

*8 Апреля 2013*

# **Инсерционное моделирование 2**

## **Лекция 6**

### **ICAR и IMS**

# Задачи

## Задачи к лекции 2

1. Показать, что, если  $u = aub$ , то  $u = au$
2. Решить уравнение  $u = aubuc + \Delta$
3. Построить алгоритм решения нелинейных уравнений (обобщение алгоритма синтеза стековых автоматов), если это возможно.

Решение: система линейных уравнений.

## Упражнение к лекции 3

Написать остальные два уравнения для сети из автоматов

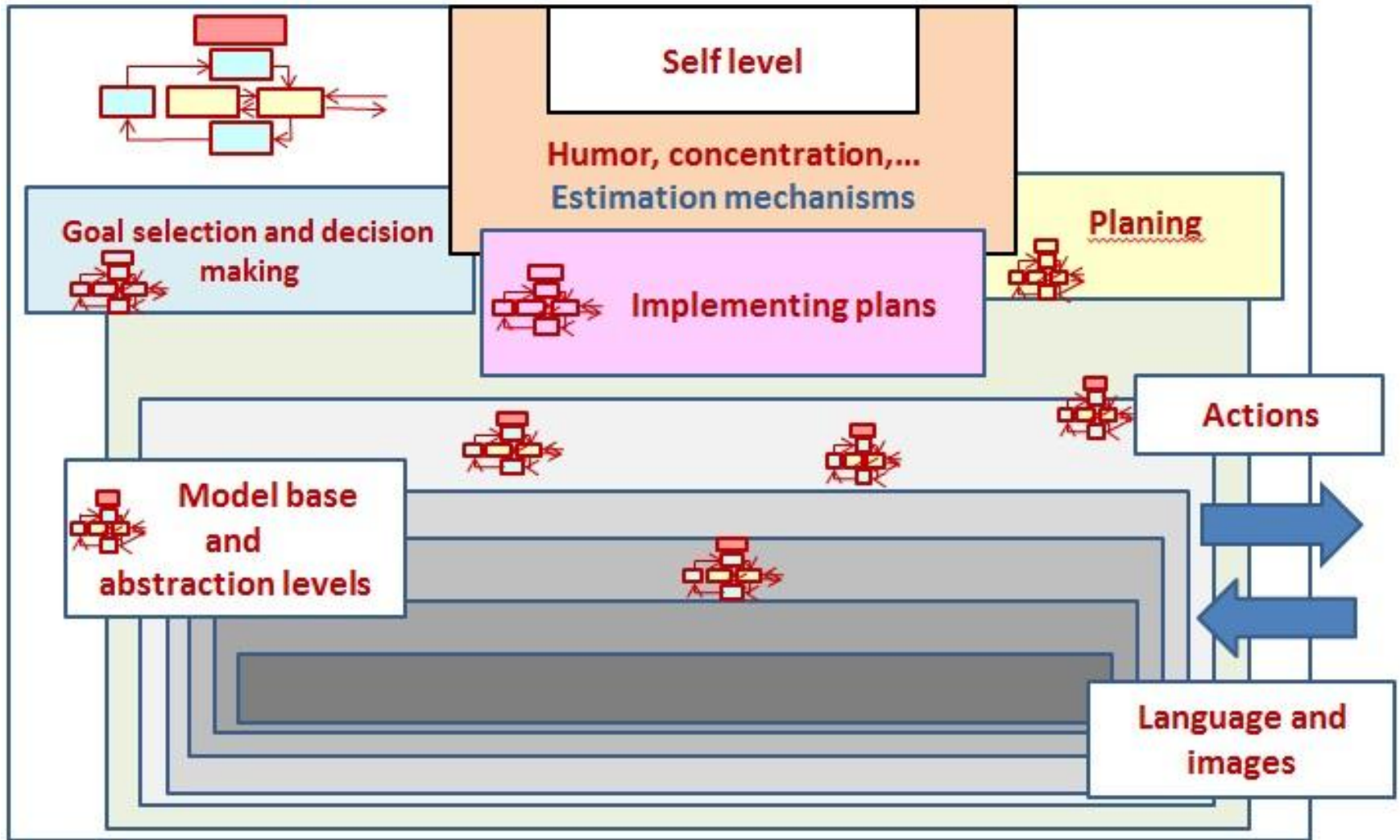
## Задачи

1. Построить поведение непрерывного нейрона
2. Построить среду для дискретных нейронов
3. Построить среду для непрерывных нейронов (аттракторы)

## Задача к лекции 4:

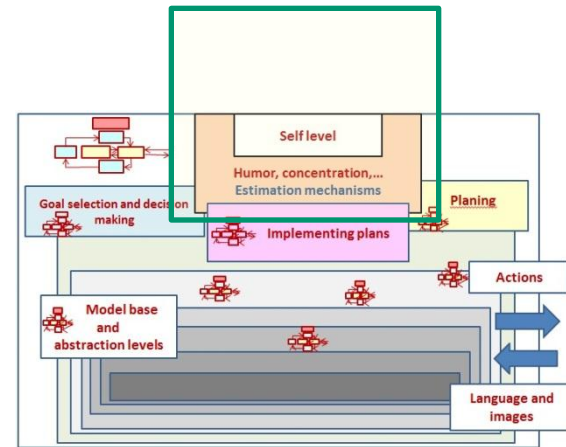
перенести алгоритмы обучения непрерывных сетей на дискретные.

# Когнитивная архитектура ICAR



# Уровень Эго

Чувства, эмоции  
Оценка успеха  
Управление системой



Инсерционная когнитивная архитектура – это инсерционная машина, которая **осознает себя**, имеет центр **оценки успешности** своего поведения и стремится к **многократному достижению максимального успеха**

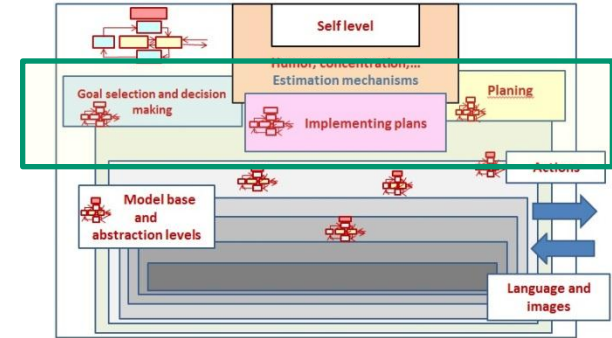
**осознает себя:** контролирует все внутренние системы, различает внутреннюю и внешнюю среду.

# Уровень деятельности

Выбор цели  
Планирование  
Реализация плана

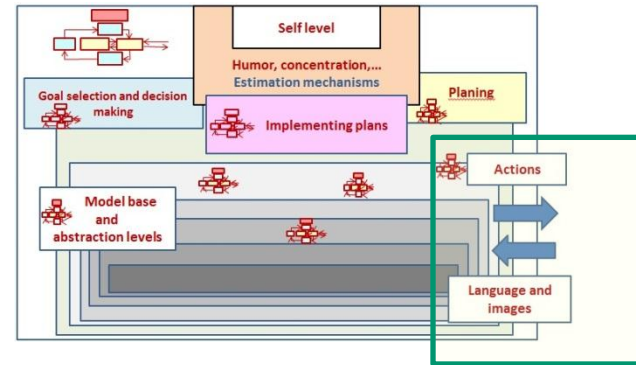
подцели,  
коррекция плана,  
использование эвристик  
и накопленного опыта

Работа в режиме реального времени  
Анализ прошлого



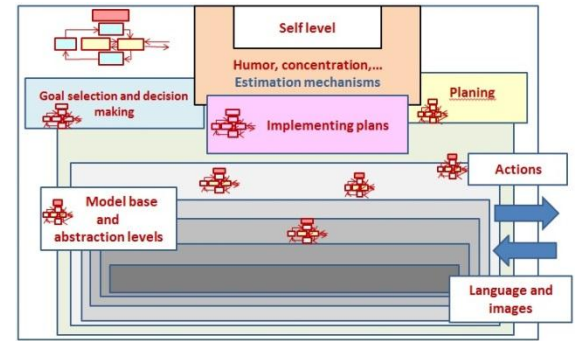
# Взаимодействие

Прием информации  
символьной  
образной  
Выполнение действий



# Проблема языка

Язык, которым мы пользуемся для общения есть **метаязык** для языка наших внутренних моделей.



**Мысль изречённая есть ложь.**

**SILENTIUM! \***

Молчи, скрывайся и таи  
И чувства и мечты свои -  
Пускай в душевной глубине  
Встают и заходят оне  
Безмолвно, как звезды в ночи,-  
Любуйся ими - и молчи.

Как сердцу высказать себя?  
Другому как понять тебя?  
Поймёт ли он, чем ты живёшь?  
**Мысль изречённая есть ложь.**  
Взрывая, возмутишь ключи,-  
Питайся ими - и молчи.

Лишь жить в себе самом умей -  
Есть целый мир в душе твоей  
Таинственно-волшебных дум;  
Их оглушит наружный шум,  
Дневные разгонят лучи,-  
Внимай их пенью - и молчи!..



# Память

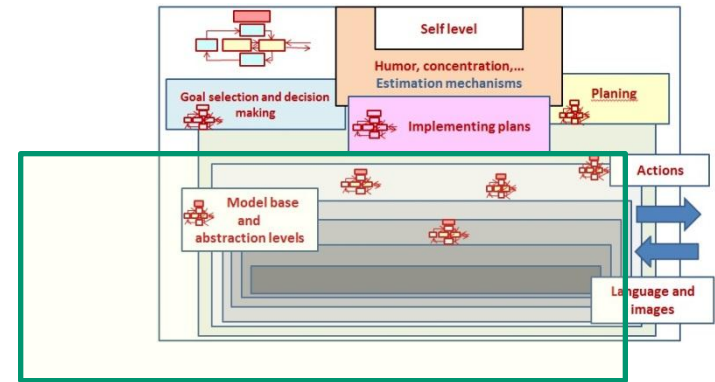
Рабочая память

Эпизодическая память

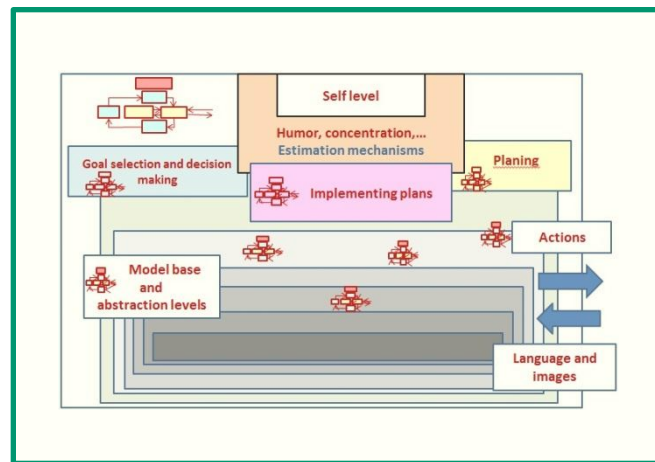
Семантическая память (онтологии)

Уровни абстракции

База инсерционных машин



# Параллелизм

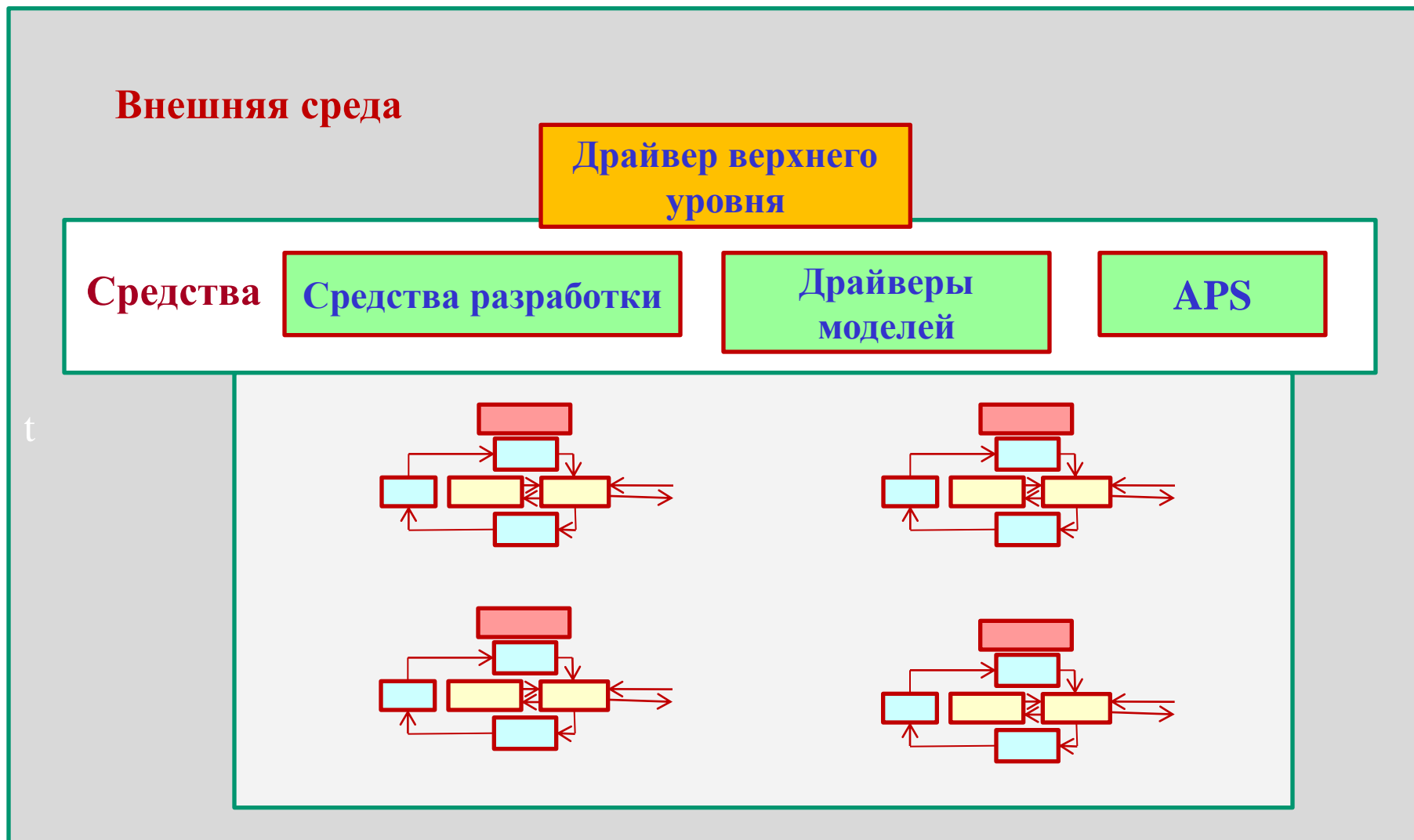


Инсерционные машины работают  
на всех уровнях параллельно.

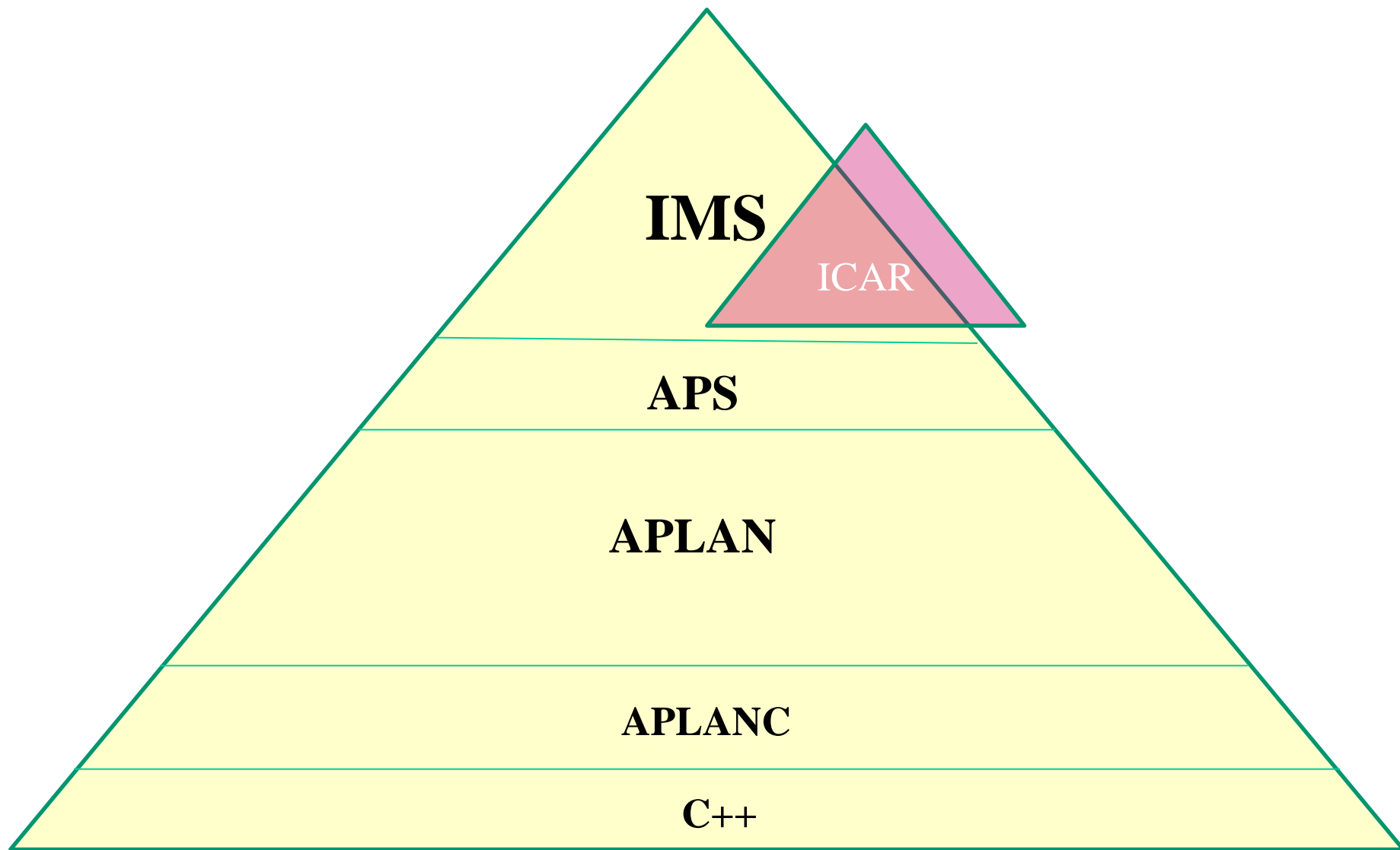
Переходы между уровнями

от глубокого подсознания к уровню Эго  
и обратно

# Система инсерционного моделирования IMS: среда разработки инсерционных машин



# Архитектура IMS



# Система APS

[www.apssystem.org.ua](http://www.apssystem.org.ua)

## История

**1987** системы переписывания термов

**1989** первая версия системы, язык APLAN.

**Применения** в компьютерной алгебре, искусственном интеллекте, моделировании параллельных процессов, верификация (Моторола)

**2004** вторая версия системы (использование смарт поинтеров)

**2009** третья версия, язык APLANC.

**усовершенствованная** машина переписывания

**2013** четвертая версия

**Интеграция** с внешними тулами (CVC, Z3, ...)

# Система APS

[www.apssystem.org.ua](http://www.apssystem.org.ua)

## Основные особенности:

Алгебраические структуры данных  
(графовые термы)

Системы переписывания термов + стратегии переписывания

Интеграция основных парадигм программирования:

императивное,  
функциональное,  
алгебраическое  
логическое.

Высшие парадигмы

параллельное,  
объектно-ориентированное  
последовательное,  
параллельное,  
агентное,  
инсерционное.

# Императивное программирование

- Структурные программы
- Программы с goto
- Программы с рекурсивными вызовами
  - Функциональное программирование
- Программы с вводом-выводом
  - Взаимодействие со средой
  - Параллельное крупноблочное программирование (обмен сообщениями)
- Интеграция основных парадигм в императивное

Лекции 4-5 2004 весна

# Инсерционная модель императивной программы

## Среда

**Состояния среды** – состояния памяти

**Действия среды** – операторы (в частности, присваивания) и условия

**Переходы среды:**

$$\begin{array}{l} s \xrightarrow{y} y(s) \\ u(s) \Rightarrow s \xrightarrow{u} s \end{array}$$

$u$  – условие     $y$  – присваивание

## Агенты

**Состояния** – программы

**Действия агента:** условия и присваивания



# Функция погружения

$$\frac{P \xrightarrow{a} P', s \xrightarrow{a} s'}{s[P] \xrightarrow{\delta} s'[P']}$$

$\delta$  зависит от посылок  
и может вовсе отсутствовать

$$s[P, Q] = s[P; Q]$$

Последовательное погружение

$$s[P, Q] = s[P \parallel Q]$$

Параллельное погружение

$$P(R1 \parallel R2 \parallel \dots)Q$$

fork/join технология

$$s[P + Q] = s[P] + s[Q]$$

недетерминированные программы

# Многоуровневые императивные среды

## Среда

### Состояния среды –

состояние памяти + погруженные  
в нее именованные агенты в состоянии  
ожидания

### Действия среды –

операторы, условия,  
вызовы агентов и возвраты,  
ввод/вывод

# Функция погружения

$$u \xrightarrow{\text{call}(f(x))} u'$$

---

$$s[(f : t[v]), p[u]] \xrightarrow{\tau} s[(f : t[v]), p[\text{call}(t, s, x, v); u']]]$$

$$\text{call}(t, s, x, v) = t'[v']$$

$$u \xrightarrow{\text{return } x} u'$$

---

$$s[t[u]] \xrightarrow{\tau} \text{return}(s, x)[\Delta]$$

# Пример семантики функций *call* и *return* (вызов по значению)

В среде  $t$  выделены атрибуты  $r$ , соответствующие формальным параметрам функции  $f$ . Среда не может двигаться, пока формальные параметры не получат значений.

$$t' = ((r := x)t), v' = v$$

$$call(t, s, x, v) = t'[v']$$

В среде  $s$  выделен атрибут  $z$ , соответствующий результату функции  $f$ .

$$return(x) = ((z := x)s)$$

$$\frac{u \xrightarrow{call(f(x))} u'}{s[f : t[v]], p[u] \xrightarrow{\tau} s[f : t[v]], p[call(t, s, x, v); u']}$$

$$\frac{u \xrightarrow{return\ x} u'}{s[t[u]] \xrightarrow{\tau} return(s, x)[\Delta]}$$

# Задачи

1. Построить семантику вызова по имени
2. Исследовать выполнение операторов и проверку условий в многоуровневых средах