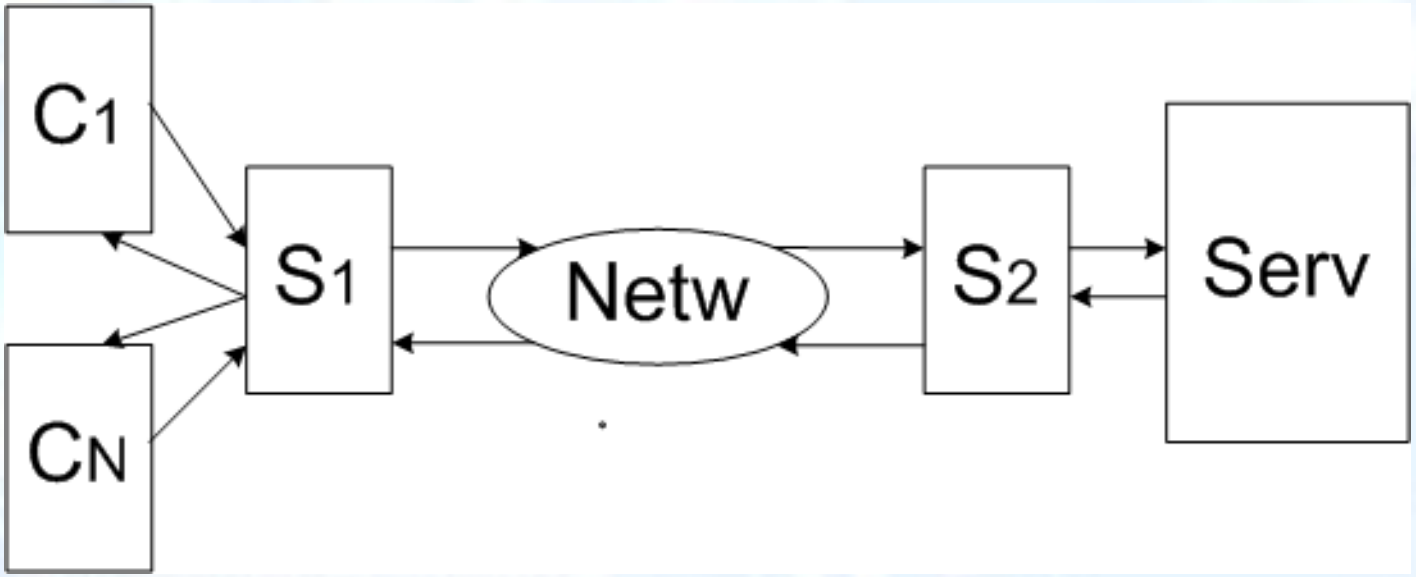
Имитационное моделирование в исследовании и разработке информационных систем

Лекция 4

Основные подходы к организации дискретно-событийных имитационных моделей.

Примеры библиотек, языков, систем моделирования

Пример моделируемой системы



Пример моделируемой системы (2)

- Состав:
  - веб-клиенты на хосте h1
  - веб-сервер на хосте h2
  - Фрагмент сети Internet между h1 и h2
  - Стек TCP/IP на h1 и h2
- Цели:
  - исследовать зависимость очереди запросов к серверу от числа клиентов;
  - воспроизвести временную диаграмму работы системы на уровне состояния сервера и клиентов, очередей запросов

Параметры

Число клиентов	N
Длина i-го запроса j-го клиента	reql(i,j)
Интервал между запросами	reqtime (i,j)
Время прохождения стека на h1	stacktime1(i)
Время прохождения стека на h2	stacktime2(i)
Время прохождения сообщения в сети	Netwtime(l)
Время подготовки ответа	Anstime(I,j)
Длина ответа	Ansl(I,j)

Предположения

- Клиент генерирует следующий запрос через интервал времени после прихода ответа на предыдущий
- Первый запрос второго и последующих клиентов?
- Стек протоколов в каждый момент времени пропускает не более одного сообщения, прочие ставятся в очередь
- Сеть начинает передачу сообщения, получив его полностью. Аналогично, лишь полностью прошедшее через сеть сообщение передаётся стеку протоколов

Предположения (2)

- В сети возможна одновременная передача в обоих направлениях
- В каждом направлении может одновременно передаваться неограниченное число сообщений (вариант: ограниченное число – не более M)

Моделирование посредством планирования событий

```
mod_time=0;
calendar.add( first_event, 0 ); //начальное событие(я)
while(!finish()) // пока не достигнуто условие окончания
{
  event=calendar.get_first_event(); // событие с мин. Временем
  mod_time=event.time;
   switch(event.type)
  {
     case type1: /* обработка */ calendar.add(события,
    mod_time+интервал );
     case type2: ….
  }
}
```

Подходы к организации моделей

- Событийно-ориентированный
- Процессо-ориентированный
  - Транзакты
  - «собственно» процессы и нити (threads)
- Агентно-ориентированный (развитие «собственно» процессного)

События для модели «клиенты-сеть-сервер»

Тип	Атрибуты (доп)	Назначение
CLIENT_REQ	номер клиента длина запроса (номер запроса)	Клиент выдал запрос
CLIENT_ANS	номер клиента длина ответа (номер ответа)	Клиентом получен ответ
STACK_OUT	номер клиента длина (номер) запроса номер хоста	Запрос вышел в сеть
STACK_IN		Запрос или ответ пришёл в стек из сети
SERV_REQ	номер клиента длина запроса (номер запроса)	запрос пришел на сервер
SERV_ANS	номер клиента длина ответа	

Событие – пример описания на Си

```
struct Event{
  int type;
  float time;
  union {
    int client_num, req_num;
    int client_num, req_num, host_num;
     …
   } params;
};
```

Обработка событий в общем цикле

```
// Клиент(ы)
case START: // запускаем всех клиентов
for( i=0; I<N; i++)
  schedule( CLIENT_REQ, 0+time[I,0], I, 0);
   break;
case CLIENT_ANS:
   schedule(CLIENT_REQ,
    curr_time+time[client_num, req_num], client_num,
    req_num+1);
// Стек протоколов
  case STACK_IN:
 schedule(STACK_OUT, curr_time+time[client_num,
    req_num], client_num, req_num+1);
```

Обработка событий в общем цикле (2)

// Сеть

// не забыть про возможность одновременного присутствия нескольких сообщений в сети

Состояние моделируемой системы

- Текущий номер запроса для клиентов;
- Клиент: создание запроса; ожидание ответа
- Стек протоколов: ожидание; передача;
- Сеть: ожидание, передача
- Сервер: ожидание запроса, выполнение запроса

Недостатки «прямолинейного» событийного подхода

- Нет структуры
- Неудобства детализации модели
- Неудобства объединения моделей и построения иерархии

Для крупных моделей используется внутри библиотечных объектов или внутри среды прогона моделей

Интересный пример: DiskSim – настраиваемая модель дисковой системы

http://www.pdl.cmu.edu/DiskSim/

Усовершенствование схемы планирования событий

Переход к ОО языку

- Компонентам системы ставим в соответствие объекты (процессы)
- Переменные состояния – располагаем внутри процессов
- События помещаем в календарь с указателем на процесс
- У процесса – собственный обработчик событий

Пример библиотеки:
http://www.inf.usi.ch/carzaniga/ssim/

Процессо (транзактно)-ориентированный подход

- Сеть узлов обслуживания (ресурсов)
- Запросы на обслуживание (транзакты)
- Очереди транзактов

Классический пример: GPSS

since 1961

- GPSS World ((C) 2007 Minuteman Software)
- Расширенный редактор GPSS World ((C) Элина Компьютерс)

Фрагмент модели на GPSS

```
        GENERATE  3.34,1.7          ;Create next customer.
        TEST LE   Q$Barber,3,Finis ;Wait if line 3 or less
*                                   else leave shop
        ASSIGN    Custnum,X$Custnum ;Assign number to
   customer
        QUEUE     Barber            ;Begin queue time.
        SEIZE     Barber            ;Own or wait for barber.
        DEPART    Barber            ;End queue time.
        ADVANCE   6.66,1.7          ;Haircut takes a few
   minutes.
        RELEASE   Barber            ;Haircut done. Give up the
   barber.
Finis   TERMINATE 1                 ;Customer leaves.
```

Процессо-ориентированный подход

- Процесс как компонент модели - контекст и поток управления;
- Функция продвижения модельного времени;
- Взаимодействие и синхронизация процессов
- Процессы модели отображаются на нити или процессы ОС;
- Привязка процессов к календарю обеспечивается средой прогона

Процессо-ориентированный подход (2): пример взаимодействия через сообщения

```
process Client <input repl; output req>(int num)
{ msg m; m = message Req;
  m.clientnum = num;
   while( 1 ){
      delay( t_req() )
      m.length = req_length();
        send(m, req);
        receive( m, repl );
   }
}
```

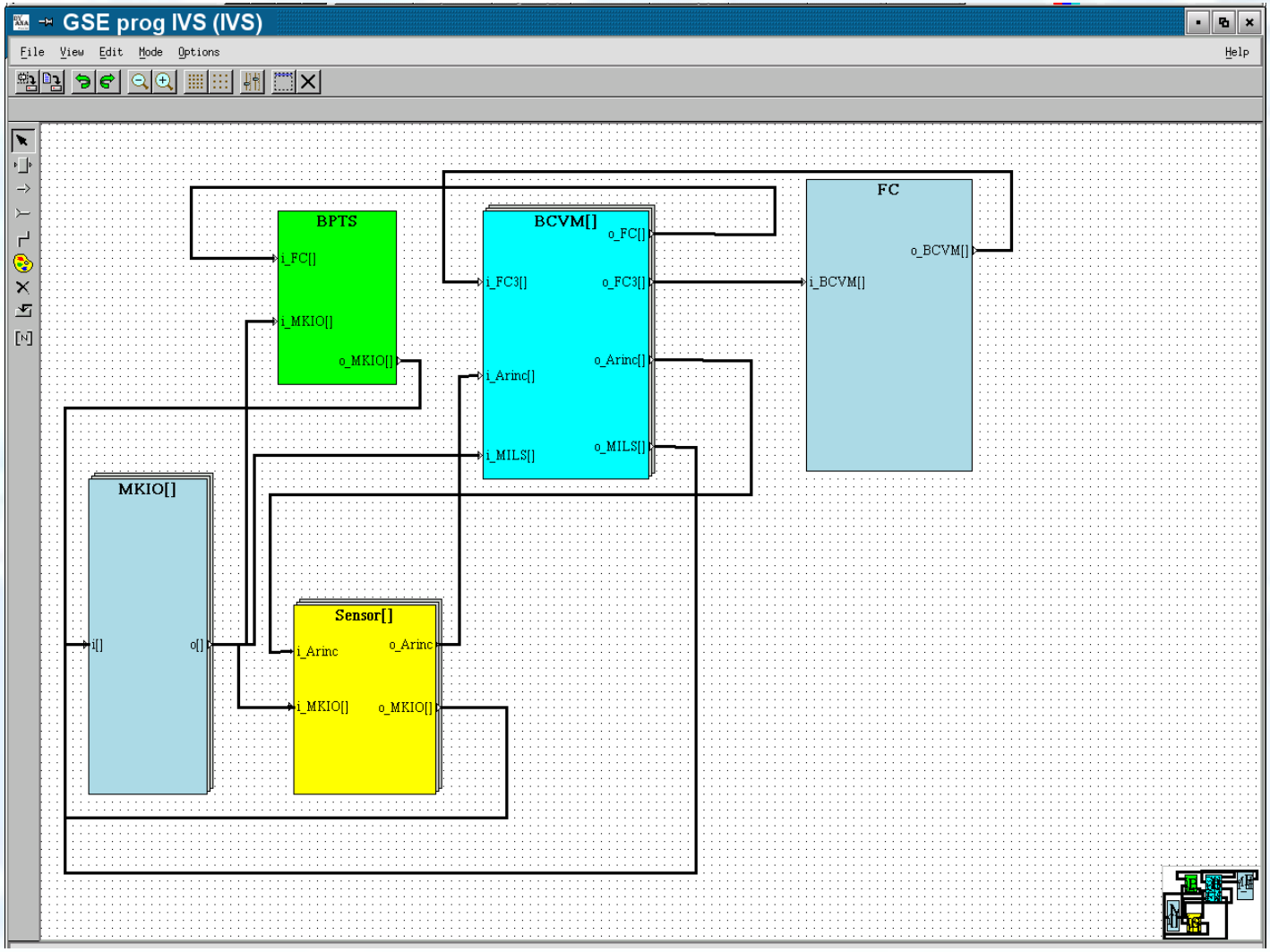
Агентно-ориентированное моделирование

С точки зрения практического применения агентное моделирование можно определить как метод имитационного моделирования, исследующий поведение децентрализованных агентов и то, как это поведение определяет поведение всей системы в целом. При разработке агентной модели, инженер вводит параметры агентов (это могут быть люди, компании, активы, проекты, транспортные средства, города, животные и т.д.), определяет их поведение, помещает их в некую окружающую среду, устанавливает возможные связи, после чего запускает моделирование. Индивидуальное поведение каждого агента образует глобальное поведение моделируемой системы.

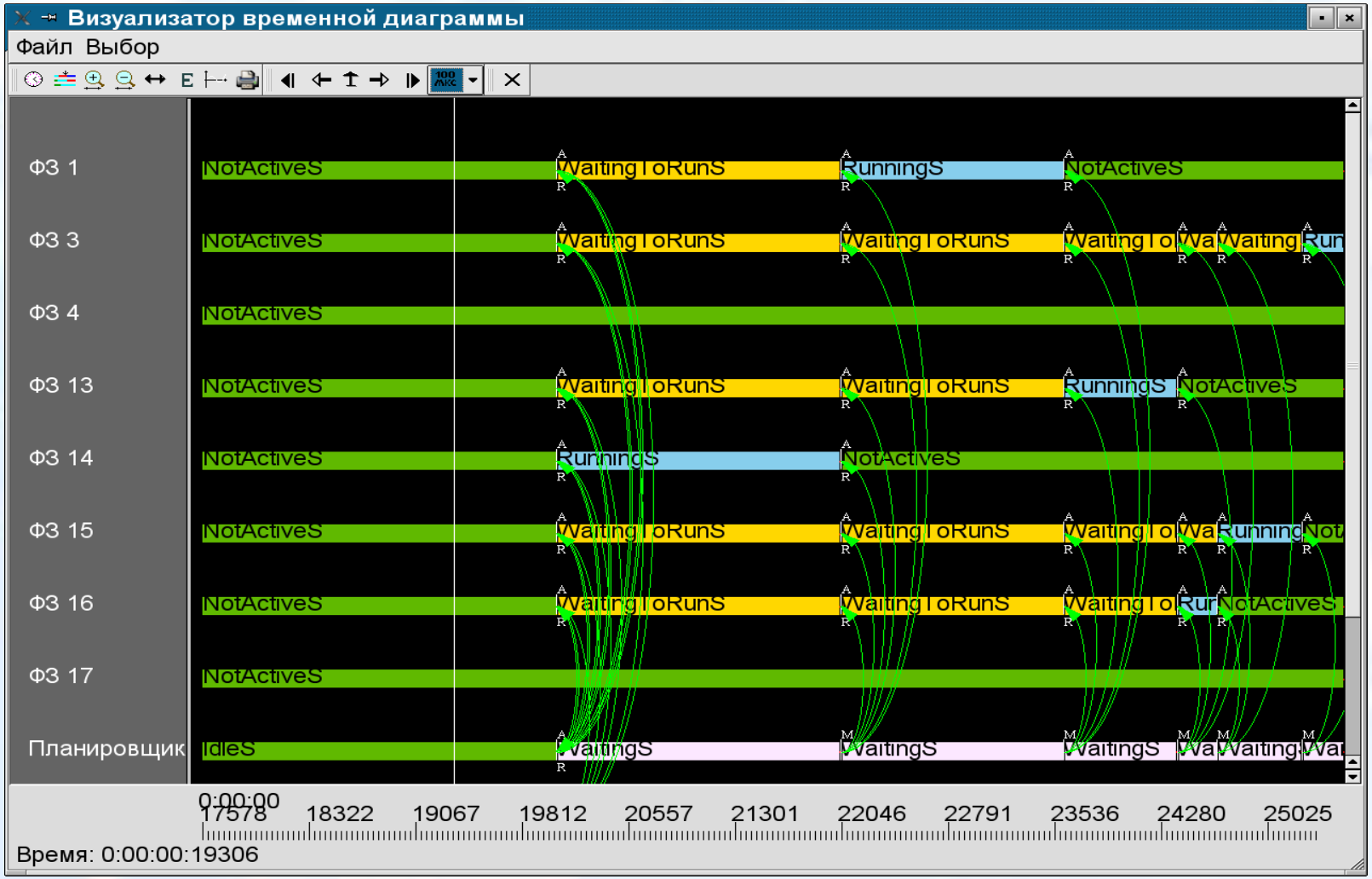
- Создание популяции агентов
  - Динамическое создание и удаление агентов
  - Среда
  - Непрерывное пространство
  - Дискретное пространство
  - ГИС пространство
  - Связи агентов и сети
  - Взаимодействие агентов
  - Синхронизация агентов
  - Сбор статистики по агентам

Типичная архитектура среды ИМ

Структурный редактор



Визуализатор трассы



Спасибо за внимание!