

МУЛЬТИ-АГЕНТНОЕ МОДЕЛИРОВАНИЕ СОЦИАЛЬНЫХ ПРОЦЕССОВ В *SWARM*

Ю.В. Фролова

In the paper the Swarm software for Multi-Agent Simulation is described

1. Введение

Компьютерные модели имеют много преимуществ по сравнению с другими подходами к исследованию социальных процессов. В частности, они дают возможность учитывать большое количество переменных, предсказывать развитие нелинейных процессов. Кроме того, они позволяют не только получить прогноз (он ведь нужен не сам по себе, а как информация для принятия решений), но и найти с помощью вычислительных экспериментов, какие управляющие воздействия приведут к наиболее благоприятному развитию событий. Таким образом, компьютерные модели социальных процессов и вычислительный эксперимент с этими моделями являются важным средством управления социальными процессами [2].

Под компьютерной моделью понимают [4]:

— условный образ объекта или некоторой системы объектов (или процессов), описанный с помощью взаимосвязанных компьютерных таблиц, блок-схем, диаграмм, графиков, рисунков, анимационных фрагментов, гипертекстов и т.д. и отображающий структуру и взаимосвязи между элементами объекта. Компьютерные модели такого вида называют структурно-функциональными;

— отдельную программу, совокупность программ, программный комплекс, позволяющий с помощью последовательности вычислений и графического отображения их результатов воспроизводить (имитировать) процессы функционирования объекта, системы объектов при условии воздействия на объект различных, как правило случайных, факторов. Такие модели называют имитационными.

Выделяются следующие функции моделей [1]:

— познавательная (дает возможность заглянуть в суть изучаемых явлений, лучше понять их);

© 2000 Ю.В. Фролова

E-mail: frolova@univer.omsk.su

Омский государственный университет

- прогнозирование;
- принятие решений (социальное планирование и управление социальными процессами);
- совершенствование измерения.

Моделирование представляет возможность нового мышления относительно социальных процессов, основанных на идеях о появлении сложного поведения из сравнительно простых действий [3].

Компьютерное моделирование — метод решения задачи анализа или синтеза сложной системы на основе использования ее компьютерной модели. Суть компьютерного моделирования заключена в получении количественных и качественных результатов по имеющейся модели. Качественные выводы, получаемые по результатам анализа, позволяют обнаружить неизвестные ранее свойства сложной системы: ее структуру, динамику развития, устойчивость, целостность и др. Количественные выводы в основном носят характер прогноза некоторых будущих или объяснения прошлых значений переменных, характеризующих систему [4]. Компьютерное моделирование является существенным средством для изучения глобального и адаптивного поведения, которое возникает в сложных системах.

Существует множество различных подходов к моделированию и пониманию социальных процессов. Развитие современных подходов к моделированию в социальных науках [3]:

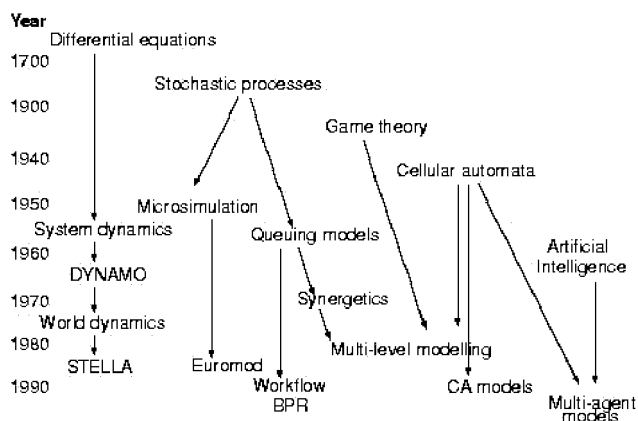


Рис.1

Остановимся на одном из них — мульти-агентном моделировании, используя специально разработанное инструментальное программное средство для построения мульти-агентных моделей *Swarm*.

2. Мульти-агентная система моделирования *Swarm*

Swarm - универсальный пакет для моделирования параллельно распределенного искусственного мира [7]. Главная часть *Swarm* это множество библиотек, располагающихся в исходной директории. Кратко, директория организована следующим образом:

- **lib, include, bin.** Установленные библиотеки, включающие файлы.
- **scr.** Swarm библиотека исходных файлов.

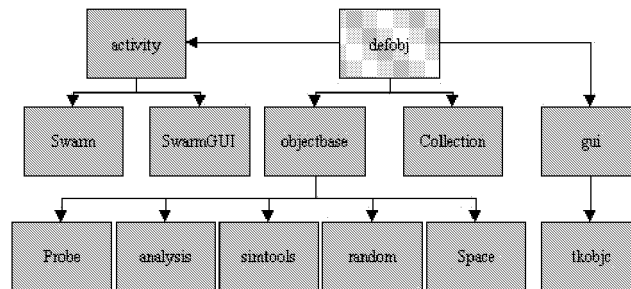


Рис.2

- **defobj.** Поддержка стиля Swarm Objective-C программированием;
- **Collection.** Коллекционная библиотека – различные объекты, которые «коллекционируют» другие объекты (Arrays, Lists и упорядоченные Lists объектов);
- **activity.** Спецификация выполнения механизма – «процесс» (Schedules, ActivityGroups и Swarms);
- **objectbase.** Базовый класс для имитированных объектов – «агенты»;
- **Space.** Библиотека пространственной среды (Управляет информацией для 2D решетки или сеток, в настоящее время главным образом поддерживаются 2-мерные пространства);
- **random.** Библиотека случайного числа (Генераторы и распределения произвольного числа);
- **simtools.** Собранные инструменты полезны для развития Swarm имитаций (Набор различных инструментальных средств);
- **gui.** Библиотека графического интерфейса;
- **tkobjc.** Объекты графического интерфейса;
- **analysis.** Объекты, которые помогают с обработкой данных (Более легкий интерфейс для графов, вычисления статистики);
- **Probe.** Расследование состояния и поведения объектов.

Swarm является системой, которая позволяет пользователям разрабатывать модели, включая множество агентов, взаимодействующих друг с другом в пределах некоторой среды. Цель Swarm состоит в том, чтобы уменьшить работу исследователям для построения и имитации искусственного мира и обеспечить стандартизованными инструментальными средствами для управления и анализа результатов моделирования. Ядро Swarm является объектно-ориентированным каркасом для определения поведения агентов и других объектов, которые взаимодействуют в моделировании. Термин «агент» относится к определяемому пользователем объекту в моделировании, который представляется фактической частью модели; объект относится к любому вычислительному объекту в системе, который может не быть частью фактически изучаемой модели (например, объект анализа, собирающий данные). Агенты обычно имеют следующие свойства [3]:

- автономность – агенты действуют самостоятельно, имея прямое управление

их действиями и внутренним состоянием;

— социальная способность – агенты взаимодействуют с другими агентами посредством некоторого «языка»;

— реактивность – агенты способны чувствовать свою среду (которая может быть физическим миром, виртуальным миром электронных сетей, или имитационным миром, включающим других агентов) и отвечать ей;

— действенность – реагируя на свою среду, агенты также способны взять инициативу, имея целеустремленное поведение.

Агент – просто более определенный термин, предназначенный для разъяснения структуры моделирования.

Моделируя процессы, которые происходят через множество параллельно взаимодействующих агентов в Swarm, исследователь, таким образом, моделирует параллельные процессы, которые происходят в некотором реальном или искусственном мире. Агенты связываются друг с другом и со средой посредством сообщения. Swarm содержит системную библиотеку, которая управляет динамическим списком объектов и оперирует сообщениями, проходящими между объектами. Каждый объект содержит некоторые локальные данные, а также спецификацию для них, чтобы среагировать на сообщения, которые посланы ему. Swarm также обеспечивает интерфейс пользователя и инструментальные средства анализа. Основное предположение в пределах данного прототипа Swarm – понятие времени. Текущая система использует дискретное время, планируя алгоритм всякий раз, когда каждый объект принимает сообщение «шаг», направляющие агента выполнить некоторое действие.

Объект в Swarm имеет три основных характеристики: Имя, Данные, и Правила. Имя объекта состоит из уникальной идентификации (ID), которая используется, чтобы послать сообщения на объект, типа (в текущей реализации это символьная строка), и модульного имени (какое, вместе с типом, может быть продуманным как имя класса). Данные являются локальными данными, которые пользователь хочет иметь на агента (например, внутренние состояние переменных). Правила – набор функций, которые обрабатывают любые сообщения, которые посланы объекту, включая сообщение «шага».

Абстрактная модель Swarm – параллельно объектная система, в которой все объекты могут быть непрерывно активными. Объекты Swarm организовываются в иерархию subswarms. Каждый swarm является совокупностью агентов и связанных объектов, а также включает планирование, рассматривающее действия для своих агентов, которые должны быть выполнены [7].

3. Heatbugs – программа демонстрации Swarm

Программа heatbugs — простая демонстрационная модель Swarm, пример того, как простые агенты, действующие только на локальной информации, могут произвести сложное глобальное поведение.

Представьте себе, что Вы хотите смоделировать жуков, которые: — испускают тепло (или некоторый другой способный рассеиваться фактор) в свою окружающую среду, по мере того, как они двигаются;

— имеют встроенную «идеальную температуру», в которую они должны предпочесть погружаться (то есть, они должны предпочесть иметь среду вокруг себя с этой температурой);

— имеют сенсор, который позволяет им чувствовать сумму тепла, исходящего от их местной окружающей среды в различных направлениях;

— имеют механизм, который позволяет им вычислять, в каком направлении окружающая температура ближе к их идеальной температуре;

— имеют механизм, который позволяет им перемещаться (в конечном темпе), например, в направлении по отношению к желаемой температуре.

Вы хотите смоделировать разнородных жуков, при этом каждый жук имеет:

— собственную идеальную температуру (из некоторого диапазона значений);

— степень распространения собственного тепла (из некоторого диапазона значений).

Окружающая среда, в которой живут жуки, имеет две основные характеристики:

— тепло рассеивается одинаково во всех направлениях, в норме, пропорционально разнице тепла между точками;

— тепло испаряется в норме, пропорционально сумме тепла.

Поведение жуков находится под влиянием их местной среды, в которую они пытаются переместиться по отношению к температуре, которая ближе к их идеальной температуре. Но также, жуки «изменяют» свою окружающую среду и среду других жуков вокруг себя. Например, жук может находиться в точке, имеющей температуру, которая соответствует его идеальной температуре, но только находясь там, жук сам изменит окружающую среду, поскольку он выделяет тепло, другие жуки могут переместиться в область, следуя своим собственным тепло-механизмам, и снова изменять среду для однажды счастливого жука.

Предполагается:

1) N жуков живут на 2-мерной квадратной решетке, размером $X \times Y$. Не более чем один жук может занять ячейку в одно и то же время;

2) Время дискретно. На каждом шаге происходит множество изменений, то есть жуки перемещаются, тепло рассеивается и испаряется;

3) Каждая ячейка в мире в настоящий момент имеет конкретное количество тепла. Тепло рассеивается синхронно между более соседними (8) ячейками, и тепло испаряется из каждой ячейки. В частности, правило обновления тепла в ячейке: $H(t+1) = e * (H(t) + k(AvgH(t) - H(t)))$, где e – степень испарения (0..1); k – степень распространения (0..1); $AvgH$ – среднее число H в 8 соседних ячейках.

Также предполагается, что каждый жук:

1) создается с фиксированной идеальной температурой (*idealTemperature*), выбранной произвольно из фиксированного диапазона;

2) создается с фиксированным выходным теплом (*outputHeat*), выбранным произвольно из фиксированного диапазона;

3) получает самое большое один шанс для того, чтобы переместиться в каждый

шаг времени. Он перемещается в соседа, который имеет температуру, самую близкую к его собственной идеальной температуре. Тем не менее, каждый жук также имеет вероятность того, что он перемещается в произвольного соседа, вместо «рационального» перемещения;

4) выдает тепло в ячейку, которую он занимает, за каждый шаг времени;

5) имеет «недовольство» (*unhappiness*), оценивая свою собственную ситуацию, то есть функцию тепла в ячейке, которую он занимает, и идеальную температуру. В частности: $U = \text{abs}(H - I) / \text{max}(H)$, где U – недовольство (*unhappiness*), H – тепло в настоящее время в ячейке, которую он занимает, I – идеальная температура агента, и $\text{max}(H)$ – максимально возможное значение тепла [5].

После запуска модели на компьютере на экране появляются три окна:

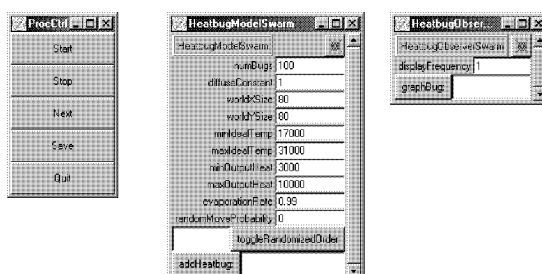


Рис.3

1. Контрольная панель, с кнопками:

Start – запускает выполнение модели (то есть дает поток времени);

Stop – останавливает модель (останавливает течение времени);

Next – передает шаг (после Stop нажмите дважды);

Save – сохраняет позицию окон;

Quit – выход из программы.

2. Окно HeatbugModelSwarm, которое позволяет управлять начальными параметрами модели (перед первым запуском).

numBugs	–	число жуков;
diffuseConstant	–	контрольная степень распространения тепла;
	0	нет распространения,
	1	максимальное «реальное» распространение,
	> 1	создание тепла из ничего!
evaporationRate	–	контрольная степень испарения тепла;
worldXSize	–	ширина решетки;
worldYSize	–	высота решетки;
minIdealTemp	–	min/max диапазон температур, используется
maxIdealTemp		для произвольного выбора, когда жуки
		создаются;
minOutputHeat	–	min/max диапазон выходного тепла,
maxOutputHeat		используется для произвольного выбора,
		когда жуки создаются;

randomMoveProbability — вероятность жука переместиться наугад (каждый шаг).

3. Окно HeatBugObserverSwarm.

После нажатия start появятся два больших окна.

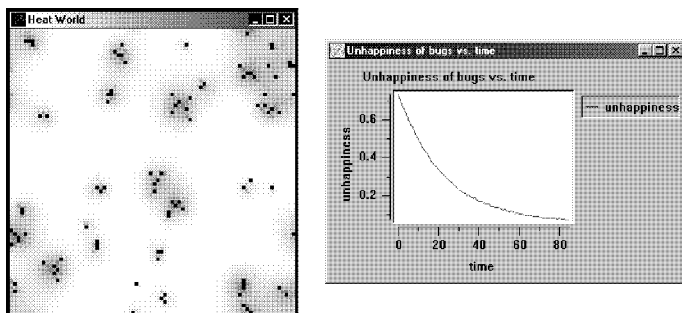


Рис.4

Первое — 2-мерное растровое изображение жуков (темные квадраты), перемещающихся в 2-мерном торе, вместе с (через некоторое время) уровнем температуры в различных местах (более серые означают более горячие). Второе — график среднего недовольства относительно времени.

Наблюдая за ходом развития процесса, можно увидеть, что большинство жуков группируются, и затем тепло начинает накапливаться в этих областях. Сначала среднее недовольство будет падать, а затем выравниваться, впоследствии повышаться.

Можно остановить модель и запустить снова, а затем выполнить ее с другими значениями параметров (например, количество жуков, диапазон температуры или выходного тепла, diffusionConstant, и т.п.)

Таким образом, Swarm обеспечивает структуру (каркас), чтобы управлять агентами и средой, а также инструментальными средствами анализа для диалогового экспериментирования.

ЛИТЕРАТУРА

1. Бартоломью Д. *Стохастические модели социальных процессов*. М.: Финансы и статистика, 1985.
2. *Опыт моделирования социальных процессов*. Киев, 1989.
3. Gilbert, G. Nigel. *Simulation for the social scientist* / Gilbert and Klaus G. Troitzsch 1999.
4. Бахвалов Л.А. *Компьютерное моделирование: долгий путь к сияющим вершинам* // Компьютерра. 1997. N 40. С.26–36.
5. *University of Michigan (Computer Lab)*
<http://www.pscs.umich.edu/LAB/Doc/SwarmStuff/intro-heatbugs.html>
6. Телеконференция *simsoc* в Интернет. <http://www.mailbase.ac.uk/lists/simsoc/>
7. *Swarm Development Group (SDG)* <http://www.swarm.org/community.html>