

# Доклад

Ротанов Денис

22 июня 2015 г.

## 1 Представление

Защищается студент группы Б8403А — Ротанов Денис Владимирович.  
Тема работы “Контролируемое исполнение программ в многозадачном окружении”.  
Руководитель — Старший преподаватель — Кленин Александр Сергеевич

## 2 Предметная область

Существует такая дисциплина как “Спортивное программирование”. В рамках этой дисциплины проводятся соревнования по решению задач на ЭВМ, для решения которых необходимо написать программу. Задача считается решённой, если программа проходит все тесты, подготовленные жюри. Такими соревнованиями являются олимпиады по программированию и АСМ ICPC. Так же существуют соревнования искусственного интеллекта.

В соревнованиях искусственного интеллекта происходит взаимодействие интеллектуальных агентов в некоторой виртуальной среде. Интеллектуальные агенты представлены программами участников, которые мы будем называть нормальными программами. Виртуальная среда предоставляется управляющей программой.

На следующих слайдах представлены скриншоты визуализаций некоторых существующих соревнований искусственного интеллекта.

Игровое соревнование жанра стратегия - двое игроков соревнуются в захвате территории.

Одно из старейших соревнований ИИ - Robocode. Множество танков соревнуются за выживание на поле, каждый танк управляется программой участника.

Одна из немногих систем предоставляющая выбор более чем одной задачи, а так же возможность писать код прямо в браузере.

Для проведения олимпиад по программированию используются автоматические системы проверки решений. На базе ДВФУ многие годы проводятся соревнования с использованием системы CATS.

Система CATS состоит из 4х компонент - базы данных, веб интерфейса, тестирующей компоненты и модуля контролируемого исполнения Spawner. Большая часть

данной работы посвящена модулю Spawner. В его задачи входит обеспечение безопасности, управление вводом/выводом и контроль за потребляемыми программой ресурсами, такими как память, время.

Рассмотрим теперь типы задач в контексте обмена данными и количества одновременно работающих решений. Можно выделить стандартные, интерактивные и многоагентные задачи.

Большинство задач представленных на олимпиадах относятся к стандартным. Такие задачи однократно считывают данные из входного файла, производят вычисления согласно условиям и выводят решение в файл, который сравнивается с эталонным решением.

В последние годы на олимпиадах так же начал появляться новый тип задач - интерактивные. В таких задачах используется игровая стратегия проверки решения. Заранее подготовленная тестирующая программа - интерактор обменивается данными с программой-решением. Интерактор на каждом ходу анализирует данные и принимает решение о продолжении тестирования.

Многоагентные задачи - это тип задач, рассматриваемый в соревнованиях искусственного интеллекта. Ключевое отличие от интерактивных в том, что **ОДНОВРЕМЕННО** происходит выполнение нескольких программ-решений и обмен данными между ними и управляющей программой, которая предоставляет им виртуальную среду.

### 3 Цель работы

Целью данной работы ставится расширение класса задач, поддерживаемых CATS, а именно добавление поддержки интерактивных и мультиагентных задач.

В рамках данной работы рассмотрение существующих решений может быть проведено в трёх областях - аналоги CATS, системы проведения соревнований искусственного интеллекта и средства контролируемого исполнения. Подавляющее большинство систем автоматической проверки решений поддерживает интерактивные задачи, но ни одна из них не поддерживает многоагентные задачи. Было рассмотрено 16 систем проведения соревнований искусственного интеллекта, данные системы узкоспециализированны - большинство систем осуществляет соревнования в пределах одной задачи. Так же не было обнаружено средств контролируемого исполнения, которые поддерживают связь стандартных потоков запускаемых процессов, а так же делегирование функций управления запускаемым процессам, что требуется для реализации многоагентных задач. Исходя из этого было принято решение о необходимости добавления поддержки интерактивных задач в CATS, а так же добавления в Spawner функционала, необходимого для поддержки многоагентных задач.

## 4 Основные проектные решения

Были рассмотрены различные варианты обеспечения обмена данными между процессами, такие как файлы, именованные пайпы, сетевое взаимодействие и как самый простой был выбран вариант обмена данными через стандартные потоки ввода-вывода, пример кода работы с которыми можно видеть на слайде. Каждый процесс в системе имеет три стандартных потока - поток ввода, поток вывода и поток вывода ошибок. В нашем случае за обмен данными будут отвечать первые два.

## 5 Проект и реализация продукта

Описание конфигурации задачи представляет собой XML файл. В данной работе в рамках добавления поддержки интерактивных задач в формат заданий добавляется тэг Run для тэга верхнего уровня Problem. Данный тэг имеет атрибут method, принимающий значения interactive или default. Что соответствует run\_method в таблице problems базы данных. Настойка cats-judge также осуществляется с помощью XML файла. В данной работе в config.xml был добавлен примитивы run\_method, содержащий шаблон командной строки запуска Spawner для интерактивных задач. Для поддержки интерактивных задач модификация Spawner не потребовалась, так как весь необходимый функционал был доступен.

На слайде представлена последовательность проверки интерактивных задач.

Так как каждый процесс имеет для обмена данными только два стандартных потока ввода-вывода, а нам требуется обеспечить связь одной управляющей программы со многими нормальными программами одновременно, то это требует мультиплексирования потоков. С этой целью был разработан протокол.

Протокол обеспечивает атомарность сообщений, сообщения должны быть разделены символом перевода строки. Протокол позволяет управляющей программе для указанной нормальной программы - отправлять ей сообщений, завершать её выполнение или ожидать сообщения. Также протокол позволяет управляющей программе определить от какой нормальной программы пришло сообщение.

Запуск в режиме многоагентных задач осуществляется через интерфейс командной строки.

На данном слайде представлена последовательность работы многоагентных задач.

Связь потоков ввода-вывода реализована с помощью двух потоков исполнения, один из потоков читает вывод процесса и записывает в свой набор буферов. Другой поток исполнения читает данные из своего набора буферов и записывает их в ввод другого процесса. Данная схема может быть расширена на любое число процессов, связывая ввод и вывод произвольных двух.

Для тестирования была разработана одна тривиальная интерактивная задача а так же адаптирована интерактивная задача из всероссийской олимпиады, а так же разработана многоагентная задача. Проводился многократный запуск Spawner на

данной многоагентной задачи для обнаружения ошибок типа race condition и deadlock. Так же проводилось тестирование сборки проекта различными компиляторами C++.

## **6 Заключение**

Итак, в результате данной работы в CATS появилась возможность использовать новый тип задач — интерактивные задачи, что выводит CATS на уровень аналогичных популярных решений. Был разработан протокол обмена данными для интерактивных многопользовательских задач и реализована его поддержка в модуле контролируемого запуска Spawner. В модуле Spawner был произведён рефакторинг, а так же исправлены и задокументированы ошибки. Дальнейшая работа в этом направлении заключается в проектировании и реализации поддержки многопользовательских интерактивных задач в другие модули системы CATS для проведения турниров ИИ. Использование нового функционала системы CATS планируется в течении летней практики в рамках учебного процесса.