



Агент-базирани архитектури

Аделина Алексиева-Петрова

АВТ, 2012-2013

Логически-базирани (символични) архитектури

- Основават се на традиционни системни техники, базирани на знанието, в които околната среда е символично представена.
- Предимството на този подход е, че човешкото познание е символично, така кодирането е по-лесно и хората могат да разберат логиката.
- Недостатъците са, че е трудно да се преведе реалния свят в точно и адекватно символично описание, както и че символичното представяне може да отнеме много време и резултатите да са на разположение прекалено късно, за да са полезни.

Реактивни архитектури

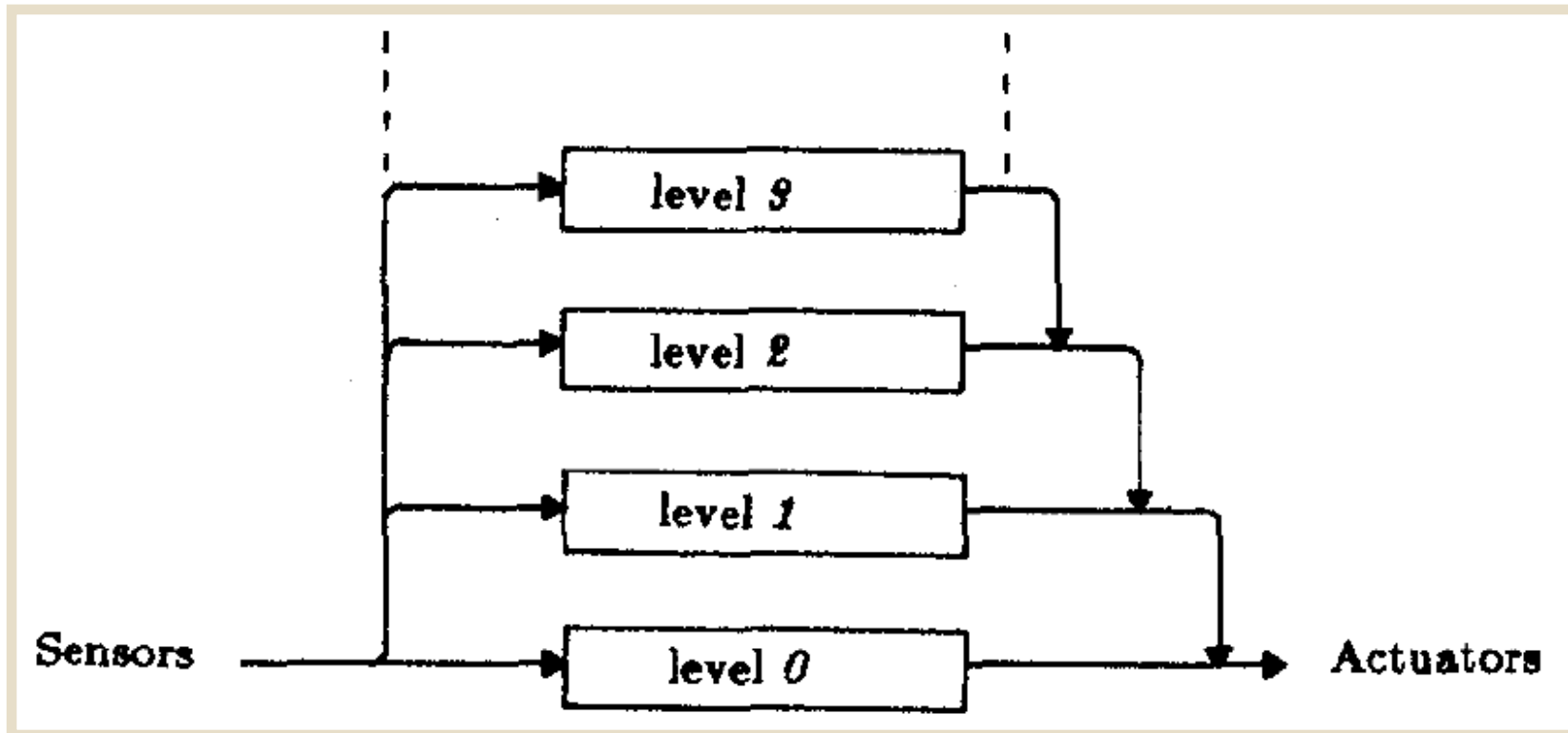
- Имплементират взимане на решение и се основават на механизъм стимул-реакция, задействан от сензор за данни.
- За разлика от логически-базираните архитектури, те не разполагат с централен символичен модел и затова не се използват никакви сложни СИМВОЛИЧНИ МОТИВИ.

Реактивни архитектури

- Най-известната реактивна архитектура е Brooks's subsumption
- В subsumption архитектурата се определят слоеве на крайни автомати, които са свързани с датчици, които предават информация в реално време

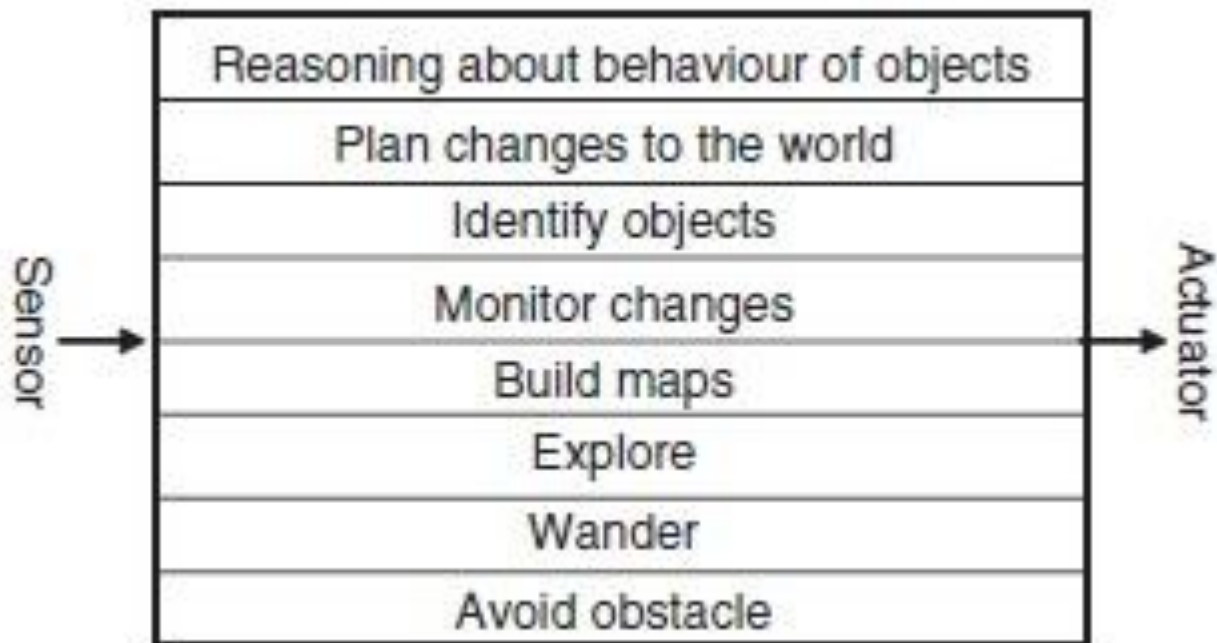
Subsumption Архитектура

- Методология базирана на слоеве за системи за управление на роботи.
- Паралелни и разпределени методи за свързване на сензори и изпълнителни механизми при роботите.



Реактивни архитектури

- Тези слоеве образуват йерархия на поведение, при което контролът е по-слаб в по-ниските нива на стека, като по този начин взимането на решения се постига, чрез целенасочено поведение



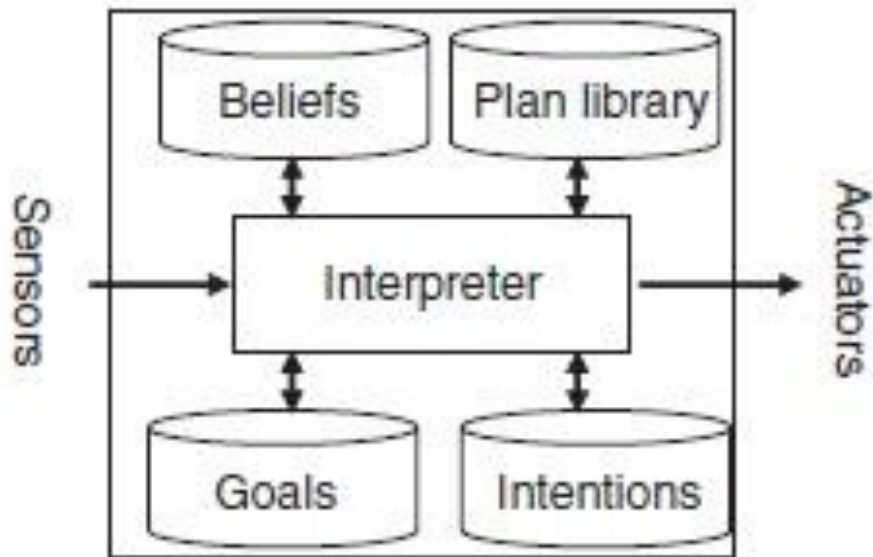
Реактивни архитектури

- Предимството на този подход е, че ще работи по-добре (т.е. ще отговаря по-бързо) в динамична среда, както и че те често са по-прости в дизайна от логически-базираните агенти.
- В действителност, сензор за данни не може да бъде достатъчен да определи подходящите действия и е почти невъзможно да се разработят агенти, които да се обучават от опита си.
- Освен това, предвид факта, че агентът действа въз основа на взаимодействието между различни видове поведение, е много трудно да се създадат реактивни агенти, които да отговорят на специфични задачи, по-специално, когато такива агенти трябва да се реализират, чрез голям брой поведения.

Вяра, Желание, Намерение BDI (Belief, Desire, Intention) архитектури

- Това са най-популярните архитектури и затова много агент-базирани системи са реализирани, именно чрез BDI (например PRS, JAM, JACK и др.) с широк спектър от приложения, демонстрирайки жизнеспособността на модела.
- Те предлагат логична теория, която определя смислово отношение между вяра, желание и намерение.
- Една от най-известните BDI архитектури е Procedural Reasoning System (PRS). Тази архитектура се основава на четири основни структури от данни: вярвания, желаниа, намерения и планове, както и преводач.

Вяра, Желание, Намерение BDI (Belief, Desire, Intention) архитектури



- В системата PRS, вярванията представляват информация, която агентът има за околната среда, но тя може да е непълна или неточна.
- Желанията са задачи, възложени на агента и съответстващи на целите, които трябва да изпълни.
- Намеренията представляват желаниа, които агентът е решен да постигне.
- И накрая, плановете уточняват начините на действие, които могат да бъдат следвани от агента, за да постигне своите намерения.

Вяра, Желание, Намерение BDI (Belief, Desire, Intention) архитектури

- Тези четири структури от данни се управляват от агент-преводач, който е отговорен за актуализиране на вярванията, за генериране на нови желания (задачи) на база на новите вярвания, и за избор на намерения от набор от активните желания в момента.
- На последно място, преводачът трябва да избере план за действие, базиран на текущите намерения на агента и процедурните знания.

JACK®

- JACK приложенията се състоят от колекция от агенти които имат вход от средата и си комуникират един с друг.
- Всеки агент е дефиниран в сферата на неговите цели, знания и възможности и след това се оставят да изпълняват своите функции автономни, вградени в средата.

Iraqi Marketplace

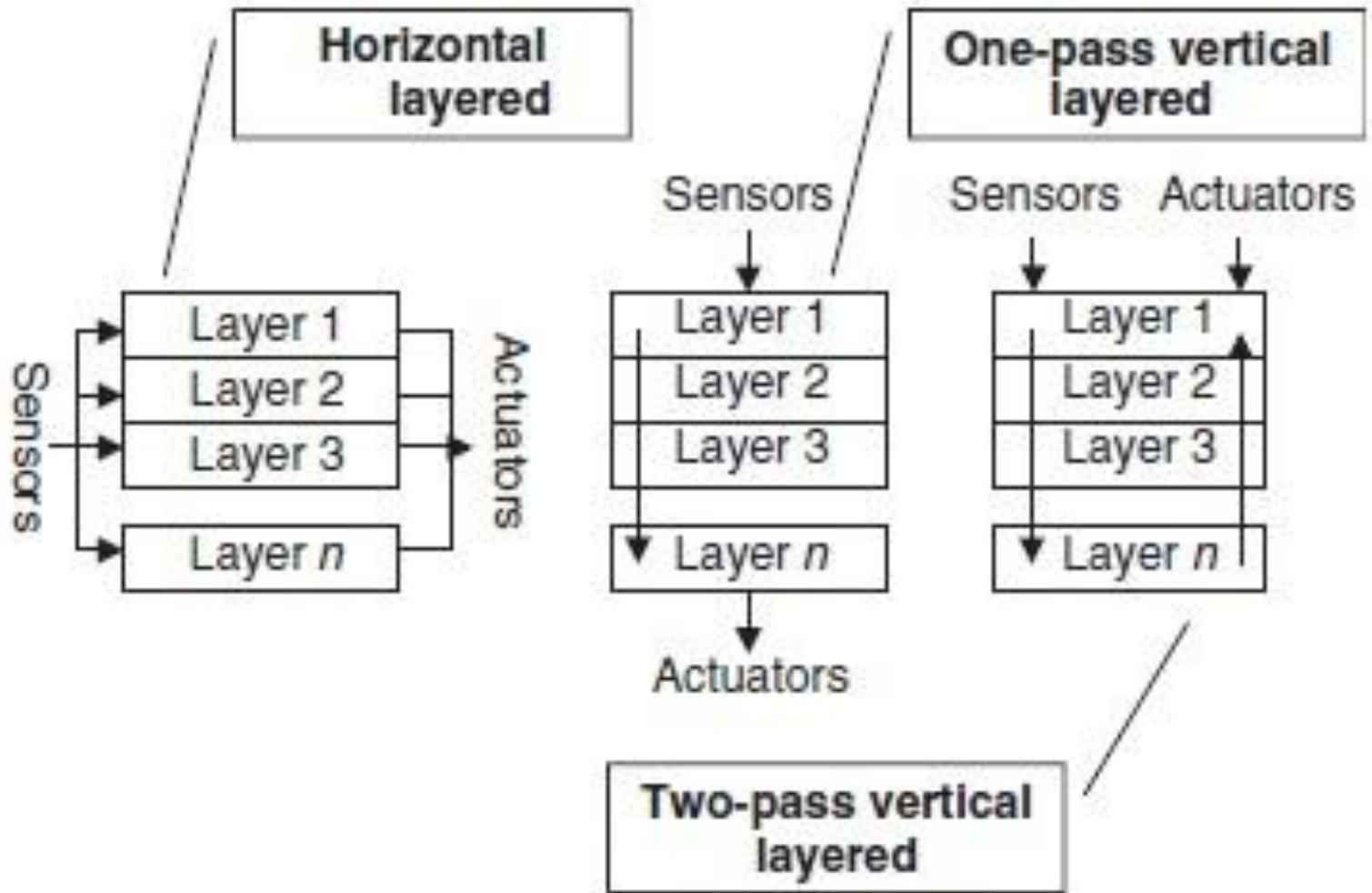
- This scenario is based in an Iraqi marketplace with civilian shoppers (CoJACK agents), a suicide bomber (CoJACK agent) on the prowl and a Marine on patrol (the human player). The other marine (to right of screen) could also be a human player but is stationary during this event. At the bottom of the screen there is a graph showing the fear level of the suicide bomber. When the action starts, the bomber's fear level is zero but it quickly goes up when he spots the Marine.



Пластови (хибридни) архитектури

- За да има гъвкавост, в подсистемите подредени като пластове на йерархията, се настаняват двата вида поведение на агентите.
- Има два вида контролни потоци в пластовата архитектура:
 - *хоризонтален*
 - *вертикален.*

Пластови (хибридни) архитектури



Хоризонталната пластова архитектура

- В *хоризонталната пластова архитектура*, слоевете са пряко свързани със сензорния вход и изхода за действие, където по същество всеки слой действа като агент.
- Основното предимство тук е простотата на дизайна, тъй като ако агент се нуждае от n на брой различни вида поведение, то архитектурата изисква само n слоя.

Хоризонталната пластова архитектура

- Въпреки това, тъй като всеки слой е със сила на агент, техните действия могат да станат несъвместими и това да доведе до необходимост от функция медиатор, който да контролира дейностите.
- Друг проблем е големият брой на възможните взаимодействия между хоризонталните пластовете – m^n (където m е броят действия в един пласт).

Вертикалната пластова архитектура

- Елиминира някои от тези проблеми, като сензорният вход и изходът за действие, които са свързани с най-много един слой.
- Така не може да се получи несъвместимост между действията на агентите.
- Вертикалните пластови архитектури се делят на два вида: архитектури с еднопосочен и с двупосочен контрол.

Вертикални пластови архитектури

- В архитектурите с едно преминаване, контролните потоци тръгват от входния слой, където получават данни от сензора и стигат до крайния слой, който генерира действия на изхода.
- В архитектурите с две преминавания, потоците от данни преминават през последователността от слоеве и след това контролните потоци се връщат обратно.

Предимства и недостатъци

- Основното предимство на вертикалните пластови архитектури е, че взаимодействието между слоевете се намалява значително.
- Основният недостатък е, че архитектурата зависи от всички слоеве, така че ако един слой се провали, цялата система се проваля.