

# Наумов Лев Александрович

Метод введения обобщённых координат и инструментальное средство для автоматизации проектирования программного обеспечения вычислительных экспериментов с использованием клеточных автоматов

Специальность 05.13.12.  
Системы автоматизации проектирования (приборостроение)

Научный руководитель  
доктор технических наук, профессор

**Шалыто Анатолий Абрамович**

# Актуальность

- Большинство современных вычислительных задач требует для своего решения ресурсов, превосходящих возможности однопроцессорных компьютеров.
- Возникают новые и задачи, которые нуждаются в параллельной или распределенной архитектуре вычислительной системы для своего решения.
- Так как в настоящее время при создании сложных программно-аппаратных комплексов значительная доля затрат идёт на разработку программного обеспечения и эта доля неуклонно увеличивается, задача создания инструментального средства для автоматизации проектирования программного обеспечения систем с параллельной и распределенной вычислительными архитектурами становится актуальной.
- Наглядной, общей и широко используемой моделью параллельных вычислений, которая может быть применена при создании такого средства, являются клеточные автоматы – дискретные динамические системы, поведение которых может быть полностью описано в терминах локальных зависимостей.

# Основные задачи диссертационной работы

- Разработка метода введения обобщённых координат для решеток клеточных автоматов.
- Разработка на основе метода введения обобщённых координат и компонентной модели декомпозиции клеточных автоматов инструментального средства автоматизации проектирования программного обеспечения вычислительных экспериментов.
- Демонстрация применения инструментального средства для автоматизации проектирования программного обеспечения ряда вычислительных экспериментов с использованием клеточных автоматов на различных вычислительных архитектурах.
- Классификация всех структур, которые порождаются одномерными клеточными автоматами из точечного зародыша с целью анализа простейших клеточных автоматов, демонстрирующих сложное поведение (в том числе, самовоспроизведение).

# Положения, выносимые на защиту и их научная новизна

1. Метод введения обобщённых координат для клеточных автоматов обеспечивает единообразие представления данных для автоматов с различными решетками, в том числе, и разной размерности.

До настоящего времени не существовало такого понятия и соответствующего метода.

2. Классификация всех структур, порождаемых одномерными клеточными автоматами из точечного зародыша, группирует все типы поведения, демонстрируемые простейшими клеточными автоматами (в том числе, самовоспроизведение) в классы эквивалентности.

Результат имеет значение для теории клеточных автоматов и полностью закрывает вопрос классификации таких структур.

# Практическая полезность

Практическое значение работы состоит в том, что разработано универсальное и расширяемое инструментальное средство для автоматизации проектирования программного обеспечения вычислительных экспериментов с использованием произвольных клеточных автоматов, которое позволяет проводить вычислительные эксперименты, в частности, на однопроцессорной, многопроцессорной и кластерной архитектурах.

- Не существовало средства, позволяющего использовать метод введения обобщённых координат и компонентную модель декомпозиции клеточных автоматов.
- Исследование существующих средств позволяет утверждать, что до настоящего времени не существовало средства, удовлетворяющего всем требованиям к средствам автоматизации проектирования программного обеспечения вычислительных экспериментов с помощью клеточных автоматов.
- До настоящего времени не существовало единого инструментального средства, позволяющего проводить проектирование программного обеспечения вычислительных экспериментов из широкого спектра предметных областей и для разнообразных вычислительных архитектур.

# Клеточный автомат

Клеточный автомат  $A$  представляет собой четверку объектов  $\{G, Z, N, f\}$ , где:

- $G$  – дискретное метрическое пространство, решетка автомата;
- $Z$  – конечное множество возможных состояний клетки. Таким образом, состояние всей решетки является элементом множества  $Z^{|G|}$ ;
- $N$  – конечное множество, определяющее окрестность клетки, то есть те  $|N|$  клеток, которые влияют на новое состояние данной. Элементы этого множества будем называть "соседями" клетки;
- $f$  – функция переходов или правила автомата. Вычислимая функция, действующая  $Z \times Z^{|N|} \rightarrow Z$  для автоматов с клетками с памятью и  $Z^{|N|} \rightarrow Z$  для автоматов с клетками без памяти.

Клеточные автоматы в общем случае характеризуются следующими фундаментальными свойствами:

1. Решетка однородна – никакие две области не могут быть отличены друг от друга по ландшафту. Это обеспечивается семантикой пространства  $G$ , а также единственностью определения окрестности  $N$ , множества значений  $Z$  и функции переходов  $f$ .
2. Взаимодействия локальны. Состояния только клеток окрестности влияют на состояние данной клетки, что обеспечивается семантикой функции  $f$ .
3. Обновление значений состояний всех клеток решетки происходит одновременно после вычисления нового состояния каждой из них (такие автоматы называются "синхронными").

# Обобщённые координаты

- Обобщённая координата – неотрицательное целое число, однозначно указывающее на клетку решетки клеточного автомата произвольной размерности.
- Метод введения обобщённых координат обеспечит единообразие представления данных для различных решеток.
- Он заключается в нумерации клеток неотрицательными целыми числами.
- Необходимо обеспечить нахождение непосредственных соседей данной клетки.
  - Без перехода к декартовым координатам.
  - Это позволит работать с более сложными окрестностями (искать соседей рекурсивно).

|    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|
| 20 | 21 | 26 | 27 | 32 | 33 | 74 | 75 | 80 |
| 19 | 22 | 25 | 28 | 31 | 34 | 73 | 76 | 79 |
| 18 | 23 | 24 | 29 | 30 | 35 | 72 | 77 | 78 |
| 17 | 12 | 11 | 42 | 41 | 36 | 71 | 66 | 65 |
| 16 | 13 | 10 | 43 | 40 | 37 | 70 | 67 | 64 |
| 15 | 14 | 9  | 44 | 39 | 38 | 69 | 68 | 63 |
| 2  | 3  | 8  | 45 | 50 | 51 | 56 | 57 | 62 |
| 1  | 4  | 7  | 46 | 49 | 52 | 55 | 58 | 61 |
| 0  | 5  | 6  | 47 | 48 | 53 | 54 | 59 | 60 |

# Различные способы введения обобщённых координат

Спиральные обобщённые координаты для треугольной решетки

Спиральные обобщённые координаты для квадратной решетки

Спиральные обобщённые координаты для шестиугольной решетки

Обобщённые координаты для треугольной решетки, базирующиеся на спиральных обобщённых координатах для шестиугольной решетки

# Спиральные обобщённые координаты для решетки из шестиугольников

## ■ Координаты углов

$$C_{n,m} = 6E(n-1) + 1 + m \cdot (a_n - 1) = 3(n^2 - n) + 1 + m \cdot n, \text{ где}$$

$$E(x) = 1 + 2 + \dots + (x-1) + x = \sum_{i=1}^x i = \frac{x^2 + x}{2}$$

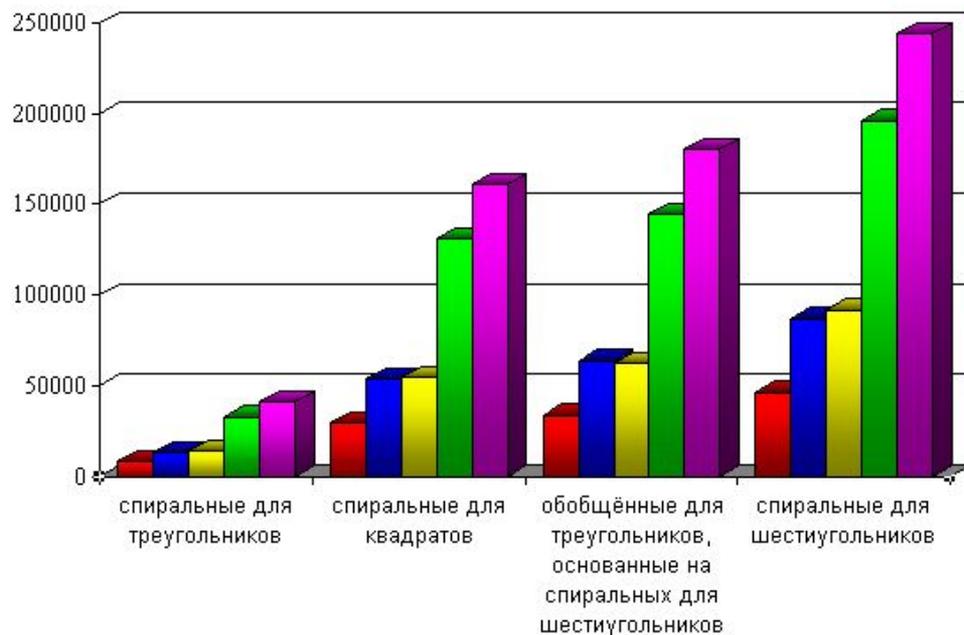
## ■ Номер кольца

$$n(a) = \left\lceil \frac{\sqrt{1 + 8 \lfloor a/6 \rfloor} - 1}{2} \right\rceil$$

## ■ Координаты соседей

| Расположение             | Индексы соседей         |                         |                         |                         |                         |                         |
|--------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
|                          | 0                       | 1                       | 2                       | 3                       | 4                       | 5                       |
| $C_{n,0}$                | $C_{n+1,0}$             | $C_{n+1,0}+1$           | $a+1$                   | $C_{n-1,0}$             | $C_{n+1,0}-1$           | $C_{n+2,0}-1$           |
| $(C_{n,0}; C_{n,1})$     | $C_{n+1,0}+a-C_{n,0}$   | $C_{n+1,0}+a-C_{n,0}+1$ | $a+1$                   | $C_{n-1,0}+a-C_{n,0}$   | $C_{n-1,0}+a-C_{n,0}-1$ | $a-1$                   |
| $C_{n,1}$                | $C_{n+1,1}-1$           | $C_{n+1,1}$             | $C_{n+1,1}+1$           | $a+1$                   | $C_{n-1,1}$             | $a-1$                   |
| $(C_{n,1}; C_{n,2})$     | $a-1$                   | $C_{n+1,1}+a-C_{n,1}$   | $C_{n+1,1}+a-C_{n,1}+1$ | $a+1$                   | $C_{n-1,1}+a-C_{n,1}$   | $C_{n-1,1}+a-C_{n,1}-1$ |
| $C_{n,2}$                | $a-1$                   | $C_{n+1,2}-1$           | $C_{n+1,2}$             | $C_{n+1,2}+1$           | $a+1$                   | $C_{n-1,2}$             |
| $(C_{n,2}; C_{n,3})$     | $C_{n-1,2}+a-C_{n,2}-1$ | $a-1$                   | $C_{n+1,2}+a-C_{n,2}$   | $C_{n+1,2}+a-C_{n,2}+1$ | $a+1$                   | $C_{n-1,2}+a-C_{n,2}$   |
| $C_{n,3}$                | $C_{n-1,3}$             | $a-1$                   | $C_{n+1,3}-1$           | $C_{n+1,3}$             | $C_{n+1,3}+1$           | $a+1$                   |
| $(C_{n,3}; C_{n,4})$     | $C_{n-1,3}+a-C_{n,3}$   | $C_{n-1,3}+a-C_{n,3}-1$ | $a-1$                   | $C_{n+1,3}+a-C_{n,3}$   | $C_{n+1,3}+a-C_{n,3}+1$ | $a+1$                   |
| $C_{n,4}$                | $a+1$                   | $C_{n-1,4}$             | $a-1$                   | $C_{n+1,4}-1$           | $C_{n+1,4}$             | $C_{n+1,4}+1$           |
| $(C_{n,4}; C_{n,5})$     | $a+1$                   | $C_{n-1,4}+a-C_{n,4}$   | $C_{n-1,4}+a-C_{n,4}-1$ | $a-1$                   | $C_{n+1,4}+a-C_{n,4}$   | $C_{n+1,4}+a-C_{n,4}+1$ |
| $C_{n,5}$                | $C_{n+1,5}+1$           | $a+1$                   | $C_{n-1,5}$             | $a-1$                   | $C_{n+1,5}-1$           | $C_{n+1,5}$             |
| $(C_{n,5}; C_{n+1,0}-1)$ | $C_{n+1,5}+a-C_{n,5}+1$ | $a+1$                   | $C_{n-1,5}+a-C_{n,5}$   | $C_{n-1,5}+a-C_{n,5}-1$ | $a-1$                   | $C_{n+1,5}+a-C_{n,5}$   |
| $C_{n+1,0}-1$            | $C_{n+2,0}-1$           | $C_{n,0}$               | $C_{n-1,0}$             | $C_{n,0}-1$             | $a-1$                   | $C_{n+2,0}-2$           |

# Сравнение эффективности различных способов введения обобщённых координат



По вертикальной оси откладывается количество клеток, для которых удастся определить соседей за одну секунду. Данные приводятся для различных способов введения обобщённых координат. Одним цветом показаны результаты, полученные на конкретной однопроцессорной системе.

Приняв **за единицу** измерения производительности средние показатели самого медленного способа – спиральных координат для треугольной решетки, можно заключить, что остальные варианты быстрее его в следующее число раз:

- спиральные координаты для квадратной решетки – в **3,9** раза;
- обобщённые координаты для треугольной решетки, основанные на спиральных координатах для шестиугольной решетки – в **4,4** раза;
- спиральные координаты для шестиугольной решетки – в **6,2** раза.

# Преимущества и недостатки обобщённых координат

## Преимущества

- Обобщённые координаты обеспечивают универсальность организации данных, позволяя хранить состояния различных решеток клеточных автоматов в одномерной структуре. При переходе от автомата с одной решеткой к автомату с другой, необходимо сменить только набор функций для определения непосредственных соседей клетки.
- Существенно более удобным становится изменение размеров решетки.
- Одномерную структуру можно рассматривать, как последовательность данных, то есть её удобно сериализовывать и сохранять.
- Обобщённые координаты – это идея, которая может быть адаптирована под конкретную задачу с учетом, например, внутренней симметрии моделируемой системы и т.п.
- Отсутствие существенной зависимости от положения нуля позволяет перемещать его в соответствии с потребностями задачи.

## Недостатки

- Скорость разрешения отношения соседства ниже, чем для декартовых координат.
- Создание математического аппарата для новых типов решёток и способов введения обобщённых координат является трудоёмким процессом

# Компонентная модель декомпозиции клеточных автоматов

Каждый клеточный автомат целесообразно представлять набором компонентов следующих четырех типов:

1. Решетка (*grid*) – осуществляет визуализацию автомата и навигацию;
2. Метрика (*metrics*) – сопоставляет клеткам координаты, вводит отношение соседства и функции вычисления расстояний между клетками.
  - Выделение этого типа компонентов позволит использовать обобщённые координаты и переходить от одних координат к только сменой компонента.
3. Хранилище данных (*datum*) – обеспечивает хранение, обмен и преобразования состояний клеток, а также некоторые аспекты визуализации;
4. Правила (*rules*) – описывает итерацию, задаёт не только функцию переходов, но и метод разделения задачи на подзадачи, оптимизацию вычислений и т.п.

При реализации целесообразно ввести дополнительный тип компонента:

5. Анализатор (*analyzer*) – позволяет наблюдать динамику изменения ключевых параметров эксперимента, строить графики, файлы отчетов и т.п.
- Необходимо отметить, что речь идет не о четырех составляющих из определения клеточного автомата ( $\{G, Z, N, f\}$ ). В данном случае декомпозицию логичнее вести по другим принципам и руководствоваться вопросам реализации автомата и проектирования вычислительного эксперимента.
  - Модель компонентной декомпозиции клеточных автоматов позволяет "собирать" автоматы с различной функциональностью, посредством незначительных изменений переходить от одной задачи к другой, пробовать решать одну и ту же задачу на разных решетках, с различными метриками и т.д.
  - Кроме того, она предоставляет возможность распределять разработку программного обеспечения эксперимента между различными исполнителями.

# Требования к инструментальным средствам (1)

1. Предоставлять пользователям дружелюбный, удобный, современный интерфейс с богатым набором возможностей для работы.
2. Наглядно визуализировать состояния решетки, обеспечивать удобную навигацию по ней.
3. Быть максимально универсальным, не накладывать ограничений на множество автоматов, которые можно использовать.
4. Выполнять итерации, используя современные вычислительные и коммуникационные технологии, быстро и эффективно распараллеливая вычисления автомата с помощью доступных ресурсов (машин вычислительного кластера, процессоров многопроцессорной системы и т.д.).
5. Оптимизировать вычисления, например, исключать не обновляемые зоны решетки из рассмотрения, использовать шаблоны для сравнения окрестностей во избежание лишних трудоемких вычислений и т. п.
6. Позволять удобно и надежно проводить продолжительные (возможно, многомесячные) вычислительные эксперименты. Несмотря на то, что почти все вычислительные ресурсы системы будут заняты осуществлением итераций, программа должна продолжать быть интерактивной и отвечать на запросы пользователя.
7. Позволять производить не только сами вычисления, но и анализ результатов вычислений, а именно, поддерживать построение разнообразных графиков, демонстрировать изменения различных

| Название                   | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----------------------------|---|---|---|---|---|---|---|
| <i>CAGE</i>                | + | + | - | - | - | + | - |
| <i>CAM Simulator</i>       | - | + | - | - | - | + | - |
| <i>CAMEL</i>               | - | - | + | + | - | + | - |
| <i>CANL</i>                | + | - | - | + | - | + | - |
| <i>CAPow</i>               | + | + | - | - | - | + | - |
| <i>CASim</i>               | - | - | - | - | - | + | - |
| <i>CAT/CARP</i>            | - | - | - | - | - | + | - |
| <i>CDL</i>                 | - | - | - | + | - | + | - |
| <i>CDM/SLANG</i>           | - | - | - | - | - | + | - |
| <i>CellLab</i>             | - | - | - | + | - | + | - |
| <i>CELLAS/FUNDEF</i>       | - | - | - | - | - | + | - |
| <i>Cellsim</i>             | + | + | - | - | - | + | - |
| <i>Cellular/Cellang</i>    | + | - | + | - | - | + | - |
| <i>CEPROL</i>              | - | - | - | - | - | + | - |
| <i>DDLab</i>               | + | + | - | - | - | + | - |
| <i>HICAL</i>               | + | - | - | - | - | + | + |
| <i>LCAU</i>                | - | + | - | - | - | + | + |
| <i>Mirek's Celebration</i> | + | + | - | - | - | + | - |
| <i>SCARLET</i>             | + | + | - | - | - | + | - |
| <i>SIMP/STEP</i>           | - | - | - | + | + | + | - |
| <i>WinCA</i>               | + | + | - | - | - | - | - |
| <i>CAME&amp;L</i>          | + | + | + | + | + | + | + |

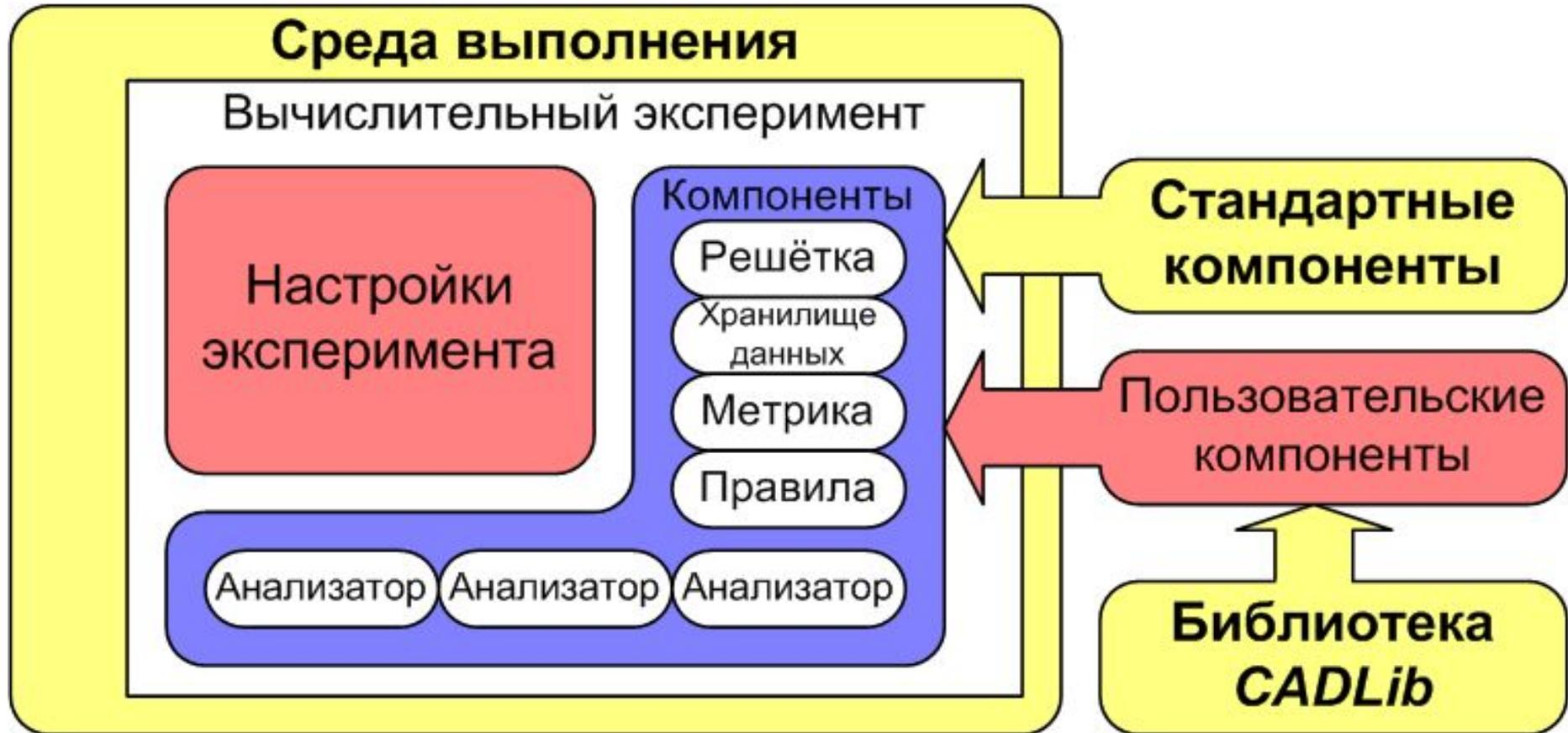
# Требования к инструментальным средствам (2)

1. Предоставлять пользователям дружелюбный, удобный, современный интерфейс с богатым набором возможностей для работы.
2. Наглядно визуализировать состояния решетки, обеспечивать удобную навигацию по ней.
3. Быть максимально универсальным, не накладывать ограничений на множество автоматов, которые можно использовать.
4. Выполнять итерации, используя современные вычислительные и коммуникационные технологии, быстро и эффективно распараллеливая вычисления автомата с помощью доступных ресурсов (машин вычислительного кластера, процессоров многопроцессорной системы и т.д.).
5. Оптимизировать вычисления, например, исключать не обновляемые зоны решетки из рассмотрения, использовать шаблоны для сравнения окрестностей во избежание лишних трудоемких вычислений и т.п.
6. Позволять удобно и надежно проводить продолжительные (возможно, многомесячные) вычислительные эксперименты. Несмотря на то, что почти все вычислительные ресурсы системы будут заняты осуществлением итераций, программа должна продолжать быть интерактивной и отвечать на запросы пользователя.
7. Позволять производить не только сами вычисления, но и анализ результатов вычислений, а именно, поддерживать построение разнообразных графиков, демонстрировать изменения различных

1. Интерфейс предложенного инструментального средства чрезвычайно богат и поддерживает множество типичных функций, присутствия которых пользователь в праве ожидать от современного программного обеспечения.
2. За визуализацию состояния решетки и удобную навигацию по ней отвечает отдельный класс компонентов. Пользователю предоставляется стандартный набор средств для этих целей, кроме того, он может быть расширен с помощью новых компонентов, обладающих конкретными специфическими свойствами, необходимыми для решения задач.
3. Инструментальное средство позволяет моделировать класс систем даже более широкий, чем класс "клеточные автоматы".
4. Средство позволяет организовывать вычисления, как на отдельном персональном компьютере, так и на многопроцессорной системе и в вычислительном кластере.
5. Средства для оптимизации вычислений включены в стандартный пакет программного обеспечения и могут быть применены при решении пользовательских задач. Этот набор также может быть легко расширен пользователями при необходимости.
6. Проект позволяет удобно и надежно проводить продолжительные вычислительные эксперименты, сохраняя при этом интерактивность и предоставляя возможности управления его течением.
7. Инструментальное средство использует специальные компоненты-анализаторы для наблюдений и изучения динамики ключевых параметров эксперимента, построения графиков, файлов-отчетов и прочих задач такого рода.

# Структурная схема инструментального средства

*CAME&L – Cellular Automata Modeling Environment & Library*



*CAME&L* = среда выполнения + библиотека *CADLib* + стандартные компоненты

# Методика организации и проведения вычислительных экспериментов

## Запуск среды выполнения.

- При проведении эксперимента в вычислительном кластере, необходимо запустить среду на всех вычислительных узлах.
- 2. **Выбор решетки.** Если подходящая решетка есть в наборе стандартных компонентов, то выбрать ее. Иначе – создать новый компонент.
- 3. **Выбор метрики.** Если подходящая метрика есть в наборе стандартных компонентов, то выбрать ее. Иначе – создать новый компонент.
- 4. **Выбор хранилища данных.** Если подходящее хранилище данных есть в наборе стандартных компонентов, то выбрать его. Иначе – создать новый компонент.
- 5. **Выбор правил.** Если подходящие правила есть в наборе стандартных компонентов, то выбрать их. Иначе – создать новый компонент.
- 6. **Настройка компонентов.** Установить требуемые значения параметров.
- 7. **Запуск эксперимента** путем выбора соответствующей команды в среде.

## Создание и выбор нового компонента

1. **Разработка компонента.** С помощью привычного средства разработки (например, *Microsoft Visual C++ 6*) создать библиотеку, содержащую класс требуемого компонента, являющийся наследником класса базового класса из библиотеки *CADLib*.
2. **Компиляция компонента** в *DLL*-библиотеку.
3. **Установка компонента в инструментальном средстве.** Для установки, удаления и изучения компонентов в среде выполнения имеется инструмент "*Components Manager*". Им и необходимо воспользоваться.
4. **Выбор компонента.**

# Применение методики на примере игры "Жизнь"

1. **Выбор решетки.** Для моделирования требуется двумерная квадратная решетка. Компонент, "*Square Basic Grid*", реализующий функциональность такой решетки, имеется в наборе стандартных компонентов.
2. **Выбор метрики.** Используем декартовы координаты. Компонент "*Cartesians*", реализующий функциональность декартовой метрики, существует в наборе стандартных.
3. **Выбор хранилища данных.** Компонент "*Booleans for Cartesians*", реализующий функциональность хранилища булевых данных для декартовой метрики, имеется в наборе стандартных.
4. **Выбор правил.** Предположим, что компонент, реализующий правила игры "Жизнь" не входит в набор стандартных компонентов. Создадим новый компонент.
5. **Настройка компонентов.** Необходимо изменить значение параметра "*Double*" хранилища данных "*Booleans for Cartesians*" на "*true*", выбрав тем самым синхронный клеточный автомат.
6. **Запуск эксперимента.**

## Создание и выбор нового компонента

### 1. Разработка компонента.

Создадим класс наследник класса CAUniRules. Переопределим его функцию SubCompute следующим образом:

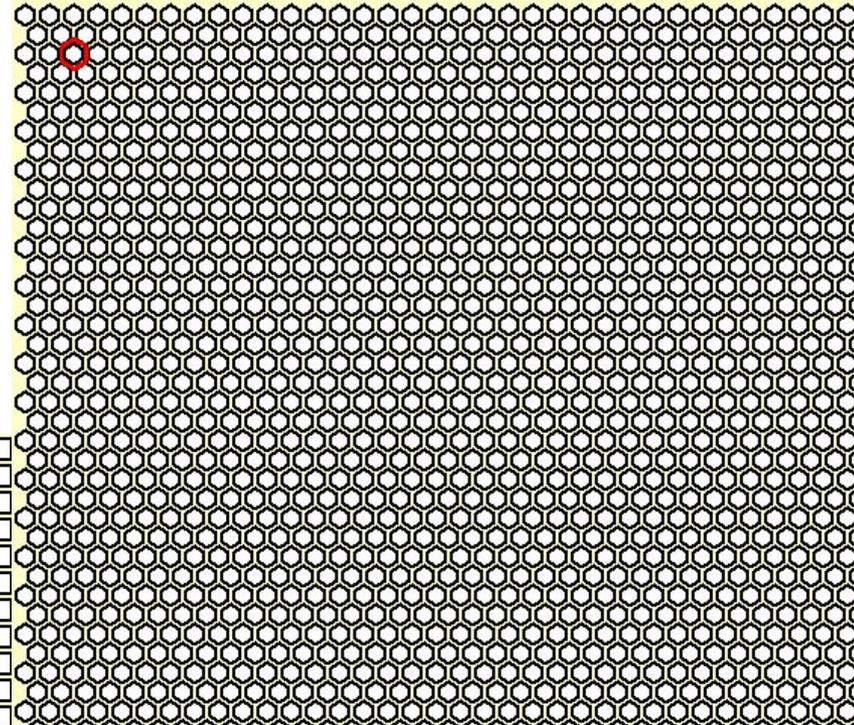
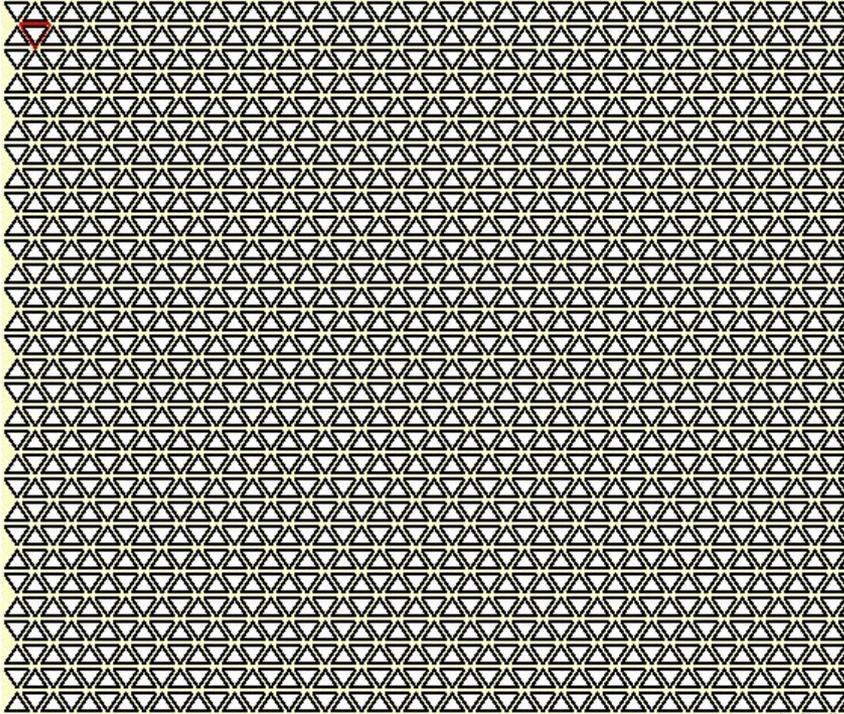
```
DATUM(CACrtsBool2DDatum);
CACell c;
CACell neig[12];
int alive=0;
for(CACell i=z.a1; i<=z.b1; i++)
    for(CACell j=z.a2; j<=z.b2; j++) {
        alive=0;
        c=GET_METRICS->ToCell(i,j,0);
        GET_METRICS->GetNeighbours(c,neig);
        for(unsigned int k=0;
            k<GET_METRICS->GetNeighboursCount();
            k++) if(datum->Get(neig[k]))alive++;
        if(datum->Get(c))
            datum->Set(c,alive==2||alive==3);
        else
            datum->Set(c,alive==3);
    }
return true;
```

2. **Компиляция компонента** в DLL-библиотеку.

3. **Установка компонента** в инструментальном средстве.

4. **Выбор компонента.**

# Уравнение теплопроводности

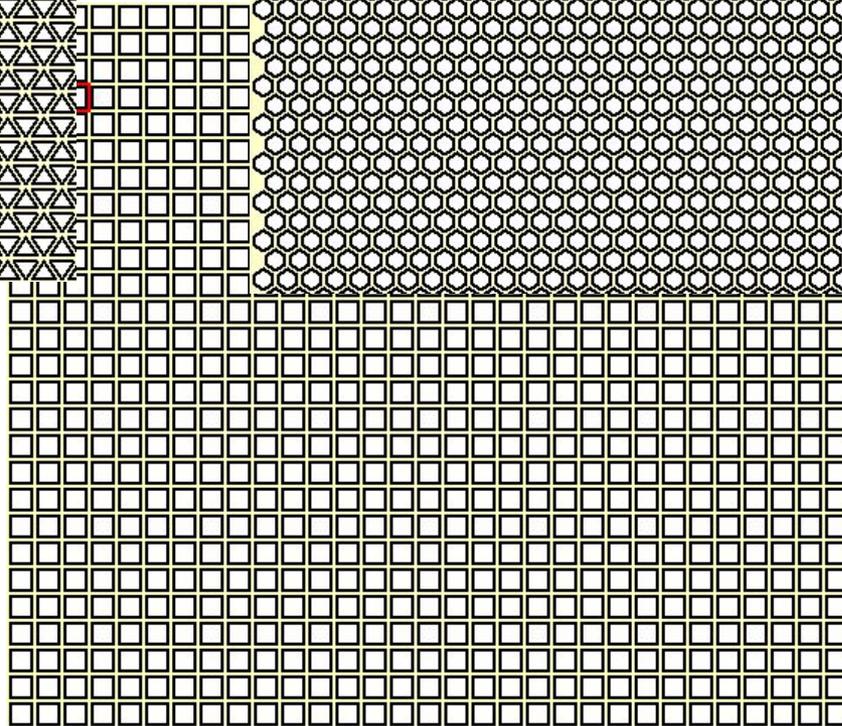


Непрерывный вариант:

$$\frac{\partial T}{\partial t} = a \cdot \Delta T$$

Дискретный вариант (при  $a$ ,  $dx$  и  $dy = 1$ ):

$$T'(x, y) = \frac{T(x+1, y) + T(x-1, y) + T(x, y+1) + T(x, y-1)}{4}$$



# Классификация структур, порождаемых простейшими клеточными автоматами из точечного зародыша (1)

- Существует 256 функций переходов для простейших одномерных двоичных клеточных автоматов.
- Рассмотрены и пронумерованы 256 структур, порождаемых во времени с помощью этих функций из точечного зародыша.
- Наиболее интересны те из них, которые обладают свойством самовоспроизведения:

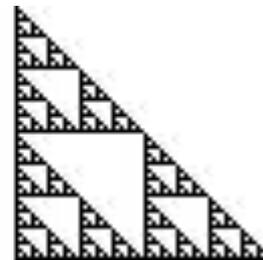
| $c_{i-1}$ | $c_i$ | $c_{i+1}$ | $c'_i$ |        |
|-----------|-------|-----------|--------|--------|
| 0         | 0     | 0         | 0      | $*2^0$ |
| 0         | 0     | 1         | 1      | $*2^1$ |
| 0         | 1     | 0         | 0      | $*2^2$ |
| 0         | 1     | 1         | 1      | $*2^3$ |
| 1         | 0     | 1         | 1      | $*2^4$ |
| 1         | 0     | 0         | 0      | $*2^5$ |
| 1         | 1     | 1         | 1      | $*2^6$ |
| 1         | 1     | 0         | 0      | $*2^7$ |
|           |       |           |        | = 90   |

```

0000000010000000
000000101000000
000001000100000
000010101010000
000100000001000
001010000010100
010001000100010
101010101010101
    
```



Структура № 90. Салфетка Серпинского



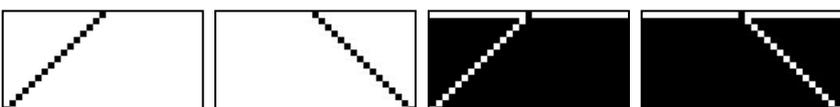
Структура № 60. Треугольник Паскаля



Структура № 22. Салфетка Серпинского из "пьедесталов"

# Классификация структур, порождаемых простейшими клеточными автоматами из точечного зародыша (2)

Впервые проведена полная классификация этих структур относительно различных операций инвариантности.

- $E$  
- $EM$  
- $EI$  
- $E(I+M)$  
- $EIM$  
- $EIMO$  
- $EIMOL6$  

# Классификация структур, порождаемых простейшими клеточными автоматами из точечного зародыша (3)

Число классов при разных вариантах классификации без сдвигов

|               | <b>L0</b> | <b>L1</b> | <b>L2</b> | <b>L3</b> | <b>L4</b> | <b>L5</b> | <b>L6</b> |
|---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <b>E</b>      | 143       | 134       | 129       | 128       | 128       | 128       | 127       |
| <b>EM</b>     | 89        | 83        | 81        | 80        | 80        | 80        | 79        |
| <b>EI</b>     | 135       | 124       | 120       | 119       | 119       | 119       | 118       |
| <b>E(I+M)</b> | 135       | 124       | 120       | 119       | 119       | 119       | 118       |
| <b>EIM</b>    | 83        | 76        | 74        | 73        | 73        | 73        | 72        |

Число классов при разных вариантах классификации со сдвигами

|               | <b>OL0</b> | <b>OL1</b> | <b>OL2</b> | <b>OL3</b> | <b>OL4</b> | <b>OL5</b> | <b>OL6</b> |
|---------------|------------|------------|------------|------------|------------|------------|------------|
| <b>E</b>      | 125        | 118        | 118        | 117        | 117        | 117        | 117        |
| <b>EM</b>     | 78         | 75         | 75         | 74         | 74         | 74         | 74         |
| <b>EI</b>     | 88         | 86         | 86         | 85         | 85         | 85         | 85         |
| <b>E(I+M)</b> | 88         | 85         | 85         | 84         | 84         | 84         | 84         |
| <b>EIM</b>    | 56         | 55         | 55         | 54         | 54         | 54         | 54         |

# Внедрение

## Научные исследования



В СПбГУ ИТМО при исследованиях по грантам Министерства образования и науки РФ, а также РФФИ.



В Университете Амстердама для организации вычислительных экспериментов на *FPGA*-системах и многих других задач.



Политехническом университете Бухареста при создании инструментального средства для обработки изображений.

## Обучение



В СПбГУ ИТМО на кафедре "Компьютерные технологии":

- при чтении лекций по курсу "Теория автоматов в программировании";
- в курсовых работах студентов по курсу "Теория автоматов в программировании".



В СПбГУ на кафедре "Вычислительная физика":

- при чтении лекций по курсу "Современные технологии программирования для научных работников";
- для демонстрации решения уравнений вычислительной физики посредством клеточных автоматов.

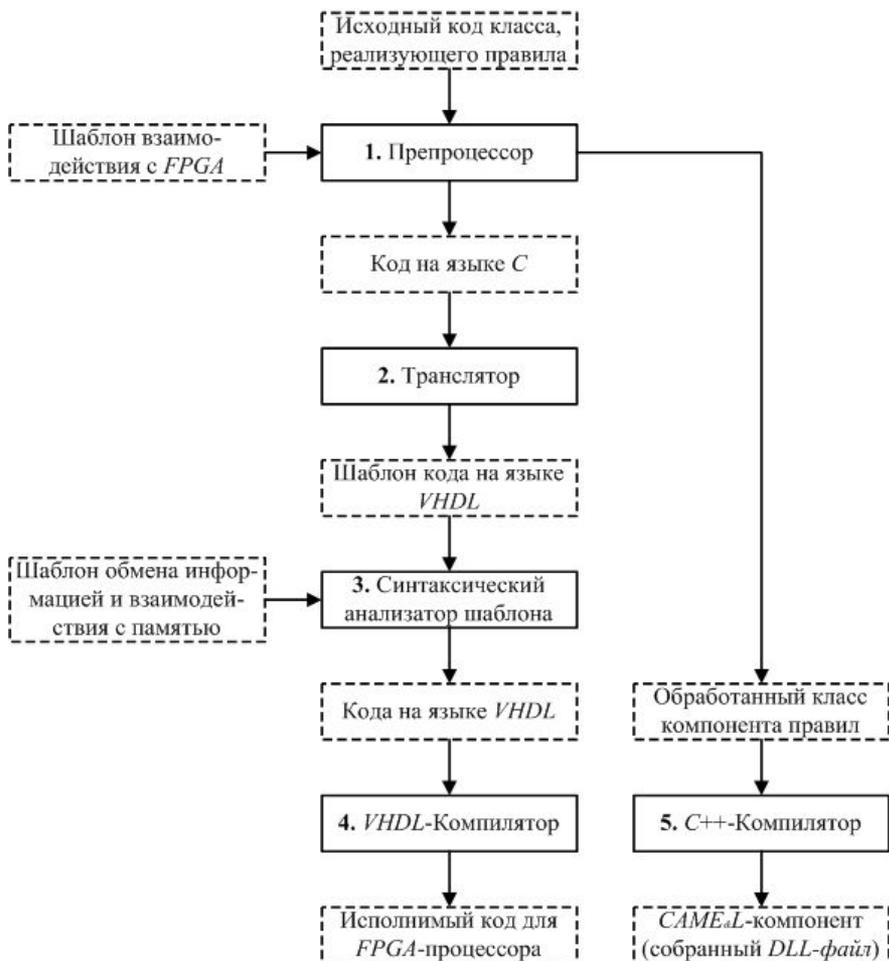


В Университете Амстердама:

- при чтении лекций по курсу "Моделирование поведения сложных систем";
- написании дипломных проектов.

# Использование инструментального средства для организации вычислительных экспериментов на *FPGA*-системах

- Разработанный комплекс средств позволяет на основе имеющегося исходного кода компонента правил автоматически создать управляющий компонент правил (содержащий, в основном, коммуникационные функции) и модуль, выполнимый на *FPGA*-процессоре (содержащий вычислительное ядро).
- При запуске эксперимента с использованием сгенерированного управляющего компонента, модуль загружается в *FPGA*-систему и начинает выполнять вычислительный эксперимент под контролем управляющего модуля, обмениваясь с ним информацией.
- Имеется возможность загрузить состояние решетки клеточного автомата из *FPGA*-процессора в среду выполнения и обратно.
- В результате, для пользователя наблюдающего эксперимент с помощью среды выполнения, он будет неотличим от выполняемого на однопроцессорной, многопроцессорной или кластерной архитектуре.



# Заключение

В диссертации получены следующие результаты:

- Введено понятие "обобщенные координаты" для решеток клеточных автоматов. Разработан метод введения таких координат.
- На основе метода введения обобщенных координат и компонентной модели декомпозиции разработано инструментальное средство автоматизации проектирования программного обеспечения вычислительных экспериментов с использованием клеточных автоматов *CAME&L*.
- Метод и инструментальное средство *CAME&L* апробированы при автоматизации проектирования программного обеспечения ряда вычислительных экспериментов с использованием клеточных автоматов на однопроцессорных, многопроцессорных, кластерных и *FPGA* вычислительных системах. Ранее было невозможно выполнить это с помощью одного и того же инструментального средства.
- Проведена классификация всех возможных структур, порождаемых одномерными клеточными автоматами из точечного зародыша. Построено множество разнообразных классов эквивалентности. Показано, что автоматы с клетками с памятью и автоматы с клетками без памяти могут демонстрировать одинаковое поведение. Таким образом, выполнен анализ простейших клеточных автоматов, порождающих сложное поведение (в том числе, самовоспроизведение).
- Результаты были внедрены в учебном процессе и при проведении научных исследований в ряде университетов.

# Публикации

1. *Наумов Л.* Как увеличить скорость "Жизни", или Эффективная организация данных для повышения скорости поиска клеток и разрешения отношений соседства при реализации клеточного автомата Джона Хортон Конвея "Жизнь". Части I и II // Информатика. 2001. № 33 (322). с. 25–27; № 34 (323). с. 20–24; Компьютеры + Программы. 2001. № 10. с. 68–73.
2. *Naumov L.* Generalized Coordinates for Cellular Automata Grids / Computational Science – ICCS 2003. Part 2. Springer-Verlag. 2003, с. 869–878.
3. *Наумов Л., Шалыто А.* Клеточные автоматы. Реализация и эксперименты // Мир ПК. 2003. № 8, с. 71–78.
4. *Наумов Л.* CAMEL – среда моделирования и библиотека разработчика клеточных автоматов / Труды XI Всероссийской научно-методической конференции Телематика'2004. Том 1. СПб.: СПбГУ ИТМО. 2004, с. 184–186.
5. *Наумов Л.* CTP (Commands Transfer Protocol) – Сетевой протокол для высокопроизводительных вычислений / Труды XI Всероссийской научно-методической конференции Телематика'2004. Том 1. СПб.: СПбГУ ИТМО. 2004, с. 188–189.
6. *Наумов Л., Шалыто А.* "Цветные" клеточные автоматы // Мир ПК. 2004. № 5, с. 64–71.
7. *Naumov L.* CAMEL – Cellular Automata Modeling Environment & Library / Cellular Automata. Sixth International Conference on Cellular Automata for Research and Industry, ACRI-2004. Springer-Verlag. 2004, с. 735–744.
8. *Наумов Л.* CTP (Commands Transfer Protocol) v. 1.2 – Новая версия сетевого протокола для высокопроизводительных вычислений / Труды XII Всероссийской научно-методической конференции Телематика'2005. Том 1. СПб.: СПбГУ ИТМО. 2005, с. 92–93.
9. *Наумов Л.* Преимущества использования обобщённых координат для многомерных решёток клеточных автоматов / Труды XII Всероссийской научно-методической конференции Телематика'2005. Том 1. СПб.: СПбГУ ИТМО. 2005. с. 93–95.
10. *Наумов Л.* CAMEL – средство для осуществления параллельных и распределённых вычисления на основе клеточных автоматов / Технологии распределённых вычислений. 2005, с. 284–286.
11. *Наумов Л.* Решение задач с помощью клеточных автоматов посредством программного обеспечения CAMEL (Части I и II) // Информационно-управляющие системы. 2005. № 5, с. 22–30; № 6, с. 30–38.
12. *Наумов Л., Шалыто А.* Классификация структур, порождаемых одномерными двоичными клеточными автоматами из точечного зародыша // Известия РАН. Теория и системы управления. 2005. № 5, с. 137–145. **Журнал из списка ВАК.**  
*Naumov L., Shalyto A.* Classification of Structures Generated by One-Dimensional Binary Cellular Automata from a Point Embryo // Journal of Computer and Systems Sciences International. Vol. 44. 2005. № 5, pp. 800–807.
13. *Наумов Л.* Сравнение специализированных сетевых протоколов CTP (Commands Transfer Protocol) и IL (Internet Link) / Труды XIII Всероссийской научно-методической конференции Телематика'2006. Том 1. СПб.: СПбГУ ИТМО. 2006, с. 266–267.
14. *Наумов Л.* Обзор программного обеспечения для решения задач с использованием клеточных автоматов // Телекоммуникации и информатизация образования. 2006. № 2, с. 114–125.

**Все статьи и инструментальное средство – опубликованы на сайте**  
<http://camellab.spb.ru>.

# Конференции, НИРы, гранты

## Конференции

1. Международная научная конференция "*International Conference of Computational Sciences*" (Санкт-Петербург – Мельбурн, 2003 г.).
2. Международная научная конференция "*Cellular Automata for Research and Industry*" (Амстердам, 2004 г.).
3. Научная школа "*Joint Advanced Students School*" (Санкт-Петербург, 2004 г.).
4. Научно-методические конференции "Телематика-2004", "Телематика-2005", "Телематика-2006" (Санкт-Петербург).
5. Научная школа "*Ferienakademie*" (Южный Тироль, 2005 г.).
6. Всероссийская научная конференция "Научный сервис в сети Интернет: технологии распределенных вычислений" (Дюрсо, 2005 г.).
7. Международная научная конференция "Высокопроизводительные параллельные вычисления на кластерных системах" (Санкт-Петербург, 2006 г.).

## Научно-исследовательские работы, гранты

1. "Разработка технологии автоматного программирования", выполненная по гранту РФФИ № 02-07-90114 в 2002-2003 годах.
2. "Разработка технологии программного обеспечения систем управления на основе автоматного подхода", выполняемая по заказу Министерства образования РФ в 2002-2005 годах.
3. Персональный грантов для студентов и аспирантов вузов Санкт-Петербурга в 2004 году.
4. "Разработка среды и библиотеки "*CAME&L*" для организации параллельных и распределенных вычислений на основе клеточных автоматов", выполненной по гранту РФФИ № 05-07-90086 в 2005-2006 годах. **При выполнении работ по этому гранту автор был ответственным исполнителем.**

## Награды

- Научно-исследовательская деятельность автора дважды отмечена стипендией Президента РФ.