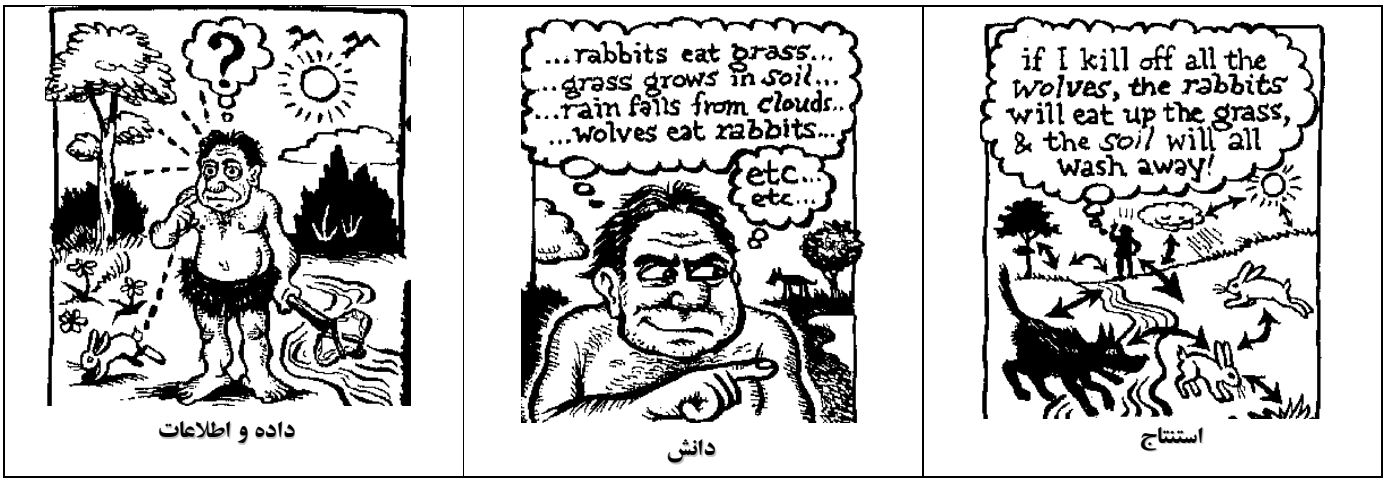
مباحث پیشرفته

دانش: به‌کارگیری داده و اطلاعات است.



شکل15-1: داده و اطلاعات، دانش، استنتاج

انواع دانش(تکمیل مباحث پیشین)

- دانش رویه‌ای(Procedural Knowledge)
  - Knowledge of processes
  - Example: how to clean your face? (men v.s. women)
- دانش توصیفی(Declarative Knowledge)
  - Knowledge of propositions or facts (declarations) that are true or false
  - Example: rabbits eat grass
- دانش ضمنی(Tacit Knowledge)
  - Knowledge which cannot be expressed in language
  - Example: how to drive a car?
- فرا دانش(Meta-knowledge)
  - Knowledge about knowledge
  - Example: According to my study experience, “types of knowledge” will be a question in the mid-term exam.

انواع استنتاج(تکمیل مباحث پیشین)

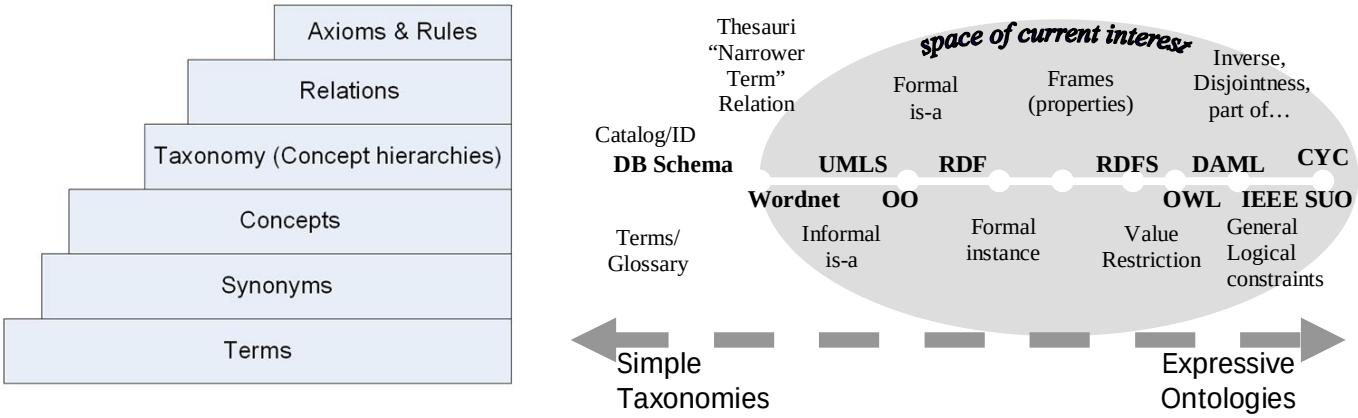
- **Deduction**: reasoning where conclusions must follow from premises
- **Induction:** inference is from the specific case to the general
- **Intuition:** no proven theory
- **Heuristics:** rules of thumb based on experience
- **Generate and test**: trial and error
- **Abduction:** reasoning back from a true condition to the premises that may have caused the condition
- **Default:** absence of specific knowledge
- **Nonmonotonic:** New evidence may invalidate previous knowledge
- **Analogy:** inferring conclusions based on similarities with other situations

آنتولوژی (هستان‌شناسی)

در معنی عام کلمه، آنتولوژی به شاخه‌ای از علم فلسفه اتلاق‌می‌گردد که به دنبال پاسخ به سؤالاتی است از قبیل؛ «هستی چیست؟» و «چه ویژگی‌های مشترکی در بین تمام موجودات وجود دارد؟». در فلسفه، آنتولوژی سیستمی از دسته‌های مختلف است که از دیدگاه خاصی نسبت به دنیا ایجاد شده‌اند.

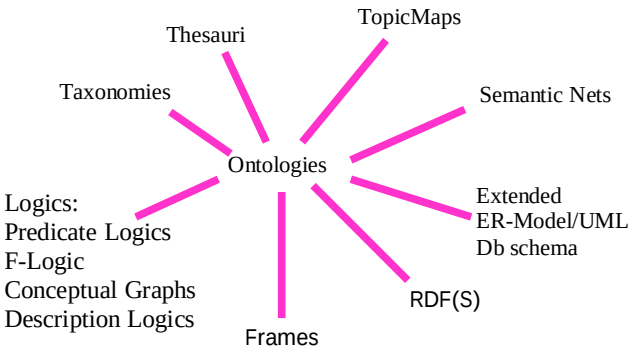
آنتولوژی در هوش مصنوعی و همچنین علوم کامپیوتر به مجموعه‌ای از لغات و فرضیات (عموماً در منطق مرتبه‌ی اول) گفته‌می‌شود که با توجه به معنی آن لغات ایجاد و به منظور توصیف یک واقعیت خاص طراحی‌شده‌اند.

آنتولوژی نقش اصلی را در مبادله اطلاعات و توسعه وب لغوی به سمت وب معنایی دارد. آنتولوژی یک مدل مفهومی است که موجودیت‌های واقعی در یک دامنه خاص و روابط بین آنها را به صورت صریح و رسمی مدل‌سازی می‌کند. ساده‌ترین توصیف از آنتالوژی شامل واژه نامه‌ای[1] از اصطلاحات[2] با معانی و تفسیر آنهاست. همانطورکه در شکل15-2 نشان‌داده‌شده‌است، محدوده تعریف آنتولوژی در مقاله‌های علمی از واژه‌نامه‌های کنترل‌شده تا مدل‌های دامنه‌ای بامعنا و منطقی است. بر اساس این برداشت، دو مولفه اصلی هر آنتالوژی، واژه نامه اصطلاحات و توصیف معنای اصطلاحات است. تفاوت بنیادین بین رویکردهای آنتولوژیکی، در نحوه توصیف روابط بین اصطلاحات است.



شکل15-2: محدوده تعریف آنتولوژی از واژه‌نامه‌های کنترل‌شده تا مدل‌های دامنه‌ای بامعنا و منطقی

نکته: آنتولوژی توصیف فرمال و صریح از واژه‌ها و مفاهیم یک دامنه خاص و ارتباطات میان آن‌هاست. ویژگی های هر یک از این مفاهیم، خصایص مختلف این مفهوم را توصیف می کنند. یک آنتولوژی به همراه یک مجموعه از نمونه های منفرد از کلاس‌ها تشکیل یک پایگاه دانش می‌دهد. بنابراین آنتالوژی یک نوع زبان بازنمایی دانش است(شکل15-3).



شکل15-3: آنتولوژی به عنوان روش بازنمایی دانش

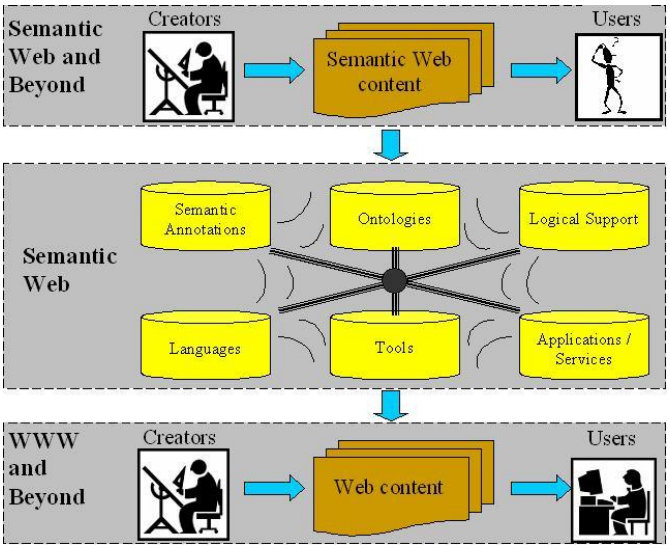
[1] Vocabulary
[2] Term

آنتولوژی و وب معنایی

همانطور که می‌دانیم استفاده از آنتولوژی به عنوان بخش اصلی تمامی برنامه‌های وب معنایی استو به عبارت دیگر قلب برنامه‌های وب معنایی است. طبق تعریف آنتولوژی عبارتست از یک تعریف صریح و رسمی برای درک یک حوزه‌ی خاص. یکی از وقت‌گیرترین مراحل برنامه‌های وب معنایی، ساختن آنتولوژی آن حوزه خاص است که نیازمند مشارکت افراد متخصص در زمینه مهندسی آنتولوژی و همچنین افراد خبره در حوزه‌ی تعریف آنتولوژی است. ساخت آنتولوژی به سه شیوه قابل انجام است:

1) کاملاً دستی و توسط افراد خبره در حوزه ساخت آنتولوژی
2) نیمه خودکار و توسط ابزارها و نرم افزارها و با دخالت انسان
3) کاملا خودکار

به علت وقت‌گیر بودن، هزینه‌ی سنگین و عدم داشتن انعطاف جهت تغییر، انتظار ساخت آنتولوژی به صورت دستی و توسط افراد خبره در آن حوزه، نامعقول است. ساخت کاملاً اتوماتیک آنتولوژی نیز اقدامی هر چند جالب اما در حالت کلی غیرممکن است زیرا به دانش و دخالت افراد خبره در آن حوزه جهت اضافه‌کردن اطلاعات مفید و پیشنهادات سازنده برای ساخت آنتولوژی، تاکیدکردن روی بخش خاصی از اطلاعات و همچنین بهبود و تغییرآنتولوژی ساخته‌شده نیاز است. در نتیجه بهترین حالت، ساخت نیمه اتوماتیک آنتولوژی و با دخالت کم انسان است. از مهمترین فیلدهایی که از آنها در ساخت نیمه‌اتوماتیک آنتولوژی استفاده‌نمود می‌توان به اکتشاف دانش، پردازش زبان طبیعی، بازیابی اطلاعات، یادگیری ماشین، استنتاج و مدیریت پایگاه داده اشاره‌نمود.



شکل15-4: کاربرد آنتولوژی در وب معنایی

یادگیری آنتولوژی

شاخه‌ای از استخراج اطلاعات (Information Extraction) که هدف آن ساخت (نیمه) اتوماتیک آنتولوژی از مجموعه‌ای از اسناد (Corpus) یا دیگر پایگاه‌های داده‌ای است. به طور کلی یادگیری آنتولوژی برقرارکردن یک نگاشت بین اجزای یک آنتولوژی است که برخی از اجزای آن از قبل داده‌شده و بعضی دیگر مجهولند وهدف ما یافتن اجزای مجهول است. یادگیری آنتولوژی می‌تواند از منابع داده‌ای مختلف مانند متن، فرهنگ لغت و یا پایگاه دانش باشد. از منظر استخراج دانش، آنتولوژی نیز نوع دیگری از مدل‌ها و البته پیچیده‌تر از مدل‌های یادگیری ماشین است که می‌بایست توسط یک زبان فرضی بیان شود.

مهندسی آنتولوژی(Ontology engineering)

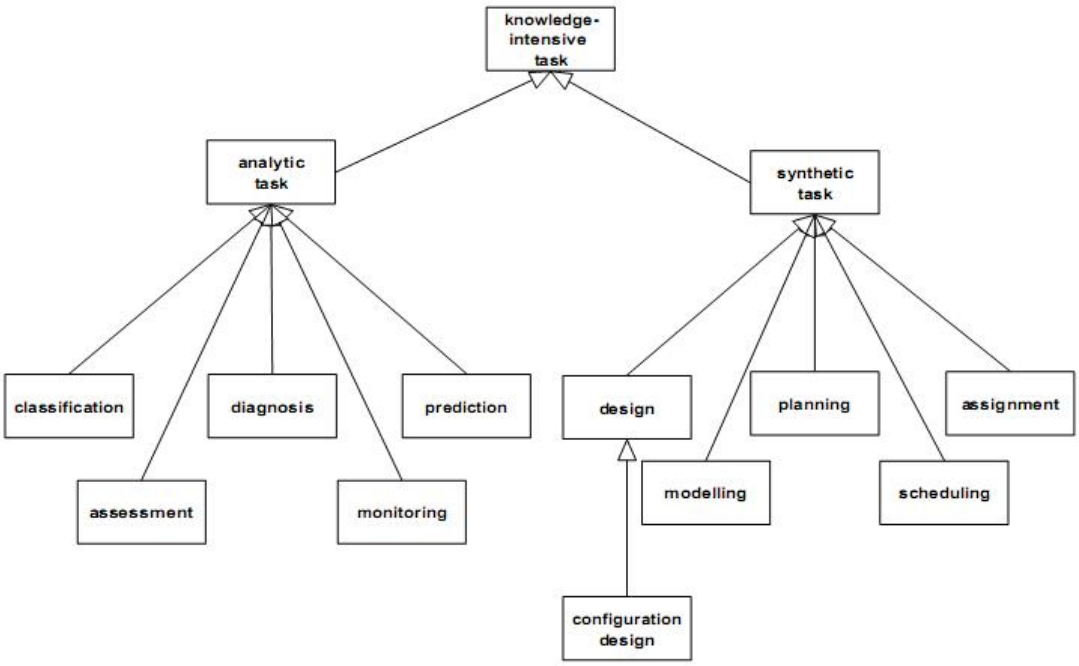
Ontology engineering is the discipline concerned with building and maintaining ontologies. It provides guidelines for building domain conceptualizations, such as the construction of subsumption hierarchies. An important notion in ontology engineering is ontological commitment. Each statement in an ontology commits the user of this ontology to a particular view of the domain. If a definition in an ontology is stronger than needed, than we say that the ontology is over-committed. For example, if we state that the name of a person must have a first name and a last name we are introducing a western bias into the ontology and may not be able to use the ontology in all intended cases. Ontology engineers usually try to define an ontology with a minimal set of ontological commitments. One can translate this into an (oversimplified) slogan: "smaller ontologies are better!". Gruber gives some principles for minimal commitments. Construction of subsumption hierarchies is seen as a central activity in ontology engineering.

انواع آنتولوژی‌ها

1- **Foundational ontologies:** Foundational ontologies stay closest to the original philosophical idea of "ontology". These ontologies aim to provide conceptualizations of general notions, such as time, space, events and processes. Some groups have published integrated collections of foundational ontologies. Two noteworthy examples are the SUMO (Suggested Upper Merged Ontology) and DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering).

2- **Domain-specific ontologies:** Although foundational ontologies are receiving a lot of attention, the majority of ontologies are domain-specific: they are intended for sharing concepts and relations in a particular area of interest. One domain in which a wide range of ontologies has been published is biomedicine. A typical example is the Foundational Model of Anatomy (FMA) which describes some 75,00 anatomical entities. Other well-known biomedical ontologies are the Unified Medical Language System8 (UMLS), the Simple Bio Upper Ontology, and the Gene Ontology.

3- **Task-specific ontologies:** A third class of ontologies specifies the conceptualizations that are needed for carrying out a particular task. For each of the task types listed in Table 14-1 one can specify domain conceptualizations needed for accomplishing this task. In general, conceptualizations of domain information needed for reasoning algorithms typically takes the form of a task-specific ontology. For example, search algorithms typically operate on an ontology of states and state transitions. Tate's plan ontology is another example of a task-specific ontology.



شکل15-5: انواع وظایف(Scriber et al, 2000)

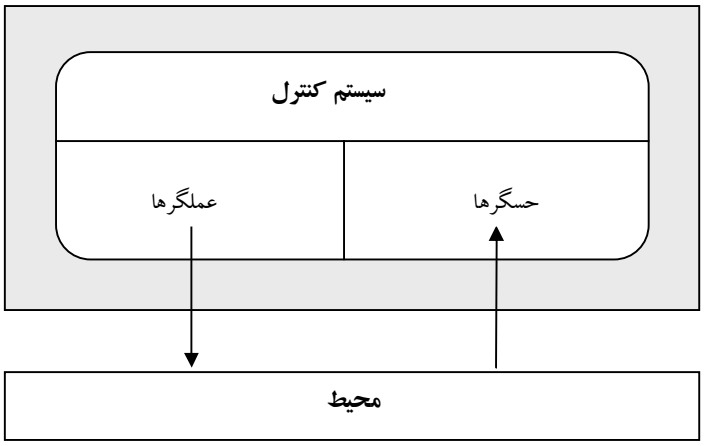
We distinguish two groups of task types: analytic tasks and synthetic tasks. The distinguishing feature between the two groups is the "system" the task operates on. "System" is an abstract term for the object to which the task is applied. For example, in technical diagnosis the system is the artifact or device being diagnosed; in elevator configuration it is the elevator to be designed. In analytic tasks the system preexists although it is typically not completely "known." All analytic tasks take as input some data about the system, and produce some characterization of the system as output. In contrast, for synthetic tasks the system does not yet exist: the purpose of the task is to construct a system description. The input of a synthetic task typically consists of requirements that the system to be constructed should satisfy.

جدول15-1: دسته‌بندی وظایف(Scriber et al, 2000)

Task type	Input	Output	Knowledge	Features
		ANALYTIC TASK TYPES		
Classification	Object features	Object class	Feature-class associations	Set of classes is predefined.
Diagnosis	Symptoms/ complaints	Fault category	Model of system behavior	Form output varies (causal chain, state, label) and depends on use made of it (troubleshooting).
Assessment	Case description	Decision class	Criteria, norms	Assessment is performed at one particular point in time (cf. monitoring).
Monitoring	System data	Discrepancy class	Normal system behavior	System changes over time. Task is carried out repeatedly.
Prediction	System data	System state	Model of system behavior	Output state is a system description at some future point in time.
		SYNTHETIC TASK TYPES		
Design	Requirements	Artifact description	Functions, components, skeletal design, constraints, preferences	May include creative design of components.
Configuration design	Requirements	Artifact description	Functions, components, constraints, preferences	Subtype of design in which all components are predefined.
Assignment	Two object sets, requirements	Mapping set 1 → set 2	Constraints, preferences	Mapping need not be one-to-one.
Planning	Goals, requirements	Action plan	Actions, constraints, preferences	Actions are (partially) ordered in time.
Scheduling	Job activities, resources, time slots, requirements	Schedule = mapping activities → time slots of resources	Constraints, preferences	Time-oriented character distinguishes it from assignment.

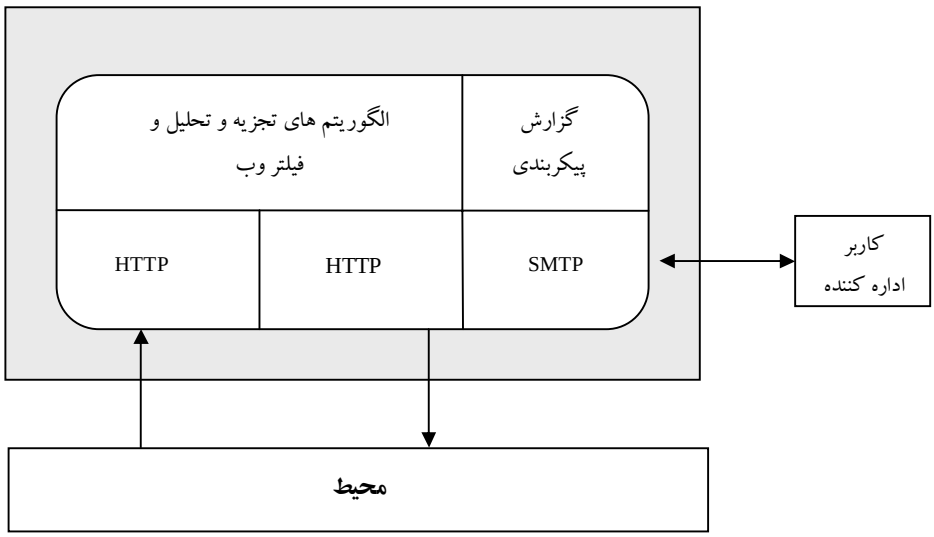
عامل‌ها و سیستم‌های خبره

عامل موجودیتی است که با استفاده از حسگرها، محیط را حس و درک نموده و مطابق با این درک عملی توسط عملگرها به محیط اعمال‌می‌کند. در شکل15-6، آناتومی کلی عامل، نشان‌داده‌شده‌است.



15-6: آناتومی یک عامل

برای مثال عنکبوت وب، عامل مجازی است که در روی شبکه گسترده وب به جمع‌آوری و فیلتر اطلاعات، می‌پردازد. در عامل عنکبوت، حسگرهای اولیه از پروتکل انتقال ابرمتن یا همانHTTP به عنوان ابزاری جهت جمع‌آوری داده‌ها از صفحات وب، به‌کارمی‌رود. سیستم کنترلی عنکبوت وب، یک کاربرد است که تقریباً با هر زبانی می‌تواند نوشته‌شود. این برنامه کاربردی به تجزیه و تحلیل صفحات وب می‌پردازد. در نهایت، عنکبوت وب می‌تواند از طریق پست الکترونیکی با استفاده از پروتکل SMPT با کاربر اداره‌کننده، تبادل‌اطلاعات داشته‌باشد. در شکل15-7، نمای کلی عامل عنکبوت وب، نشان‌داده‌شده‌است.

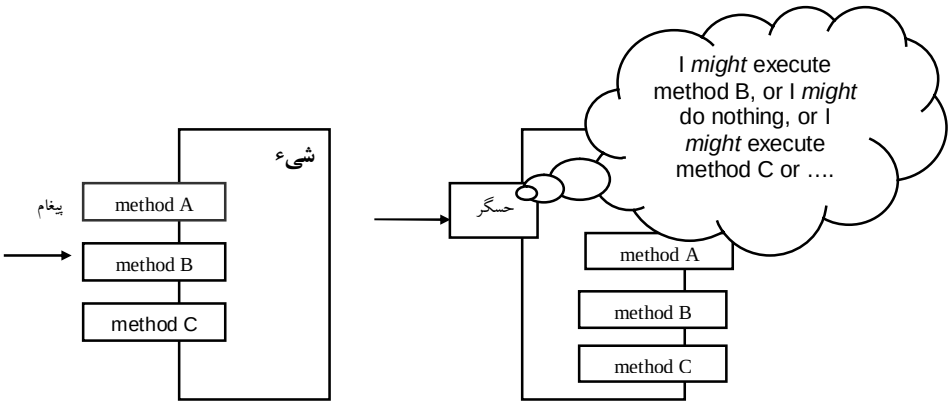


15-7: عامل عنکبوت وب

پرسش: چه شباهت‌ها و تفاوت‌هایی بین عامل‌ها و سیستم‌های خبره وجود دارد ؟

عامل‌ها و شیءها (Agents and Objects)

مفهوم عامل را می‌توان از سایر مفاهیم مشابه مثل اشیاء هم متمایز کرد. برای مثال، عامل‌ها؛ خودمختاری بیشتر، رفتار انعطاف‌پذیرتر، واکنشی‌تر و اجتماعی‌تر از اشیاء دارند. بطور کلی در هر عاملی انتظارمی‌رود حداقل یک ریسمان کنترلی، وجود داشته باشد. این ریسمان کنترلی در انتخاب عمل توسط عامل، کاربرد دارد. البته ممکن است بیشتر از یک ریسمان در عامل وجود داشته باشد. شیء بر خلاف عامل، کنترلی بر روی رفتار خود ندارد. در شکل 15-8، نمای کلی شیء و عامل، نشان‌داده‌شده‌است.



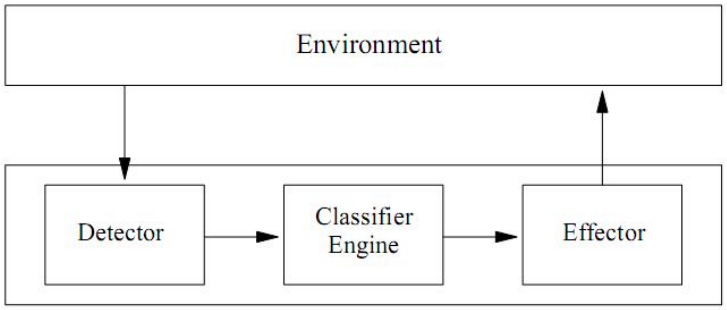
شکل 15-8: نمای کلی شیء و عامل

انواع عامل‌ها (مطابق دیدگاه راسل)

- **Table-driven agents:** Use a percept sequence/action table in memory to find the next action. Implemented by a (large) lookup table
- **Simple reflex agents:** Based on condition-action rules, implemented with an appropriate production system; stateless devices with no memory of past world states
- **Agents with memory:** Have internal state that is used to keep track of past states of the world
- **Agents with goals:** Agents that have state and goal information that describes desirable situations. Agents of this kind take future events into onsideration.
- **Utility-based agents:** Base decisions on classic axiomatic utility theory in order to act rationally

سیستم دسته‌بند (Classifier System)

A system which classifies inputs based on detector messages and associates them with actions (action messages).
A population of classifies (rules) are used that include an antecedent and a consequent. Classifies can be created or modified dynamically.



شکل15-9: سسیستم دسته‌بند ساده ایستا (بدون قابلیت یادگیری)

- The **detector** senses the environment and encodes the input from the environment as a message.
- The **classifier engine** searches the classifier list for the classifier that matches the message most.
- The selected classifier is activated and sends an output (**effector**) to the environment.

پرسش: منظور از دسته‌بند (Classifier) در سیستم دسته‌بند چیست؟

Each classifier has two parts, a condition and an action.

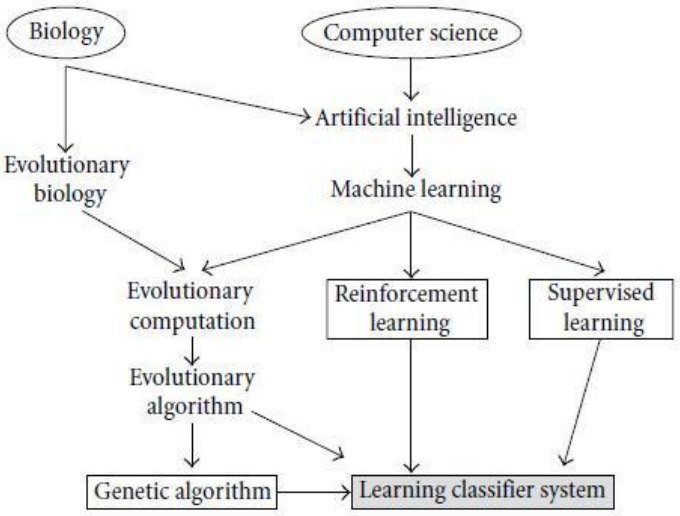
Classifier List									Legend	
	C_0	C_1	C_2	:	A_0	A_1	A_2	Strength	C_0	Hungry
1)	1	#	0	:	1	0	0	1	C_1	Tired
2)	#	#	1	:	0	1	0	2	C_2	In Danger
3)	0	0	0	:	0	0	1	3	A_0	Find Food
4)	0	1	0	:	0	1	0	1	A_1	Find Shelter
5)	1	1	1	:	0	1	0	1	A_2	Wander Around
6)	0	0	0	:	1	0	0	1		

شکل15-10: یک مثال از دسته‌بند

سیستم دسته‌بند یادگیر (Learning Classifier System)

Learning Classifier System (LCS) = Learning + Classifier System

سیستم دسته‌بند یادگیر توسط جان هلند در سال 1975 به منظور مدل‌سازی پیدایش شناخت بر اساس مکانیزم‌های تطبیقی ابداع شد. هسته اصلی LCS مجموعه‌ای از قوانین(جمعیتی از دسته‌بندها) است. LCS از دو مفهوم تکامل و یادگیری بهره‌می‌گیرد. این دو مفهوم به ترتیب توسط دو سازوکار الگوریتم ژنتیک و یک مکانیزم مناسب یادگیری برای حل مسائل مختلف فراهم می‌شود(شکل15-11). هدف اولیه LCS حل مسائلی بود که به نحوه‌ای در تعامل با محیط بودند نظیر حرکت یک روبات در مسیرهای پرپیچ و خم. در LCS محیط، منبع ورود داده‌ها به الگوریتم است. البته اطلاعاتی که از محیط دریافت‌می‌شود وابسته به حوزه مساله مورد نظر است.

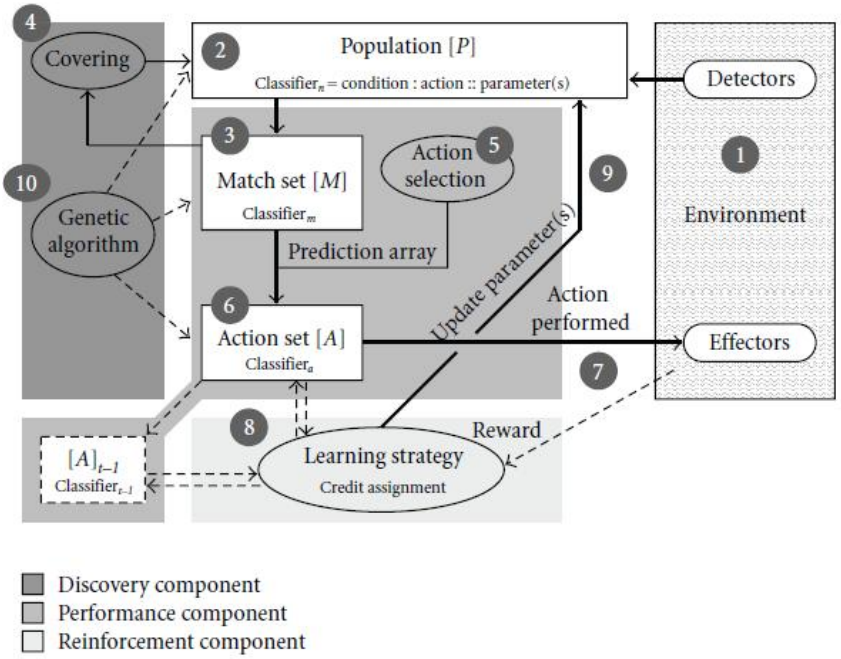


شکل 15-11: روند شکل‌گیری LCS

دو مکانیزمی که باعث حرکت به جلو سیستم دسته‌بند یادگیر می‌شود عبارت است از:

1- **اکتشاف**: عمدتا از طریق الگوریتم ژنتیک فراهم‌می‌شود. اکتشاف به ***"اکتشاف قوانین"*** اشاره‌دارد. به عبارتی دیگر قوانینی را معرفی‌می‌کند که در حال حاضر در جمعیت موجود نیست. در حالت ایده‌آل، قوانین جدید در گرفتن پاداش بهتر از قوانین موجود عمل می‌نمایند. از همان آغاز به کار الگوریتم LCS ، عمل اکتشاف تقریبا همواره از طریق استفاده از یک الگوریتم ژنتیک بوده است. الگوریتم ژنتیک یک تکنیک جستجوی محاسباتی است.

۲- **یادگیری**: در زمینه هوش مصنوعی می‌توان چنین تعریف نمود: یادگیری یعنی بهبود عملکرد در برخی از محیط‌ها از طریق کسب دانشی که ناشی از تجربیات کسب شده در آن محیط‌هاست. یادگیری تقویتی (که از طریق تخصیص اعتبار فراهم می‌شود) یکی از مکانیزم‌های ضروری معماری LCS است.



شکل 15-12: معماری LCS و ترتیب اجرای مراحل آن

هر دسته از جمعیت در LCS علاوه بر *"شرط"* و *"عمل"* دارای یک یا چندین پارامتر اختصاصی نظیر برازندگی است. این پارامترها در طی فرایند یادگیری توسط الگوریتم LCS بروزرسانی می‌شوند. بروزرسانی این پارامترها در پی توزیع پاداش یا تنبیه در بین دسته‌های که دخالتی در کسب آن داشته‌اند، روی‌می‌دهد. مکانیزم یادگیری به منظور دو هدف اصلی است:

- شناسایی دسته‌هایی که برای کسب پاداش در آینده مفید هستند.
- تشویق برای کشف قوانین بهتر

نکته: در پیاده‌سازی‌های مختلف LCS از سیاست‌های یادگیری مختلفی استفاده‌شده‌است و دلیل اصلی آن است که مسائل مختلف در حوزه‌های مختلف، نیازمند شیوه‌های یادگیری مختلفی هستند.

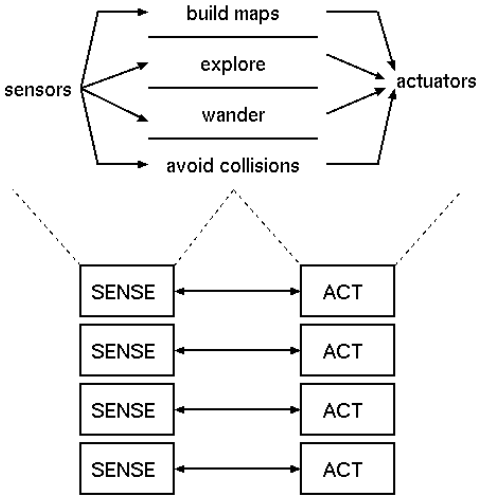
معماری‌های عامل‌های هوشمند

پرسش: منظور از معماری عامل (Agent architectures) چیست؟

Agent architectures: Internal representation and reasoning capabilities of an agent

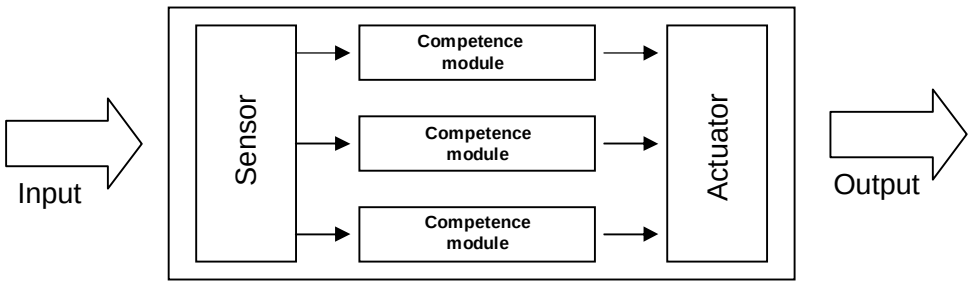
1- معماری سلسله‌مراتبی بروکس (Brooks)

در شکل15-13، نمای کلی از معماری بروکس، نشان‌داده‌شده است. این معماری به صورت لایه‌ای و سلسله مراتبی است که هر لایه متناظر با یک ماژول است. لایه‌های بالاتر، می‌توانند رفتارهای لایه پایین‌تر را بازنویسی کنند.



شکل15-13: نمای کلی از معماری بروکس

از مزایای چنین عامل‌های واکنشی: سادگی، سرعت بالا در عکس العمل، اقتصادی بودن و استحکام است. همچنین رفتارهای پیچیده را با ترکیب گروهی از رفتارهای ساده، مدل‌می‌کند. در شکل15-14، معماری عامل‌های واکنشی سلسله مراتبی نشان‌داده‌شده‌است.



شکل15-14: معماری عامل واکنشی سلسله مراتبی

با این حال، معماری واکنشی برای ساخت عامل‌های واکنشی داری معایب و محدودیت هایی است نظیر اینکه فرایند یادگیری در آن مشکل است و معمولاً فاقد یادگیری است. مدیریت و مهندسی رفتار عامل‌ها وقتی که تعامل مولفه‌ها و محیط زیاد باشد، برای طراح عامل، بسیار پیچیده خواهد بود. به عبارت دیگر، این نوع معماری فاقد متدولوژی است، بنابراین طراحی، تحلیل و مهندسی آن برای سیستم‌های و محیط‌های پیچیده، سخت و مشکل است.

2- معماری سطح دانش

Knowledge-level architectures: A deliberative agent's knowledge determines its behavior in accordance with Newell's Principle of Rationality, which states that:
if an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action.

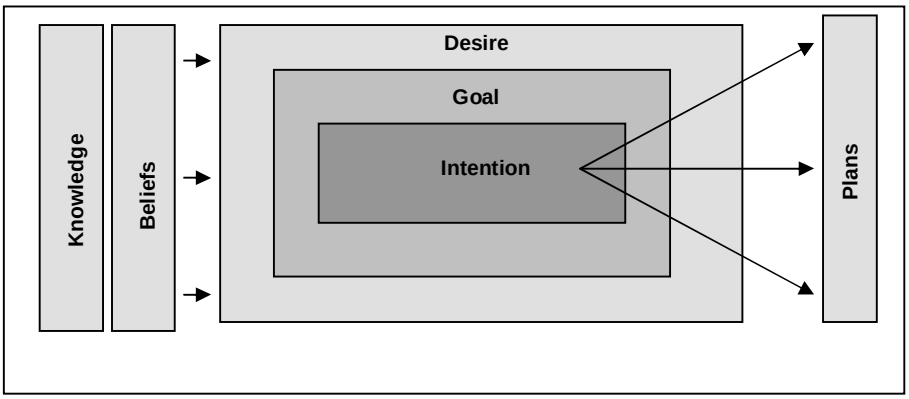
One of the most important manifestations of this approach is known as the **beliefs–desires–intentions (BDI)** architecture.

معماری BDI به عنوان یک معماری شناختی مبتنی بر دانش دارای ویژگی‌های زیر است:

1- دارای قصد و نیت[1] است؛ هر عملی به منظور رسیدن به هدف خود انجام‌می‌دهد.

2- عاقلانه[2] رفتارمی‌کند؛ همیشه در انجام هر عمل، اصل عقلانیت را جهت رسیدن به هدف دنبال می‌کند.

3- بازنمایی دانش؛ معمولاً نمایش منطقی از محیط برای هدایت روند استدلال، استفاده‌می‌شود.



شکل15-15: معماری BDI

معماری BDI دارای دو پروسه اصلی است؛ 1- تصمیم‌گیری برای اینکه عامل در رسیدن به هدف به چه چیزی نیاز دارد؟ 2- تصمیم‌گیری برای اینکه عامل چگونه به هدف خود برسد. معماری شناختی BDI دارای سه مولفه(مدل درونی) باور[3]، مطلوب[4] و مقصود[5] است. باور عامل، شامل دانش عامل از محیط پیرامون و خود عامل است. هنگامی‌که محیط تغییرمی‌کند، عامل سعی به جمع‌آوری تجارب نموده و سازگاری باورهای خود را با تجارب جمع‌آوری‌شده، بررسی‌می‌کند و در صورت لزوم، باورها را تغییرمی‌دهد. عامل BDI، مبتنی بر باور جاری خود، تعدادی انتخاب[6] ایجادنموده و یکی از آنها را به عنوان هدف یا مطلوب انتخاب‌می‌نماید. پس از آن، یکی از چندین مسیر ممکن را برای رسیدن به هدف، انتخاب‌می‌کند(برآورده شدن مطلوب). مسیر انتخاب‌شده توسط عامل، مقصود عامل را بازنمایی‌می‌کند.

نکته: تئوری BDI فرایند استدلالی شبیه انسان، فراهم‌می‌کند؛ نخست تصمیم‌می‌گیرد به چه چیزی می‌خواهد برسد، سپس یک راه و روشی از انتخاب‌های ممکن انتخاب‌می‌کند و در نهایت عملی را در جهت رسیدن به آن، انجام می دهد. از نظر شهودی، این تئوری به سادگی قابل درک است. با این وجود، برای بازنمایی منطقی مسأله، دو موضوع مهم مطرح است:

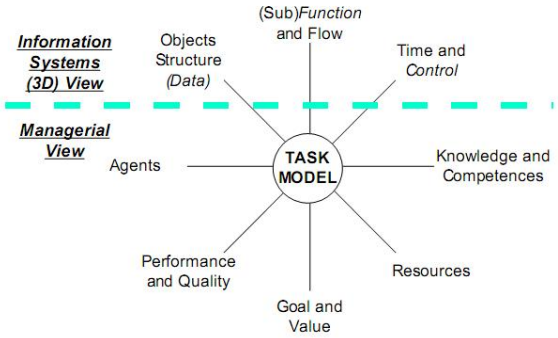
1- چگونگی مسأله دنیای واقعی به توصیف منطقی، تبدیل شود؟

2- چگونه اطلاعات موجودیت‌های دنیای واقعی و پروسه‌ها، در قالب منطقی بازنمایی شوند؟

[1] Intentional
[2] Rational
[3] Belief
[4] Desired
[5] Intention
[6] Option

تمرین‌ها

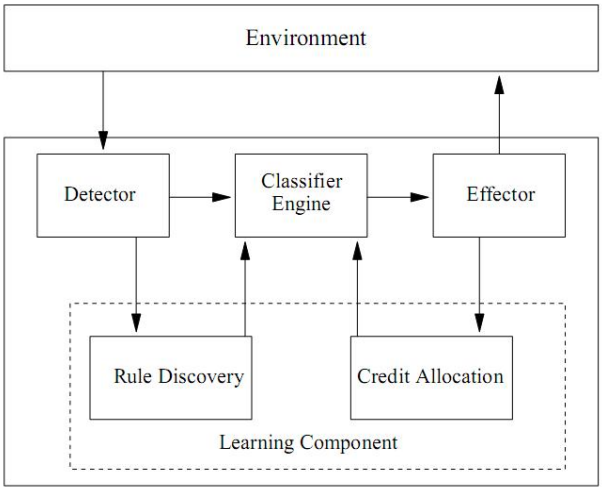
تمرین15-1: در متدلوژی کامن‌کدس نمای کلی مدل وظیفه (Task Model) مطابق شکل 15-16 است. با یک مثال این مدل وظیفه را تشریح کنید.



شکل15-16: نمای کلی از مدل وظیفه کامن کدس(Scriber et al, 2000)

تمرین15-2: سیستم دسته‌بند(ساده و یادگیر) را با مفهوم سیستم خبره مقایسه کنید. چه شباهت‌هایی دارند؟

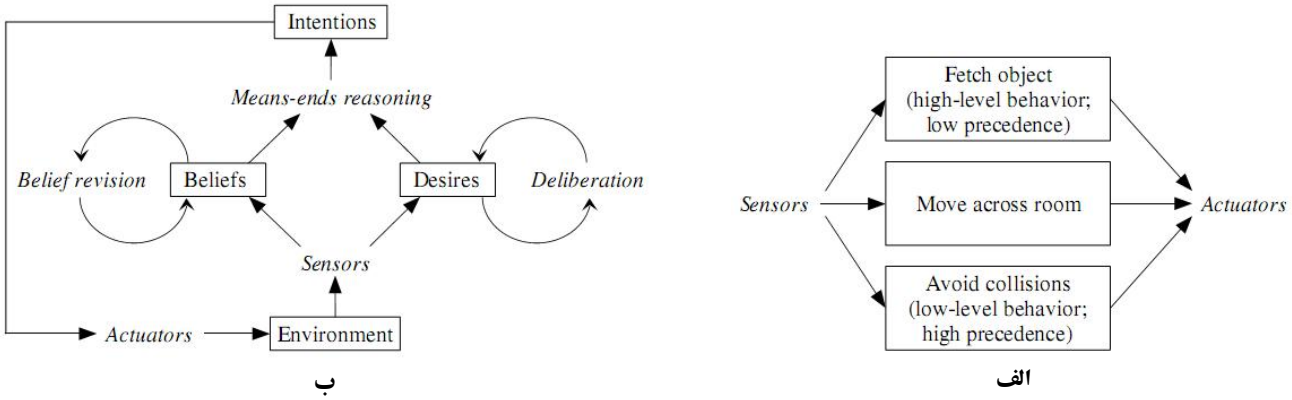
تمرین15-3: با یک مثال روند کار سیستم دسته‌بند یادگیر مطابق شکل 15-17 را توضیح‌دهید.



شکل15-17: سسیستم دسته‌بند یادگیر

تمرین15-4*: برای LCS سبک‌های مختلفی ارائه‌شده‌است. این سبک‌ها را نام برده و توضیح‌دهید تفاوت اصلی آنها در چیست؟

تمرین15-5: با دو مثال یکی برای شکل 15-18 الف و دیگری برای شکل 15-18ب چگونگی کارکرد این دو معماری را توضیح‌دهید.



شکل15-18: **الف)** معماری سلسله‌مراتبی Brooks **ب)** معماری BDI (Hopgood, 2011)