26 Октября 2012

Инсерционное моделирование 1

Лекция 7

Инсерционные машины

Инсерционная машина

Среда моделирования поведения системы, представленной в виде композиции среды и погруженных в нее агентов (инсерционная модель)

Входной язык инсерционной машины описывает поведение системы и ее компонент.

Интерпретатор обходит пространство состояний моделируемой системы, осуществляет поиск целевых состояний или принимает решения по взаимодействию с внешней средой.

Инсерционная машина реального времени взаимодействует с внешней средой

Аналитическая инсерционная машина исследует пространство состояний

Система инсерционного моделирования IMS

Предназначена для разработки инсерционных моделей и машин. Разработана в ИК им. В.М.Глушкова. Использована в промышленных Разработках для фирм Моторола и Uniquesoft. На ее основе создана система VRS верификации программного обеспечения распределенных систем.

Входной язык инсерционной машины

Action Language

Синтаксис

<AL>::= <terminal> | <action> | <action>.<AL> |
<AL> + <AL> | <functional expression> |
<env state>[<list of AL>]
<terminal>::=Delta | bot | 0

Ядро AL – без функциональных выражений и функции погружения

Описание системы (модели)

Состоит из трех частей:

Функция погружения

Система функциональных определений

Начальное состояние системы (выражение AL)

Функция погружения

определяет основную часть инсерционной машины

Система функциональных определений

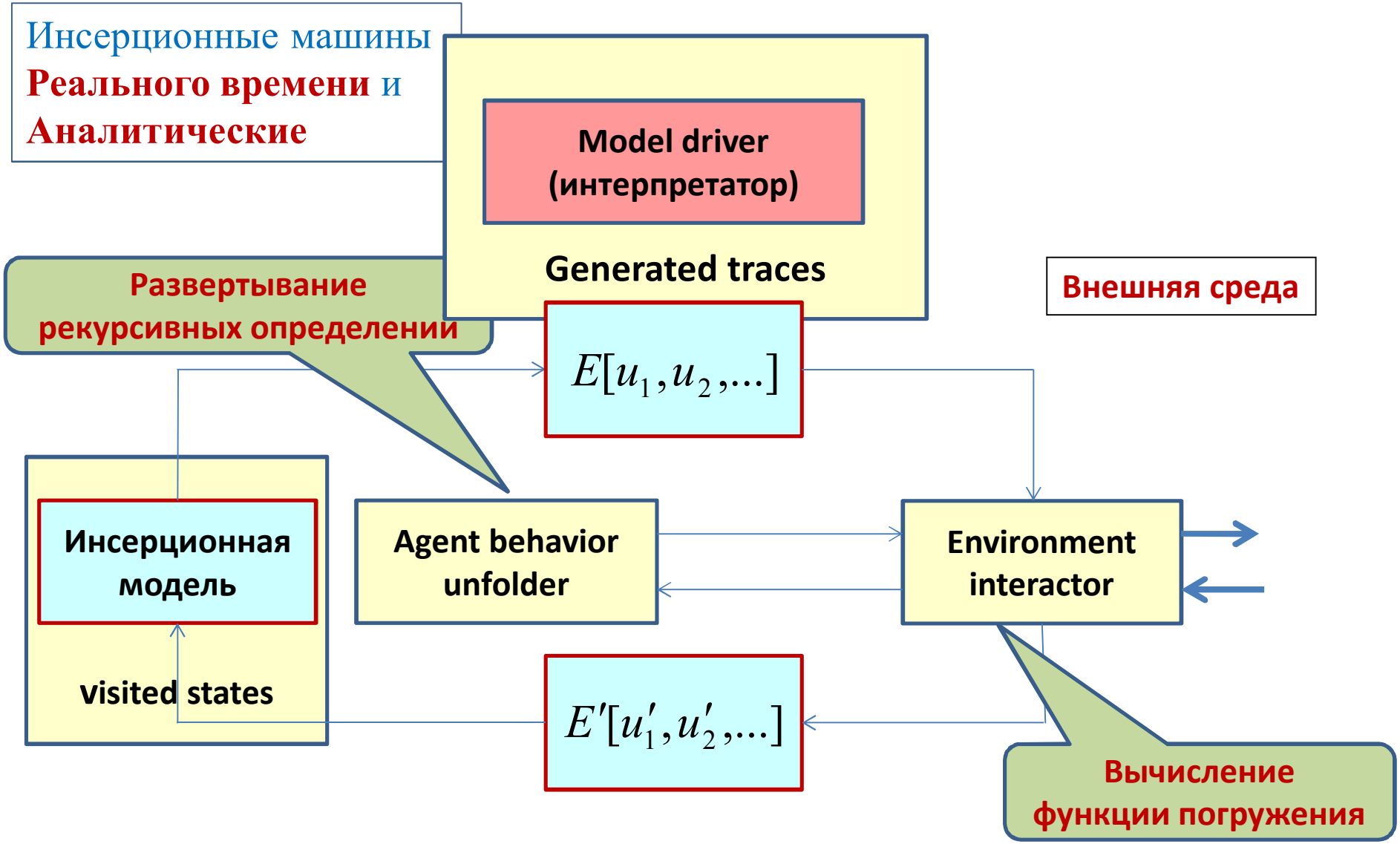
определяет набор стандартных поведенческих функций инсерционной машины и описание модели (системы уравнений для определения поведения агентов)

Базовый язык программирования

поддерживает разработку описания системы (модели)

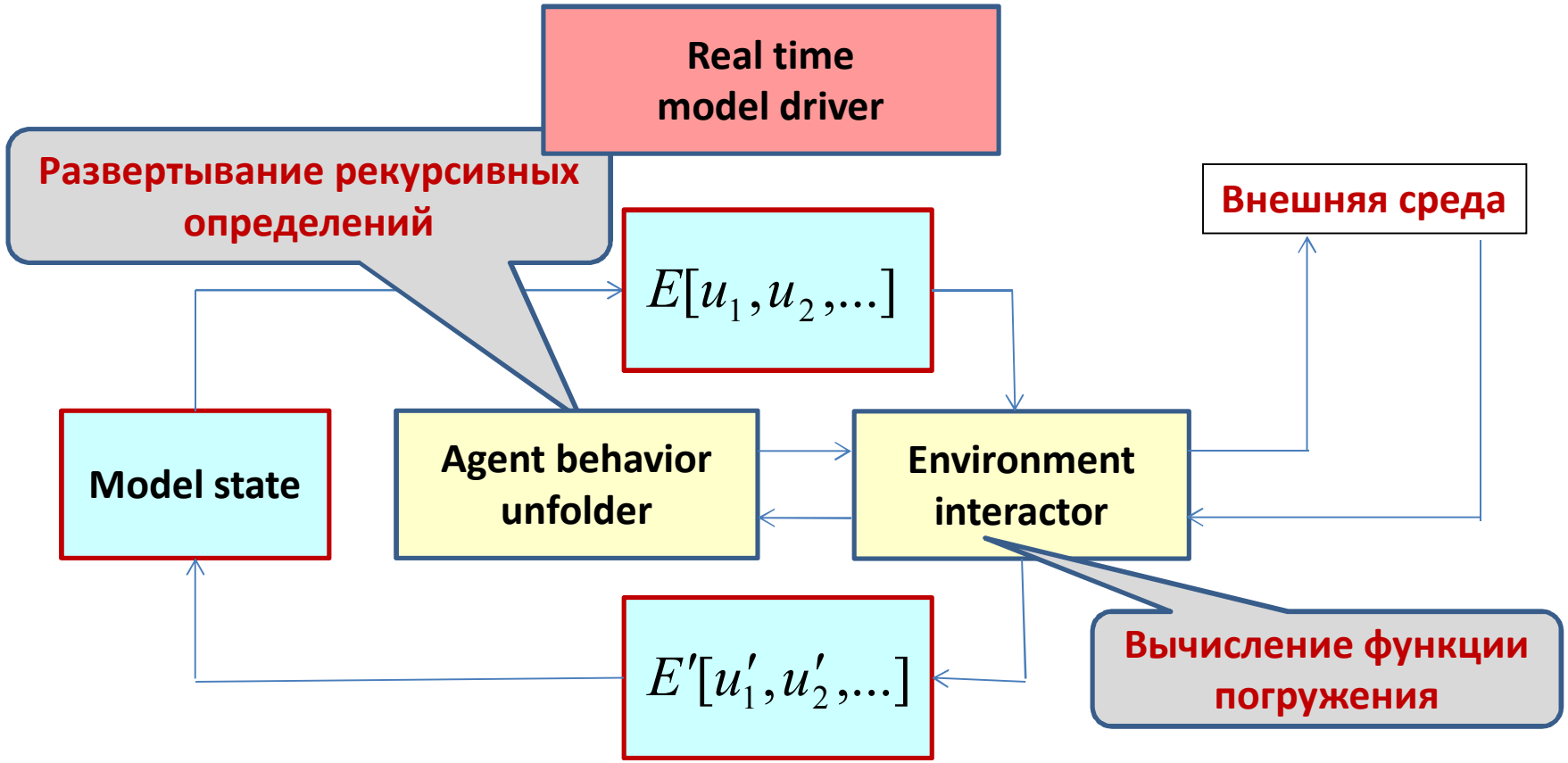
В системе IMS таким языком является APLAN, входной язык системы алгебраического программирования APS.

Архитектура инсерционной машины



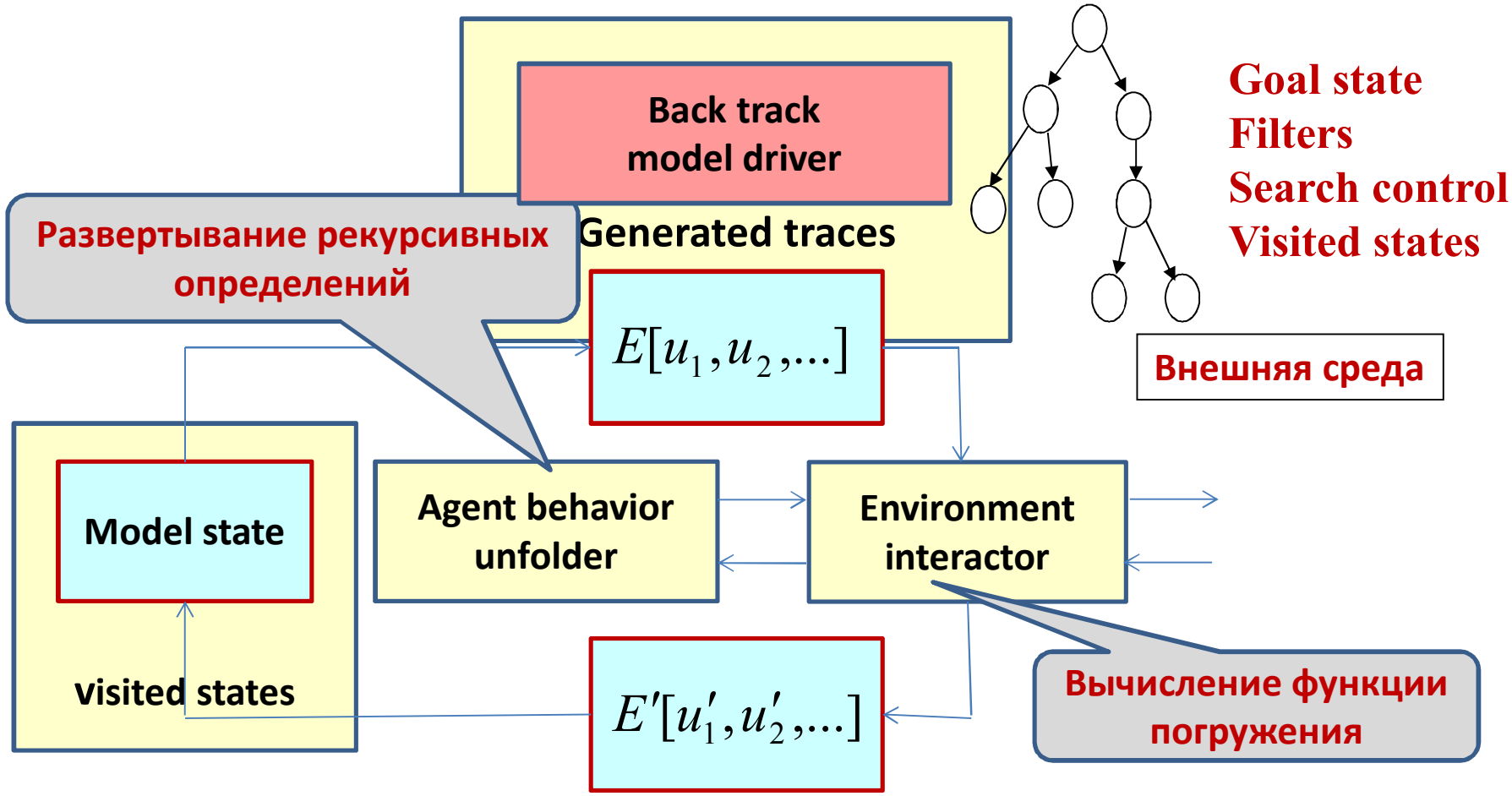
Инсерционная машина реального времени

взаимодействие с внешней средой в условиях временных ограничений



Аналитическая инсерционная машина

(верификация, прогнозирование, принятие решений)



Атрибутные среды

Логическая база (словарь, сигнатура)

Система типов с областями значений (int, real, …)

Набор типизированных функциональных символов и предикатов

Предикат – функциональный символ типа $(\xi_1, \xi_2, ..., \xi_m) \to \text{Bool}, m \geq 0$

Интерпретированные функциональные символы (+, *, …)

Интерпретированный символ арности 0 – **константа**

Атрибуты – неинтерпретированные функциональные символы

Атрибут арности 0 – **простой атрибут**, арности > 0 – **функциональный**.

Базовый логический язык – типизированный (многосортный) язык первого порядка

Может быть расширен модальностями темпоральной или нечеткой логики

Атрибутные выражения: $f(t_1, t_2, ..., t_m), m \geq 0$

Формулы: атрибутные выражения → термы → литералы →

пропозициональные связки, кванторы, ограниченные кванторы.

Конкретные атрибутные среды

Неразложимое состояние конкретной среды – частичное отображение $s \subseteq A \to D$ множества A константных атрибутных выражений в множество их значений D сохраняющее типы.

Действия – такие же, как действия императивной среды: условия и присваивания. Новым является использование структур данных в левых частях присваиваний.

Программы -- агенты

$$\text{if } \alpha \text{ then } P \text{ else } Q = \text{check } \alpha.P + \text{check } \neg\alpha.Q$$

$$\text{while } \alpha \text{ do } P = \text{if } \alpha \text{ then}(P; \text{while } \alpha \text{ do } P)\text{else } \Delta$$

Параллельная композиция – программы над общей памятью

Символьные атрибутные среды

Неразложимое состояние символьной среды – формула базового языка.

Действия – преобразователи формул (predicate transformers)

Состояние вида $E[u_1, u_2, \ldots]$ отождествляется с формулой

$$E \wedge state(m_1) = u_1 \wedge state(m_2) = u_2 \wedge \ldots$$

где $m_1, m_2, \ldots$ имена агентов.

Конкретное состояние *s* можно отождествить с формулой вида

$$t_1 = a_1 \wedge \ldots \wedge t_n = a_n$$

Состояние символьной среды *F* **покрывает множество** конкретных состояний

$$\{s : A \to D \mid s \models F\}$$

Локальные определения

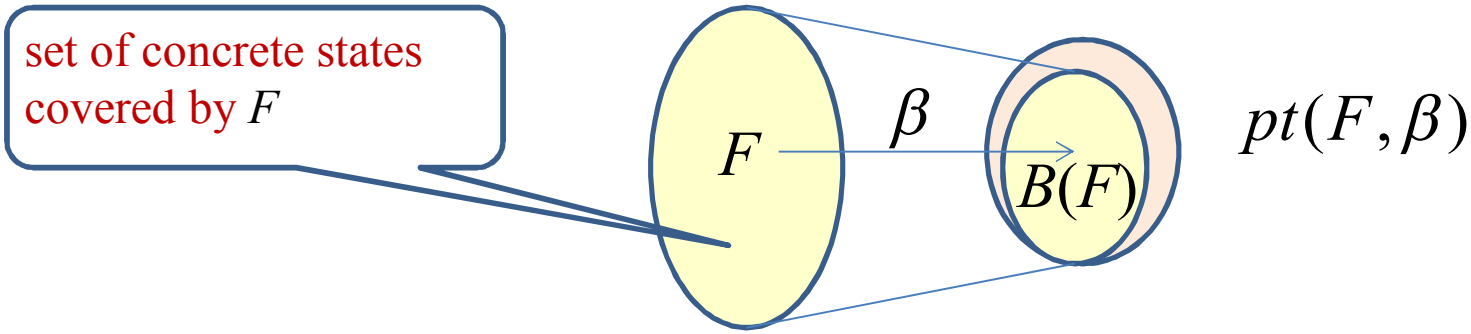
$$B = \forall(x,y)(\alpha(x,y) \rightarrow <u(y)> \beta(x,y))$$

x, y are lists of parameters
α and β are formulae
β can include assignments

Local description *B* is applicable to *E* if

$$E \wedge \exists(x,y)\alpha(x,y) \neq 0$$

Application of local description *B* to a state *E*

$$\exists x \alpha(x,z) \wedge E \neq 0 \Rightarrow$$

$$E \xrightarrow{u(z)} \exists x(pt(\alpha(x,z) \wedge E), \beta(x,z)) = B(E)$$