Трансформационные системы

- преобразуют входные данные в выходные (компиляторы, архиваторы)
- запускаются по мере надобности

Реагирующие системы

- реагируют на изменения внешней среды в темпе, задаваемым средой (системы контроля и сбора данных)
- постоянно действующие

Мониторинг,
синхронная
реагирующая
система



Прерывания,
асинхронная
реагирующая
система



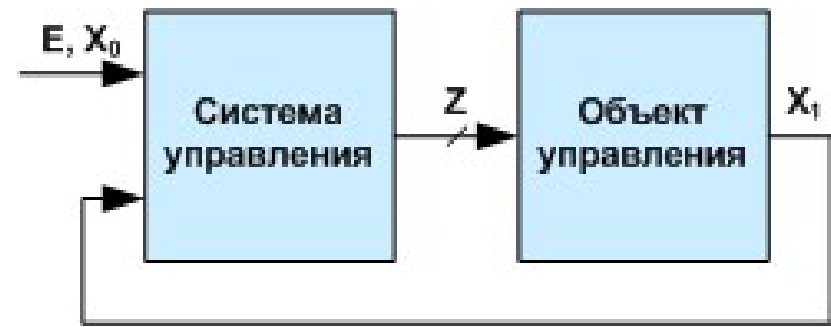
Предыстория идеи

7

- 1991 – автоматное программирование (Шалыто)
- 2001 – SWITCH-технология (Шалыто)
- 2005 – программное обеспечение для управления ПЗС-системой с DINACON-2 (Афанасьева)
- 2005 – MetaAuto (Канжелев)
- 2008 – ПО для ПЗС+DINACON-3 (Афанасьева)
- 2012 – automata objects cooperation (Новиков)
- 2015 – переработка программного обеспечения для DINACON-4 (Афанасьева)
- 2016 – кооперативное взаимодействие автоматных объектов, язык ЧАО (Афанасьева)
- 2017 – ПО для ПЗС+DINACON-5 (Афанасьева)

Поведение реагирующих систем определяется с помощью моделей, основанных на переходах состояний.

Программные системы предлагается разрабатывать в виде совокупности автоматизированных объектов управления, каждый из которых содержит **систему управления** (система взаимодействующих конечных автоматов) и **объект управления**.



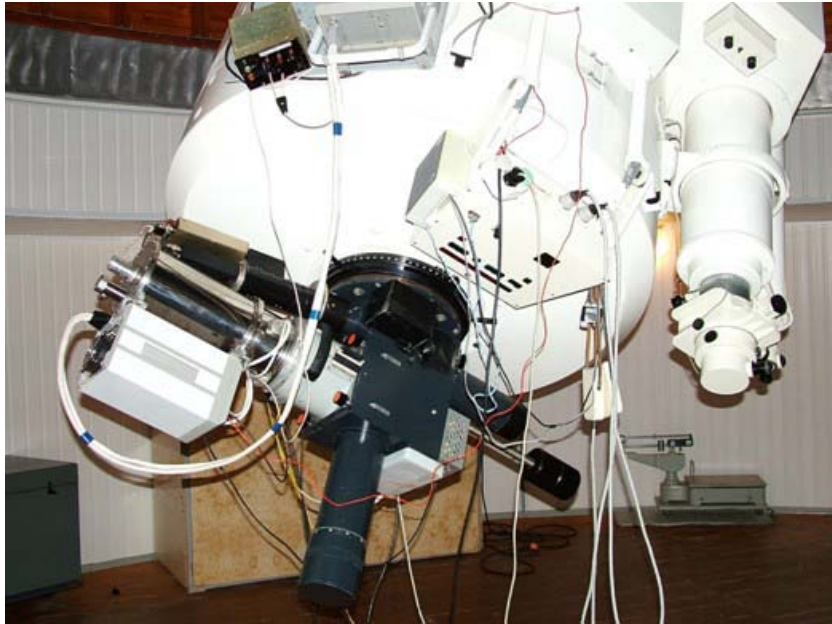
- Состояния
- События и входные переменные
- Функции переходов
- Выходные воздействия
- Система конечных автоматов
- Объект управления

Преимущества:

- Обладает наибольшей эффективностью для систем со сложным поведением
- Формальное и понятное описание поведения
- Автоматическая генерация кода по диаграммам переходов
- Возможность верификации программ
- Проектная документация

Недостатки:

- Основная сложность в автоматном программировании – построение автоматов
- В большинстве случаев автоматы проектируются вручную



ПЗС-система: основной инструмент для фотометрических и спектроскопических методов наблюдений

- *охлаждаемая камера со встроенным матричным ПЗС-фотоприемником изображения*
- *контроллер, управляющий работой камеры*
- *интерфейс передачи данных в удаленный компьютер*

Функциональные требования: обеспечение проведения астрономических наблюдений:

- инициализация контроллера
- накопление, считывание данных в различных режимах
- сохранение полученных изображений
- визуализация и анализ данных
- настройка ПЗС-системы
- автоматический мониторинг и постоянный вывод телеметрии

Нефункциональные требования:

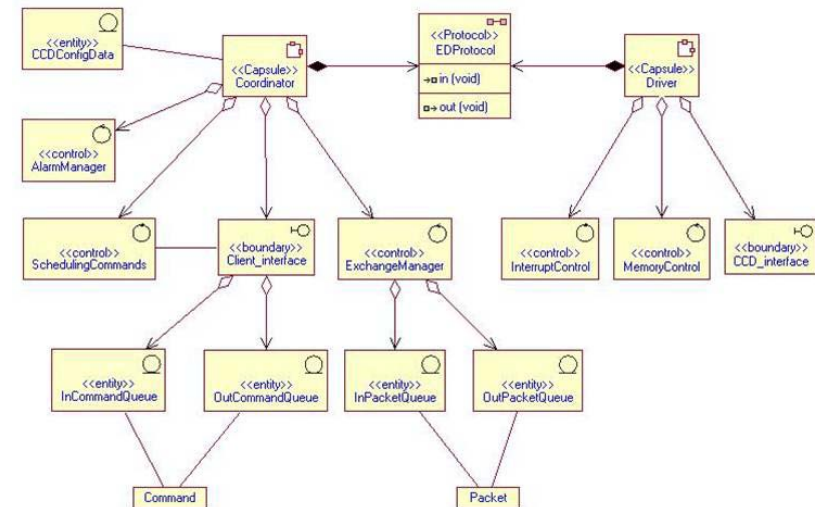
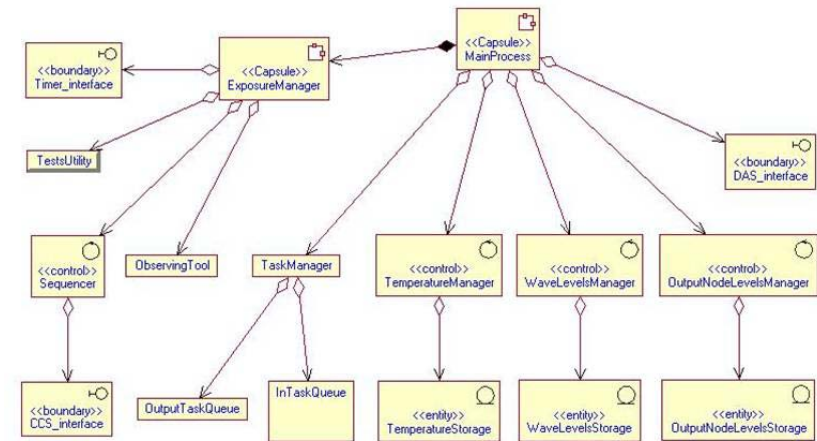
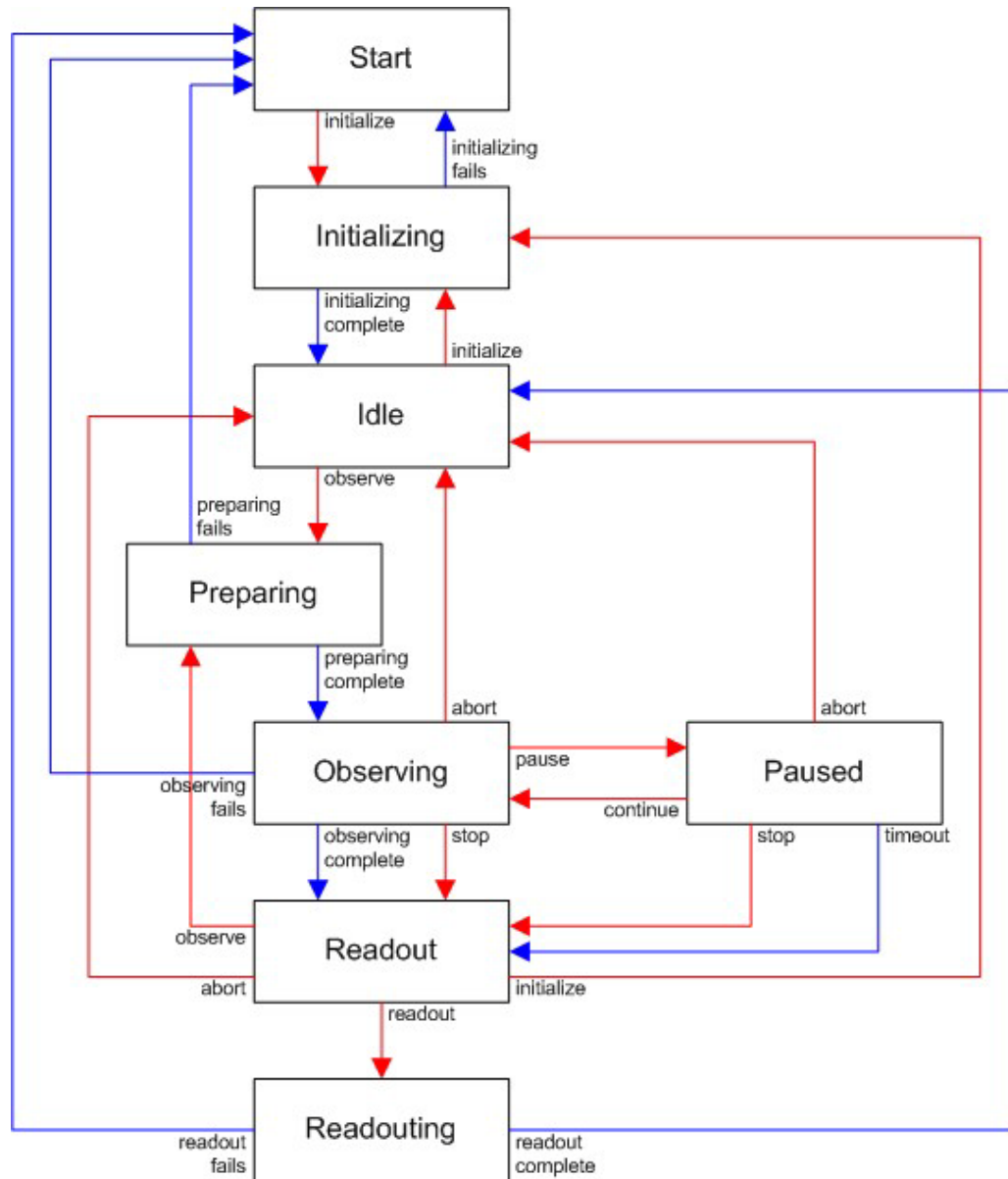
- быстрая адаптация к новым ПЗС-системам
- эффективность и надежность в эксплуатации
- гибкость и расширяемость
- возможность повторного использования

DINACON-2 (2005, UML + RUP)



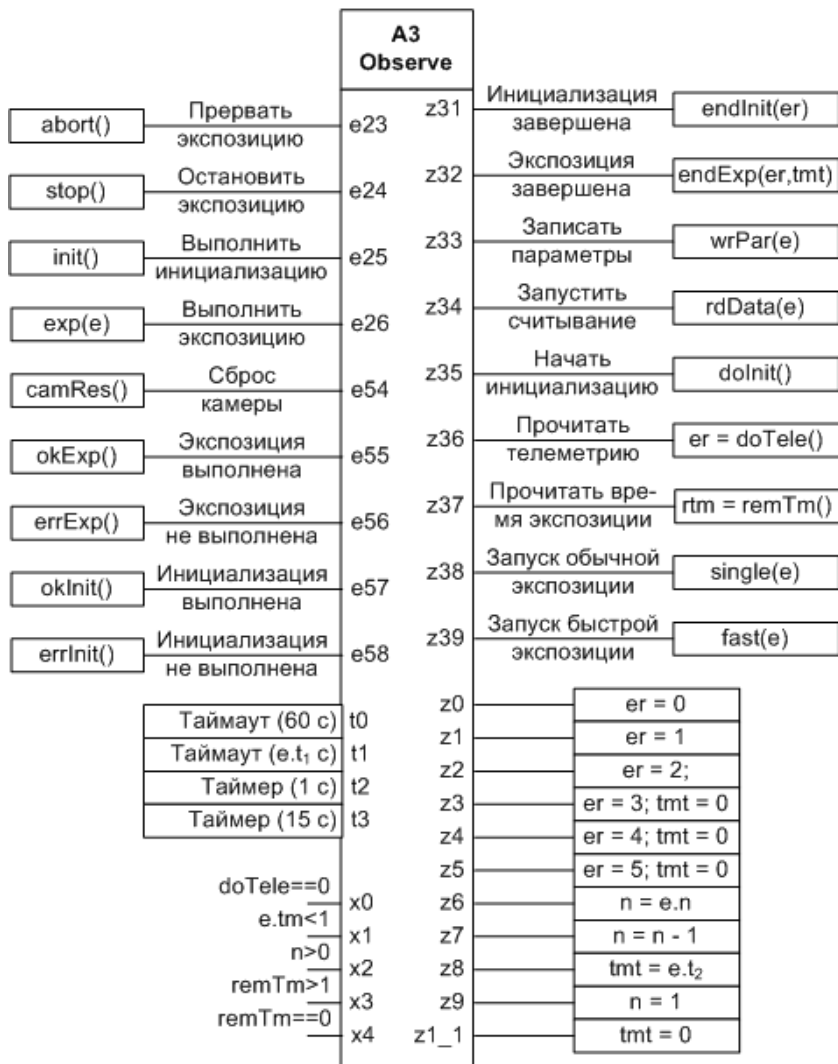
DINACON-3 (2008, UML + COMET)

13



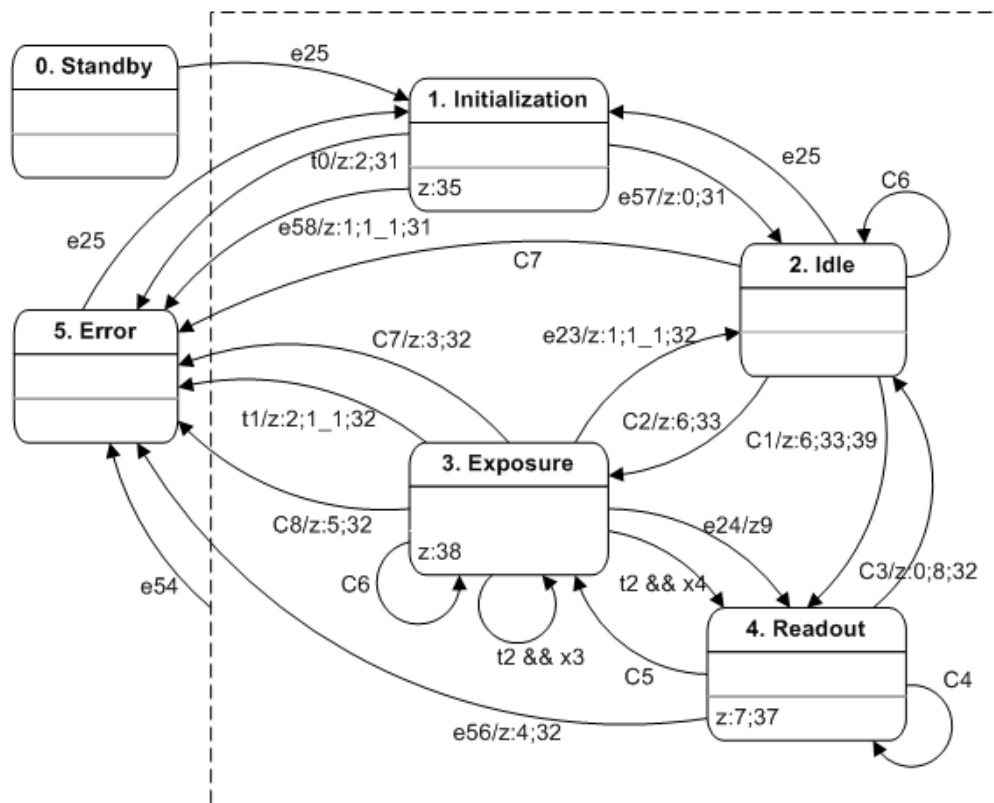
Управление процессом наблюдений (2015)

14



C1 = e26 && x1
 C2 = e26 && ¬x1
 C3 = e55 && ¬x2
 C4 = e55 && x2 && x1
 C5 = e55 && x2 && ¬x1
 C6 = t3 && x0
 C7 = t3 && ¬x0
 C8 = t2 && ¬x3 && ¬x4

Автомат	
Имя:	A3
Название:	Observe

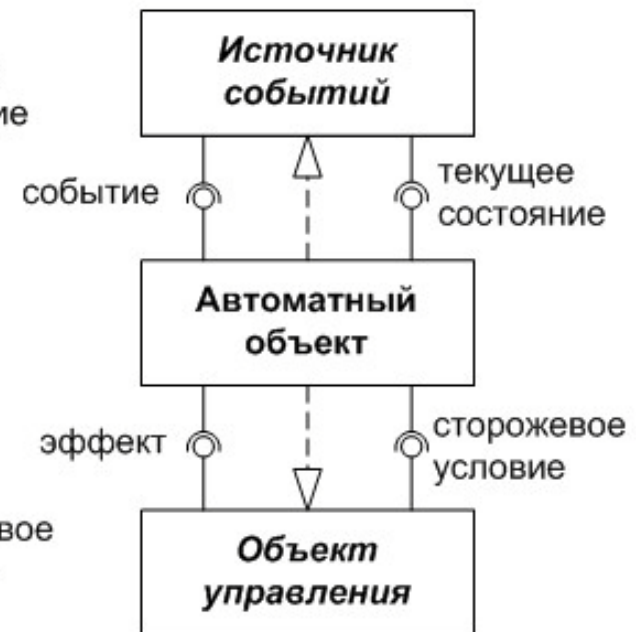
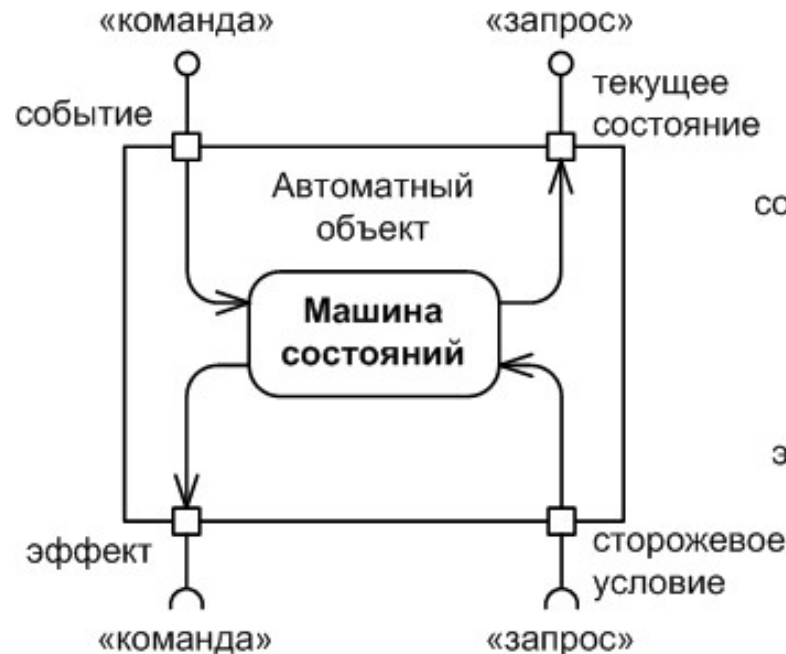


er	Код ошибки
e	Параметры экспозиции
n	Количество экспозиций
tmt	Таймаут

Описание поведения → **модель взаимодействующих автоматных объектов**: источники событий, управляющие автоматы и объекты управления **уравниваются в правах** и являются автоматными объектами.

Взаимодействие: специализированные интерфейсы

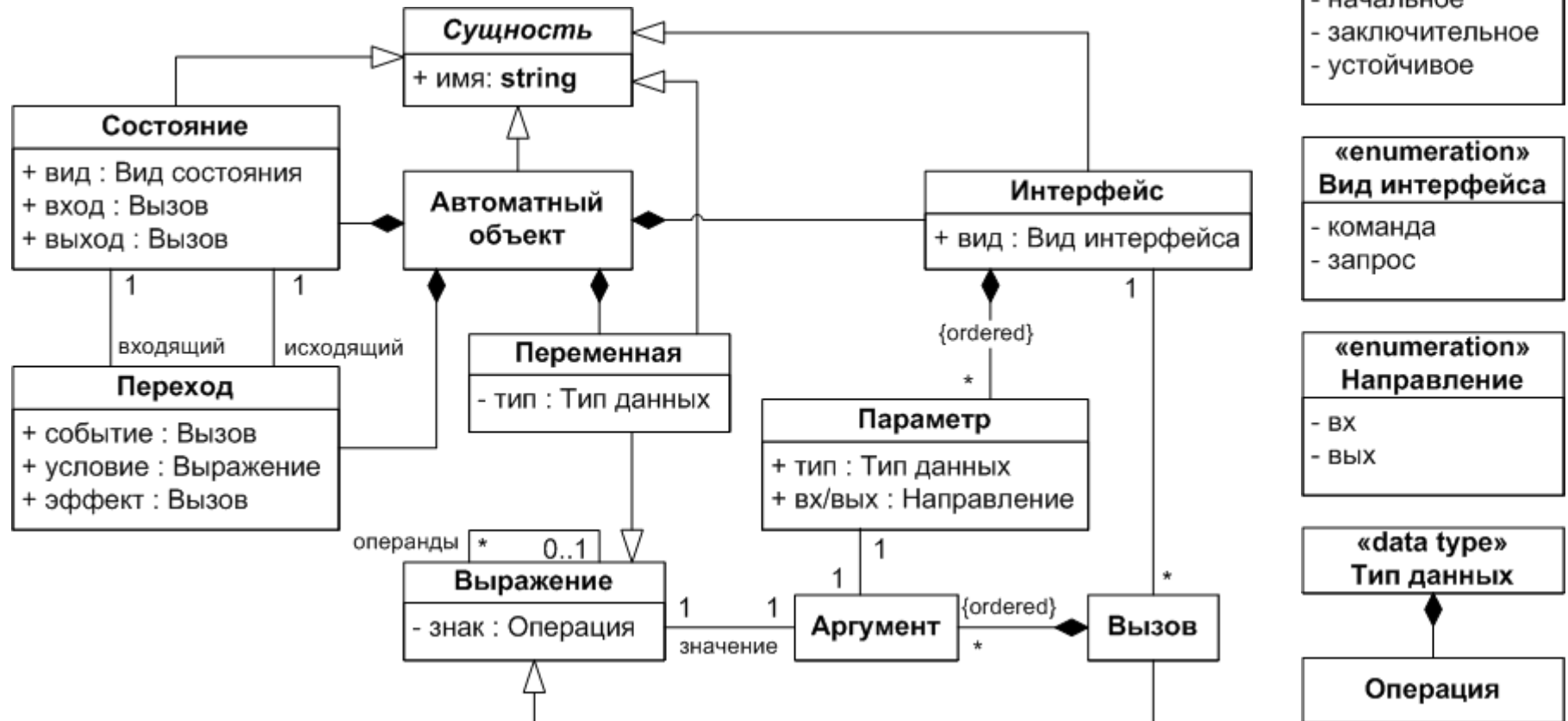
- Событие
- Текущее состояние
- Эффект
- Сторожевое условие



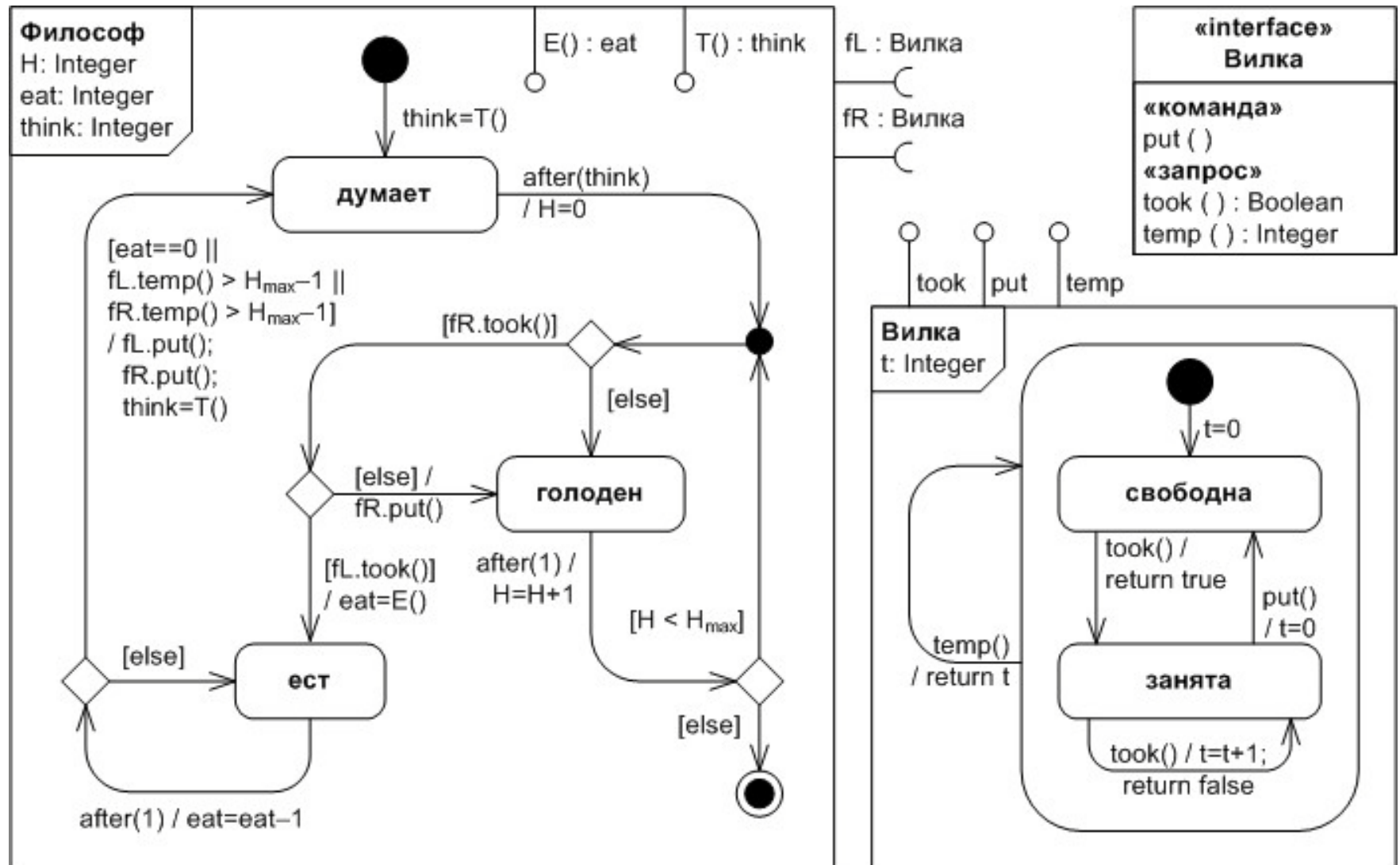
Интерфейсы взаимодействия

- «запрос» доставляет значения и не меняет состояние – синхронный вызов
- «команда» меняет состояние объекта, но не доставляет значений – асинхронный вызов
- **событие** на переходе = предоставляемая команда, аргументы событий инициализируют локальные переменные
- **сторожевое условие** на переходе = логическое выражение над значениями, которые доставляют требуемые запросы
- **эффект** = требуемые команды, аргументы – значения локальных переменных
- автоматный объект может предоставлять запросы о текущем **состоянии** и значениях локальных переменных
- **схема связей** = событие $\leftrightarrow$ эффект, условие $\leftrightarrow$ состояние

Язык **ЧАО** (CIAO: Cooperative Interaction of Automata Objects) предназначен для визуализации сложного поведения и быстрого прототипирования реагирующих систем



Обедающие философы



Метод построения архитектуры программного обеспечения с помощью модели взаимодействующих автоматных объектов сводится к следующим шагам:

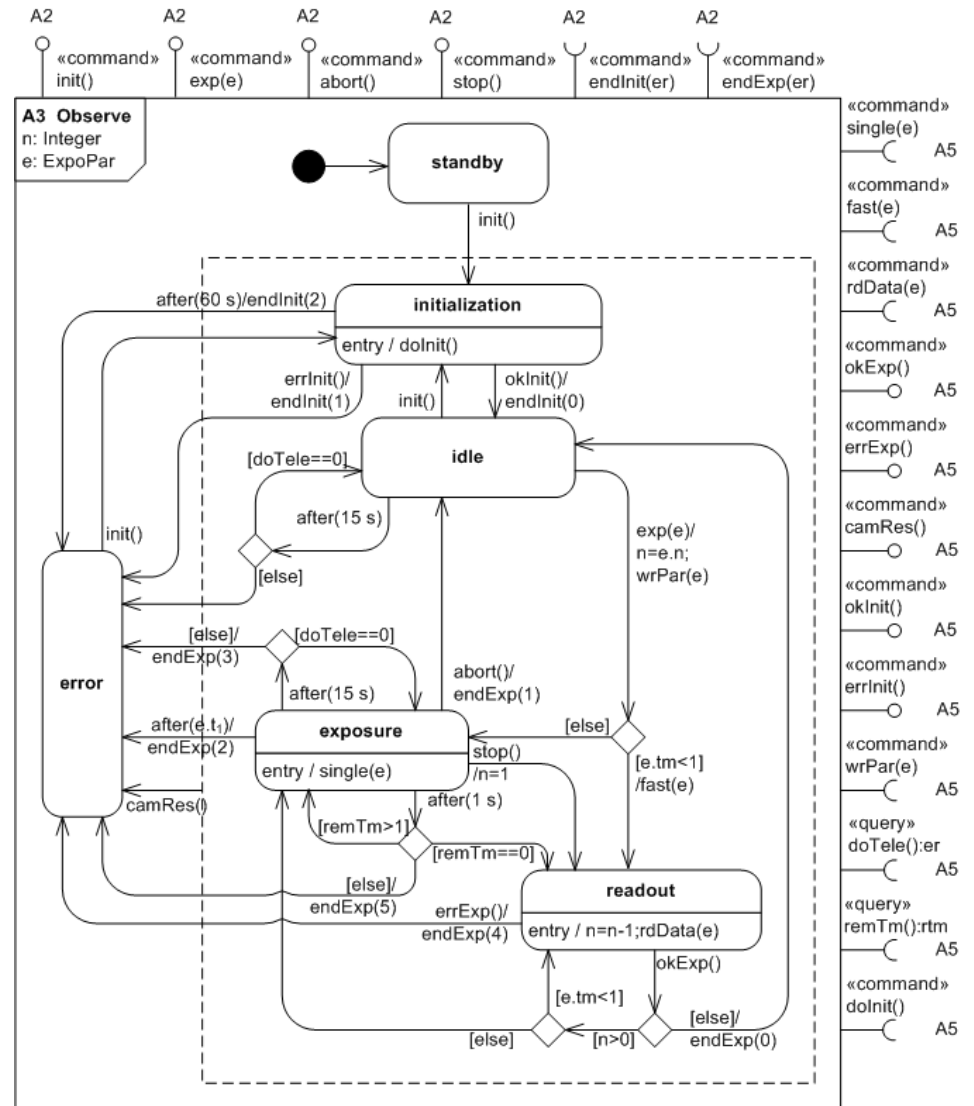
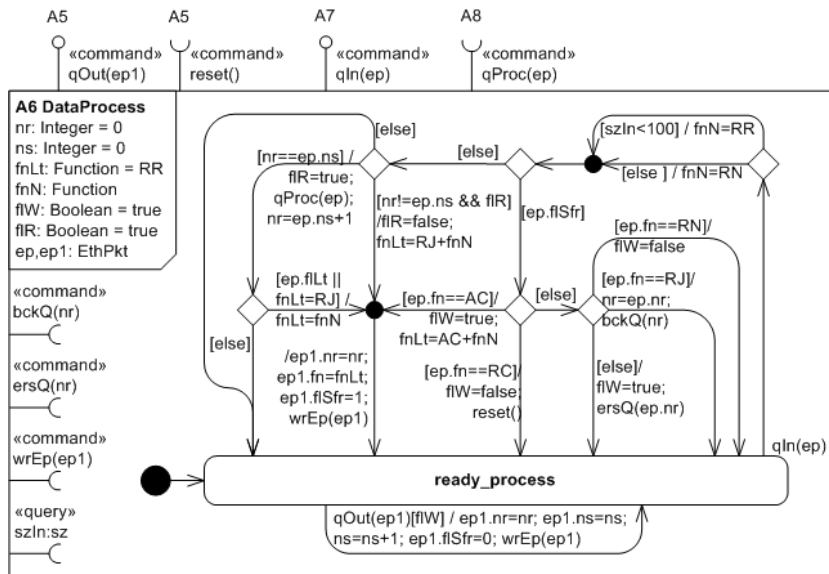
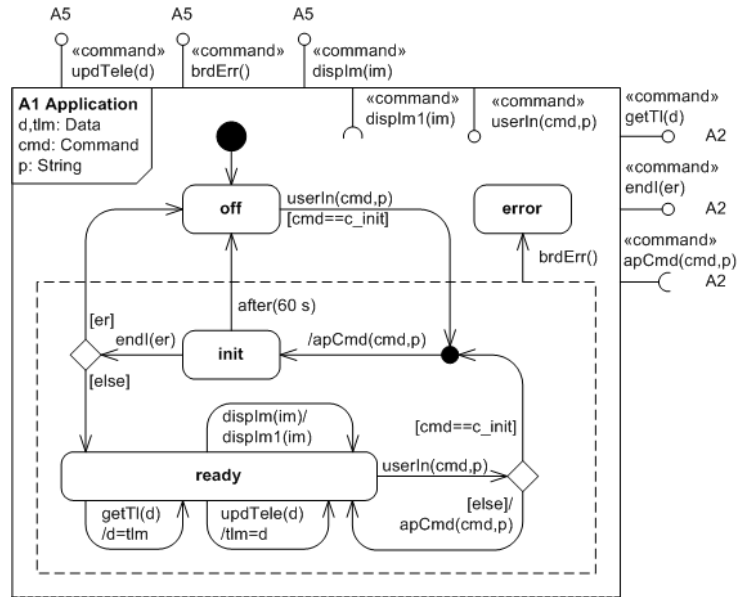
- выделение автоматных объектов
- построение автоматов
- построение диаграммы связей автоматов
- выделение компонентов
- определение классов и интерфейсов

Выделение автоматных объектов

Были выделены следующие объекты со сложным поведением:



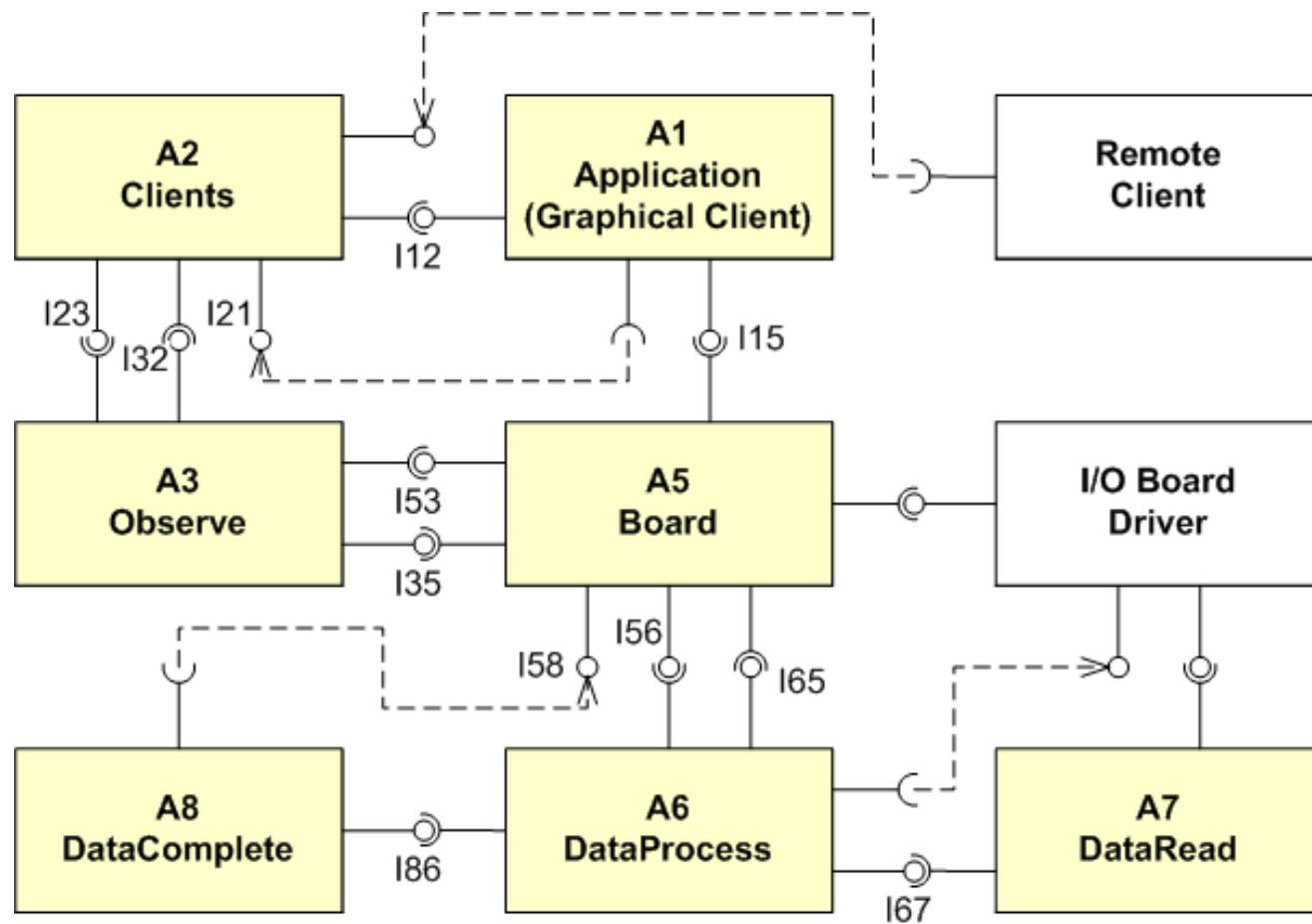
Построение автоматов



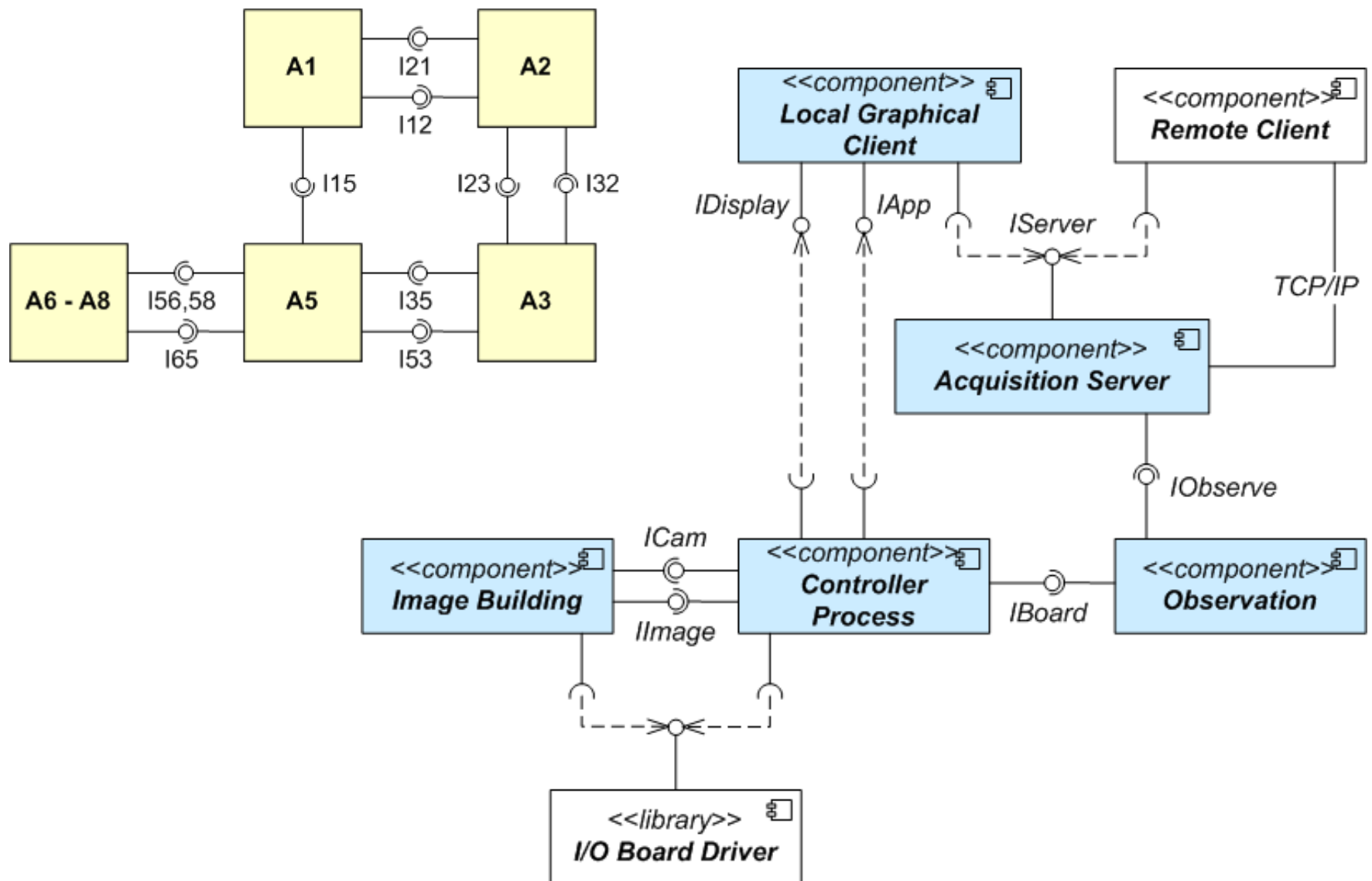
«command» single(e) A5
«command» fast(e) A5
«command» rdData(e) A5
«command» okExp() A5
«command» errExp() A5
«command» camRes() A5
«command» okInit() A5
«command» errInit() A5
«command» wrPar(e) A5
«query» doTele():er A5
«query» remTm():rtm A5
«command» dolnit() A5

Построение диаграммы связей автоматов

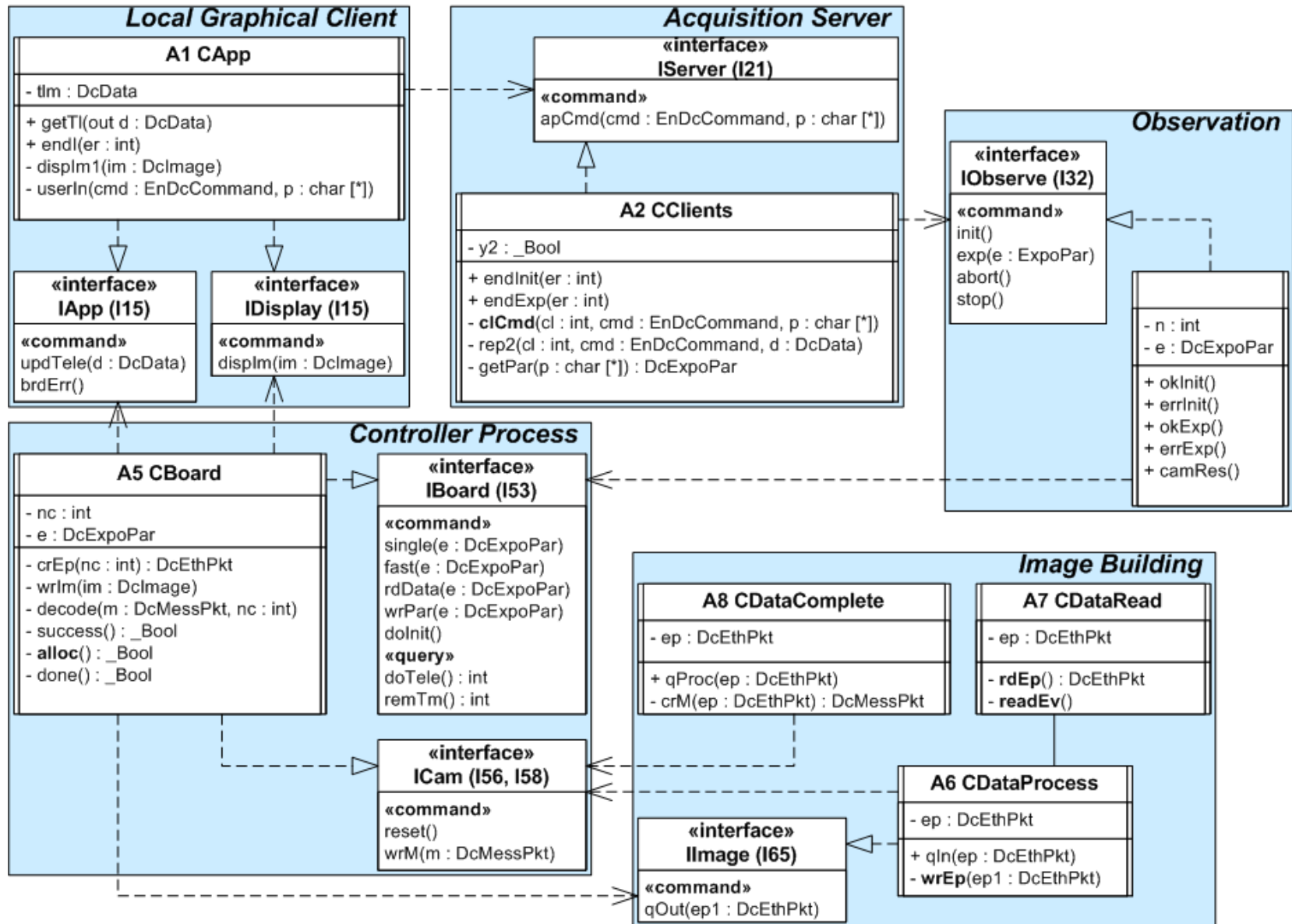
22



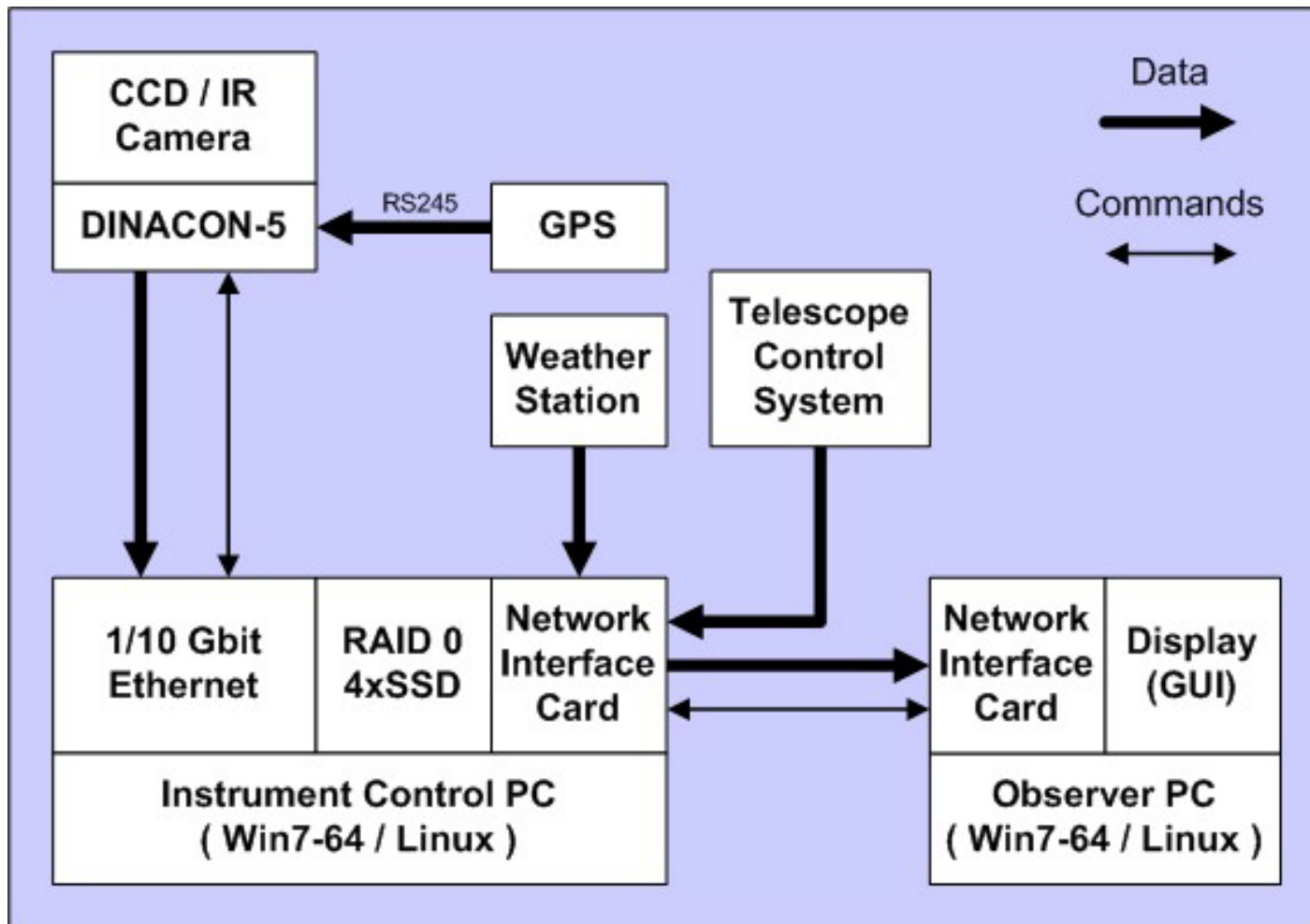
Выделение компонентов



Определение классов и интерфейсов



Результаты (1)



Возможности и преимущества ПО

- управление режимами работы ПЗС
- сохранение видеоданных
- визуализация и предварительный анализ изображений
- настройка, телеметрия и диагностика
- контроль температуры и вакуума
- многоуровневая эмуляция устройств
- быстрый просмотр/редактирование FITS-заголовка
- ведение подробного протокола команд
- интерактивный или автоматический режим наблюдений
- графический редактор временных диаграмм

Полученная архитектурная модель полностью соответствует требованиям, предъявляемым к программе сбора астрономических данных:

- протокол контроллера сконцентрирован в одном компоненте
- надежная передача данных обеспечивается распараллеливанием задач по приему и обработке
- внесение изменений не нарушает логики программы, так как вычисления полностью отделены
- реализованные шаблоны компонентов и готовые компоненты можно повторно использовать в других системах

- Применение модели взаимодействующих автоматных объектов повышает эффективность астрономических наблюдений и позволяет значительно сократить сроки разработки программного обеспечения для широкого спектра ПЗС-систем
- Реализована программа, изоморфная модели, и ставшая каркасом для программ управления новыми системами регистрации
- Методика применена при переработке программ управления ПЗС-камерой в составе спектрографа SCORPIO, и с июня 2016 г. в режиме постоянной эксплуатации на телескопе БТА продемонстрировала **0** отказов

- 1. Bock C.** Three Kinds of Behavior Models // Journal of Object-Oriented Programming. 1999. N 12 (4). P. 36-39.
- 2. Harel D.** Statecharts: a Visual Formalism for Complex Systems // Science of Computer Programming. 1987. Vol. 8. P. 231-274.
- 3. Шальто А. А.** Парадигма автоматного программирования // Научно-технический вестник СПбГУ ИТМО. 2008. Вып. 53. С. 3-24.
- 4. Поликарпова Н. И., Шальто А. А.** Автоматное программирование. – СПб.: Питер, 2011. – 176 с.
- 5. Atiskov A. Y. et al.** Ontology-Based Analysis of Cryptography Standards and Possibilities of Their Harmonization / A. Y. Atiskov, F. A. Novikov, L. N. Fedorchenko, V. I. Vorobiev, N. A. Moldovyan // Theory and Practice of Cryptography Solutions for Secure Information Systems. – Hershey: IGI Global, 2013.

- 1. Afanasieva I. V.** Data acquisition and control system for high-performance large-area CCD systems // Astrophysical Bulletin IET – 2015, Vol. 70, No. 2, pp. 232-237.
- 2. Афанасьева И.В., Новиков Ф.А.** Архитектура программного обеспечения систем оптической регистрации // Информационно-управляющие системы – 2016. – № 3 (82). – С. 51-63.
- 3. Новиков Ф.А., Афанасьева И.В.** Кооперативное взаимодействие автоматных объектов // Информационно-управляющие системы – 2016. – № 6 (85). – С. 50-64.

Участие в конференциях

1. Всероссийская астрономическая конференция «От эпохи Галилея до наших дней» ВАК-2010 (2010)
2. Международная астрономическая конференция «Настоящее и будущее малых и средних телескопов» SMT-2015 (2015)
3. Международная астрономическая конференция «Физика звезд: от коллапса до коллапса» (2016)
4. XXIV Международная научно-техническая конференция по фотоэлектронике и приборам ночного видения (2016)
5. XLVI Научная и учебно-методическая конференция Университета ИТМО (2017)
6. VIII Всероссийская научная конференция «Системный синтез и прикладная синергетика» (2017)