

سیستم های خبره

پروژه دات کام

www.Prozhe.com

پیشگفتار

نقطه آغاز هوش مصنوعی اندکی بعد از جنگ جهانی دوم می باشد در آن زمان وانیبیا توجه به مسائل یسبرنتیک ، زمینه را برای پیشرفت هوش مصنوعی به وجود آورد. در سال 1950 (آلن تیورینگ) آزمایشی مینی بر اینکه آیا ماشین قادر است با فرآیندهای مغز انسان رقابت نماید مطرح کرد در سال 1958 در کالج دارتموت جلسه ای برگزار شد

فصل اول

تاریخچه مختصری از سیستم های خبره و

هوش مصنوعی

فصل دوم

مفاهیم اساسی سیستم های خبره

فصل سوم
نمایش دانش

پکیده مؤلف

هوش مصنوعی دانش ساخت ماشین ها یا برنامه های هوشمند است تعریف دیگری که از هوش مصنوعی می توان ارائه داد به قرار زیر است :

هوش مصنوعی شاخه ای است از علم کامپیوتر که ملزومات محاسباتی و اعمالی همچون ادراک (perception) - استدلال (reasoing) - یادگیری (Learning) را بررسی می کند و سیستمی جهت انجام چنین اعمالی ارائه می دهد. و همچنین مطالعه روش هایی است برای تبدیل کامپیوتر به ماشینی که بتواند اعمال انجام شده توسط انسان را انجام دهد. آیا تنها این نکته که هوشمندترین موجودی که می شناسیم انسان است کافی است تا هوشمندی را به تمام اعمال انسان نسبت دهیم اما حداقل توجه به این نکته کاملاً واضح ، لازم است که برخی از جنبه های ادراکی انسان مثل شنیدن و دیدن کاملاً ضعیف تر از موجودات دیگر است .

هوش مصنوعی فن آوری پیشرفته ای است که اخیراً از سوی محققان علوم مختلف مورد توجه زیادی قرار گرفته است و هوش مصنوعی در واقع روشی است که زیربنای لازم برای انجام عملیات هوشمند توسط انسان توسط کامپیوتر را فراهم می آورد. یکی از کاربردهای عمومی هوش مصنوعی سیستم های خبره هستند. سیستمهای خبره ابزارهای کامپیوتری هستند که همانند یک متخصص در حوزه تخصصی خود در مسائل مربوط به آن حوزه مشاوره می دهند و در صورت لزوم تصمیم گیری می نمایند . کاربردهای اصلی هوش مصنوعی عبارتند از :

الف) پردازش زبان طبیعی (ترجمه ، خلاصه سازی ، محاوره زبان طبیعی)

ب) اثبات قضایا (اثبات وجود / عدم وجود ویژگی های مورد نظر)

ج) سیستم های بنیادی (د) پردازش صوت و تصویر

ه) برنامه ریزی و هدایت رباط (و) سیستم های خبره (ز) عامل های نرم افزاری

در طراحی و ساخت سیستم های خبره افراد زیر به صورت گروهی با هم کار می کنند.

الف) متخصص دامنه: کسی که دانش و فن کافی در زمینه خاصی را دارد.

ب) مهندس دانش: شخصی که طراحی، ساخت و آزمایش سیستم خبره را به عهده دارد.

ج) کاربر: شخصی که با سیستم کار خواهد کرد و پذیرش نهایی سیستم بستگی به نظر او دارد.

اما از مزایای سیستم های خبره می توان به موارد زیر اشاره کرد:

1- همواره در دسترس است در حالی که متخصص تنها عموماً در ساعات خاصی از شبانه روز قابل دسترس می باشد.

2- از نظر جغرافیایی متخصص در هر لحظه، تنها می تواند در یک محل حضور داشته باشد در حالی که یک سیستم خبره قابل نسخه برداری بوده و در هر لحظه می تواند در چندین محل از آن استفاده کرد.

3- متخصص ممکن است بر اثر عواملی گوناگون مانند، ترک سازمان، بازنشستگی، مرگ و میر و... پس از مدتی امکان ادامه فعالیت نداشته باشد و غیر قابل جایگزین باشد در صورتی که سیستم خبره قابل جایگزین است.

4- سیستم های خبره معمولاً رفتارهای یک نواختی دارند و تحت تأثیر زمان و مکان قرار نمی گیرند.

5- در انجام کارهای تکراری عموماً سرعت بالاتری دارند یک معماری سیستم های خبره از

قسمت های زیرین تشکیل شده است: (الف) پایگاه دانش

(ب) موتور استنتاج (ج) واسط سازنده (و) برنامه های جانبی (ز) تشریح نتایج

با سپری شدن 5 نسل الکترونیکی از لامپ خلاً تا اختراع و استفاده روزافزون از تراشه (spu) بشر خلاق و مبتدع در صدد شکافتن علوم انفورماتیک و دسترسی به نسل ششم یعنی رایانه های هوشمند است. در زمینه توسعه بازیها در هوش مصنوعی و سیستم های خبره تا حدی به بازی شطرنج پرداخته شده که غالباً عده ای هوش مصنوعی را با شطرنج همزمان به خاطر می آورند. مک کارتی از بنیان گزاران هوش مصنوعی است این روز را آنقدر اغراق آمیز می داند (زیرا این روز آنقدر سرعت در حال پیشرفت است) که می گوید: محدود کردن هوش مصنوعی به شطرنج مانند این است که علم ژنتیک را از زمان داروین تا کنون تنها محدود به پرورش لوبیا کرده باشیم. اما در مورد سیستم های خبره یا همان هوش مصنوعی در زمینه کاربردی تا کنون به کار گرفته شده است که تعدادی از این زمینه ها عبارتند از: کشاورزی، تجارت، شیمی، ارتباطات، کامپیوتر، آموزش الکترونیک، مهندسی محیط، زمین شناسی، تصویر پردازش، اطلاعات، حقوق، ساخت و ساز، ریاضیات، پزشکی، هواشناسی و ...

در ساخت سیستم های خبره و طراحی آن افراد باید به صورت گروهی کار کنند سپس می بینیم که توجه به سیستم های خبره در عصر تکنولوژی و پیشرفت برای عقب نماندن از دنیای متمدن امروزی لازم است.

مقدمات هوش مصنوعی

تحقیق در مورد هوش مصنوعی در دهه 1940 که مصادف با پدیدار شدن اولین نسل از کامپیوترها در مراکز تحقیقاتی بود، شروع شد.

اصول اولیه ماشین منطقی که در ریاضیات پایه ریزی شده بود با تلاش کرت گاول، آلوتز و چرچ، آلن تورینگ مهیا شد. همچنین روایت هدراسل در سال 1913 به تولید روشهای صوری در استدلال منطقی کمک های شایانی کرد. تحقیق عمده همه این افراد روی موضوع حسابهای احتمالی و پیشنهادی متمرکز شد.

این منطق صوری و نمادین به عنوان یک بخش مهم و قابل توجه در سیستم های هوش مصنوعی نشان داده شد. تورینگ در سال 1950 میلادی ماشینی را توسعه داد که ثابت می کرد یک پردازشگر ساده عددی می تواند علامتها را به خوبی اعداد بدون نقص و صحیح، پردازش کند و به کار ببندد. او این مسأله را که برای مقایسه هوش ماشین و هوش انسان مورد استفاده قرار گرفت در آزمایشی به نام تست تورینگ آشکار ساخت. بدین ترتیب تا سال 1956 میلادی که جان مک کارتی برای نخستین مرتبه اصلاح هوش مصنوعی را در یک کنفرانس (که در کالج دارت موث آمریکا برگزار شد) مورد استفاده قرار داد، اصطلاح هوش مصنوعی وجود نداشت.

دانشمندان متعددی که مشغول کار و بررسی این زمینه نوین مطالعه بودند در کنفرانس گرد هم آمدند و تصمیم آنها برای سازماندهی جدید این علم چنان بود که اعلام کردند در 25 سال آینده،

کامپیوترها تمام کارها را انجام خواهند داد و فعالیت های ما تنها به کارهای تفریحی محدود خواهد شد. و 25 سال بعد، همین زمانی است که ما در آن هستیم و پرواضح است که این قبیل پیشگویی ها از روی خوش بینی بوده و هوش مصنوعی این توقعات را برآورده نکرده است. یکی از عمده ترینو قابل توجه ترین برنامه های هوش مصنوعی منطق تئوری یا منطق علوم نظری نوول، شاو و سایمون در سال 1963 بود. این برنامه قادر بود 38 قضیه از 52 قضیه وایت هل راسیل (سال 1913) را در ریاضیات پایه اثبات کند. تعدادی از این برنامه ها در یک تراز، صلاحیت استعداد دانش آموزان فارغ التحصیل نشده را در ریاضیات نشان می دادند و باید گفت هنگامیکه زبان کامپیوتری را براساس منطق بولین (دوگانه) در نظر بگیریم، این امر کاری خارق العاده نمی باشد.

برنامه های هوش مصنوعی برای حل مسائل عمومی

بعد از سال های 1950 میلادی، محققین هوش مصنوعی سعی نمودن روش هایی برای حل مسأله براساس استدلال های بشر، ارائه نمایند. چنین پروژه ای در سال 1976 به وسیله نوول و سایمون توسعه یافت که به عنوان الگوریتم های حل مسائل عمومی یا (GPS) شناخته شد. یک مسأله مشخص در GPS با اصطلاحاتی از قبیل حالت های موجود، حالت اولیه و حالت نهایی توصیف می شود. به عنوان مثال در بازی شطرنج، عملگرها، قوانینی هستند که مهره های مختلف را قادر می سازند از یک حالت به حالت دیگر منتقل شوند. مسائل این چنینی بایک جستجوی فضای حالت مشخص می شوند. مسئله فوق یکی از مسائل اصلی هوش مصنوعی در کاربردهای امروزی می باشد که بعداً راجع به آن بحث خواهد شد.

GPS در زمینه های گوناگون حل مسئله از قبیل اثبات قضایا ، محاسبات انتگرال و انواع خاص از جداول منطقی موفق بود. با وجود این موفقیت ها به دو دلیل این استراتژی عمومی رد شد : اول، جستجوی فضاهای مسأله ها، رشد یافته ، خیلی عمومی و کلی شد که انفجار ترکیبی نام گرفت . دومین مشکل، بیان چگونگی حل مسأله بود ، مسائل ریاضی نسبتاً به راحتی نمایش و توصیف می شدند اما بیان و نمایش مسائل زندگی روزمره بسیار سخت و باورنکردنی بود. بنابراین در سالهای 1960 میلادی ، تحقیقات هوش مصنوعی به جهتی سوق پیدا کرد که الگوریتم های جستجو و تکنیکهای بیان دانش بهبود یابد .

پیدایش KBS

یکی از کمبودهای راه حل مسائل عمومی این بود که اندازه مسئله بزرگ شد. بنابراین فضای جستجو به وجود آمده بطور قابل توجهی رشد کرد بنابراین تنها با ساخت برنامه هایی که کمتر عمومی هستند و تمرکز روی دانش خاص مسئله می توانیم این قبیل فضای جستجو را کاهش دهیم. از این رویک عرصه جدید برای تحقیق در سال 1970 پدیدار شد و واترمن (1986) یک برنامه هوشمند با عالی ترین کیفیت و دانش مشخص در دامنه مسئله ایجاد نمود که لنات و گودها بعداً در سال 1991 آن را اصل دانش نامیدند. آنها این مسئله را به شرح زیر بیان کردند: اگر برنامه ای کار پیچیده ای را به خوبی اجرا کند، آن برنامه می بایست راجع به محیطی که در آن عمل می کند، دانسته هایی داشته باشد. در نبود دانش، همه آنچه که مشخص می شود بر اساس جستجو و استدلال است که کافی نیست . از زمانی که اولین محصول پایگاه های دانش پدیدار شد ، یک شاخص در محدوده های زندگی واقعی به حساب می آمد. مانند تشخیص بیماری های عفونی و یا پیشگویی ذخایر معدنی در مناطق جغرافیایی مختلف دنیا. این تاریخچه ای از آزمایش بر روی مسائل

زندگی واقعی بود تا بفهمیم که آیا تصورات با سعی و تلاش قابل دسترسی هستند یا خیر؟ سیستم DENDRAL اولین سیستم در این دسته بود که ساخته شد. کارروی این سیستم در سال 1965 میلادی بامدریت ادوارد فیگن باوم (1982) شروع شد. این سیستم به این دلیل به کار گرفته شد تا ساختارهای شیمیایی ذرات ناشناخته را معین کند. این سیستم ها برای حل مسائلی به کار برده شدند که نیاز به سرویس دهی یک خبره داشتند بنابراین به عنوان سیستم های خبره شناخته شدند. همچنین این سیستم ها به عنوان سیستم های مبتنی بر دانش یا سیستم های دانش شناخته می شوند. به این دلیل که سیستم ها دارای دانش مربوط به محدوده هایی خاص می باشند. سیستم های خبره تجربی (مبتنی بر تحقیق) در جدول 1-1 آمده است.

جدول 1-1 برخی سیستم های خبره تجربی

نام	مبدأ	سال	محدوده کار
DENDRAL	دانشگاه استانفورد آمریکا	1965	ساختارهای مولکولی داده شده راروی اجزاء تشکیل دهنده کشف می کند.
MACSYMA	هیت - آمریکا	1968	یک برنامه وسیع جذاب که انواع متفاوت از مسایل ریاضی را حل می کند و حساب انتگرال را در بر می گیرد
PROSPECTOR	موسسه تحقیقاتی استانفورد	1974	زمین شناسان در اکتشافات معدنی کمک می کند. همچنین

			می تواند یافته های زمین شناسان را در منطقه مهم پیشگویی کند
<i>MYCIN</i>	دانشگاه استانفورد آمریکا	1976	یک سیستم پزشکی بی سابقه برای کمک به پزشکان در انتخاب آنتی بیوتیک هایی برای کنترل عفونت های پیشرفته و شدید.
<i>XCON</i>	<i>DEC</i>	1980	در ساختار کامپیوترهای بزرگ <i>Dec Vax</i> کاربرد دارد

ظهور سیستم های خبره تجاری

بسیاری از سیستم های خبره که در سال 1970 میلادی ساخته شده بودند، آزمایشی بودند و اساساً به تحقیق دانشگاهی محدود می شدند. سیستم های خبره تجاری تا سالهای 1980 میلادی وجود نداشتند و محققان شروع به تبدیل سیستم ها از نوع آزمایشگاهی به سیستم های تجاری کردند. سیستم *XCON* (مک در موت در سال 1982) از جمله اولین این سیستم ها بود. *DEC* این سیستم را در اوایل دهه 1980 میلادی ساخت که نمونه ای از یک سیستم خبره موفق است. این سیستم قادر است براساس مبالغ هنگفت سرمایه پس انداز شده یک شرکت امکان برگشت سرمایه و سود را در کوتاه مدت محاسبه نماید.

XCON یک پیکربندی مشتری با دقت 98٪ در مقایسه با 70٪ برای انسان دارد و کاری را که انسان در یک زمان متعارف باید به طور کامل انجام دهد، انجام می دهد. موفقیت XCON شروع یک پیشرفت تجاری برای سیستم های خبره بود. در اواخر سالهای 1980 میلادی شرکت های بزرگ و کوچک به ارائه قابلیت های سیستم خبره اقدام نموده و ترقی کردند. سرمایه گذاری در دو برنامه مهم هوش مصنوعی کرد. این اتفاقات در آن زمان باعث ایجاد منافع تجاری با استفاده از هوش مصنوعی و سیستم های خبره شد.

هوش مصنوعی در قرن بیست و یکم

هم اکنون چندین پروژه هوش مصنوعی وجود دارند که جالبترین آنها Cyc است. (لنات و گودها در سال 1991 میلادی) که در سال 1984 میلادی به راه افتاد. Cyc از کلمه Encyclopaedia گرفته شده است. CYC به ذخیره مقدار زیادی از دانش واقعی مبادرت می ورزد. این پروژه در شرکت تکنولوژی کامپیوتر و میکروالکترونیک (MCC) در استین و تگزاس پایه گذاری شده و یکی از جاه طلبانه ترین پروژه هاست که در هوش مصنوعی به عهده گرفته شده است. راهنما و مؤسس پروژه، داگ لنات نقل می کند که: علت عقب ماندن CYC این است که اولین نسل از سیستم های خبره فاقد حواس 5 گانه هستند و این امر موجب می شود که آنها شکننده (ضربه پذیر) باشند. به این معنی که بسیاری از سیستم های خبره نسل اول خیلی خوب عمل کردند اما صرفاً به کرانه های قلمرو دانش خود محدود شدند. این سیستم ها به طور مشهود قادر به پاسخگویی در جهان واقعی نخواهند بود و از این رو سیستم های انسانی بیشتر مورد استفاده قرار می گرفتند. چرا که آن سیستم ها هیچ مزیتی نداشتند.

جدول 1-2: پیشرفت سیستم های فبره از سال 1990 تاکنون

نام	توسعه دهنده	سال تکمیل	شرح
<i>GPSS</i>	<i>NASA, USA</i>	1993	یک سیستم خبره زمانبندی که چرخه عملیات را برای فضاپیمای شاتل از یک پرواز تا پرواز بعدی زمانبندی می کند.
<i>NSSP</i>	<i>Nippon steel, Japan</i>	1992	طرح یک سیستم خبره که برای نیازمندیهای مشتری بکاربرده شده و بر مبنای استدلال عمل می کند.
<i>FRAUDWATCH</i>	<i>Touche Ross, UK</i>	1992	سیستم خبره نمایش که توسط بانک <i>Barclays</i> برای تشخیص کلاهبرداری در کارتهای شناسایی استفاده می شود.
<i>DART</i>	<i>DARPA, USA</i>	1990	سیستم خبره که برای برنامه ریزی منطقی بکار می رود.
<i>LINKMAN</i>	<i>Blue Circle pic, UK</i>	1991	سیستم خبره کنترل فرآیند که برای کنترل انرژی مصرفی در تولید بکار می رود.

سیستم های خبره بعد از گذشت 15 سال سود قابل توجهی برای سازمان ها آورده اند. برخی از محیط های موفق در زیر آمده است.

◀ تشخیص (سیستم MYCIN)

◀ طراحی (سیستم NSPP)

◀ برنامه ریزی (سیستم DART)

◀ پیکربندی (سیستم XCON)

◀ زمانبندی (سیستم GPSS)

انسان متخصص در مقایسه با سیستم های فبره

با وجود این موفقیت ها، چند مثال از شکست های سیستم خبره وجود دارد. کاربرد مناسب سیستم های خبره برای یک پروژه خاص در فصل های بعدی بحث خواهد شد. به هر حال مزایا و معایب سیستم های خبره و انسان متخصص می تواند به صورت جدول 1-2 خلاصه می شود.

مزایای کامپیوتری

◀ انسان متخصص موقت است. به عنوان مثال انسانها ممکن است تغییر شغل بدهند و یا بیمار شوند و غیره ولی کامپیوتر دائمی است.

◀ انسان متخصص همیشه ثابت قدم نیست. چرا که انسان می تواند روزهای تعطیلی داشته باشد یا بعضی اوقات برنامه زمانی کاملاً مشغول داشته باشد. و همه این عوامل ناسازگار روی انجام کارها تأثیری گذارند. کامپیوترها همیشه هستند و با هر گونه شرائط ناسازگار بطوریکه نواخت کاری کنند.

◀ انسان متخصص به راحتی قابل انتقال نیست و یک انسان بطور همزمان نمی تواند در دو مکان حضور داشته باشد ولی کامپیوتر نسبت به انسان راحتتر قابل انتقال است. به عنوان مثال اجرای یک سیستم خبره در یک کامپیوتر می تواند در یک سایت متفاوت روی کامپیوترهای دیگر نیز صورت گیرد و یا حتی از اینترنت بار گردد.

◀ هزینه انسان متخصص زیاد است . حقوق ماهیانه کارمندان ازبهای کامپیوتر شخصی ، سخت افزار کامپیوتر و نیز نرم افزار آن بیشتر است .

مزایای انسان

◀ انسان ها خلاق هستند اغلب اوقات الهام بخش هستند ولی کامپیوترها چنین نیستند.
◀ انسان ها انعطاف پذیر هستند و به آسانی خود را با شرایط وفق می دهند و یا دانش و تخصص خود را با قلمرو دانش خود بکار می بندند و تنها در یک محدوده خاص از مسایل تمرکز دارند.
◀ گرچه سیستم های خبره قابلیت یادگیری دارند اما در مقایسه با انسان قدرت یادگیریشان بسیار ضعیف می باشد.

فوائد سیستم های فبره

فوائد سیستم های خبره می تواند ، هم در سازمان وهم در انجام کارهای انفرادی و شخصی درون سازمان مشاهده شود .

مزایای سازمانی سیستم های فبره

◀ **نگهداری دانش :** دانش همیشه موجود است . برخلاف انسان متخصص که ممکن است جریان کار را عوض نموده و یا کناره گیری کند و غیره.
◀ **توزیع دانش :** دانش می تواند در شرکت های مختلف و یا در جای دیگر دنیا توزیع شود و این کار با به کارگیری شبکه ها و با تکثیر سیستم های خبره روی سخت افزار انجام می شود.

◀ **آموزش:** در شرح قابلیت های سیستم خبره ، کاربران می توانند زنجیره استدلال تصمیماتشان را ببینند و درکی بهتر از مسأله بدست آورند.

◀ **رقابت :** سیستم های خبره معمولاً به شرکت ها یک لبه رقابت آمیز می دهند که باعث افزایش سرعت پاسخ دهی و دقت تصمیمات و غیره می شود.

◀ **کاهش قیمت :** قیمت ایجاد اجزا به وسیله کاربر پایین تر است. چون کامپیوترها هزینه زیادی را برای به اشتراک گذاری دانش خود صرف نمی کنند.

مزایای فردی سیستم خبره

◀ **دسترسی به دانش :** متخصص خبره همواره بر روی سخت افزار هر کامپیوتر قابل استفاده است.

◀ **آموزش :** ارزش آموزش می تواند یک مزیت برای کارفرما و کارمند باشد.

◀ **سازگاری :** کاربر خواهد دانست که خبره جایز الخطا نیست و این طور نیست که سیستم خبره روزهای تعطیل داشته یا احساس بیماری کند و یا اینکه خیلی چیزهای دیگر در فکرش باشد.

سیستم های خبره به عنوان سیستم های پشتیبانی تصمیم گیری

سیستم های پشتیبانی تصمیم گیری (DSS)¹ یک شاخه از تحقیق در عملیات (OR)² هستند که به استنتاج نیاز دارند و برای مسایل مدیریت پیچیده تکنیک های کمی را بکار برده ، به محاسبات طولانی و پیچیده نیز نیاز دارند. سیستم های پشتیبانی تصمیم گیری (DSS) رسماً به عنوان کاربرد OR در مدیریت مراحل مختلف تصمیم گیری تعریف شده اند . مشخصات اصلی سیستم های پشتیبانی تصمیم گیری (DSS) به شرح زیر است :

◀ استفاده از تکنیک های OR به وسیله نرم افزارهای پردازش اطلاعات .

◀ اصلاح فرآیند تصمیم گیری در مدیریت با گسترش توانایی مدیرانی که این تصمیمات

1-Decision Support System

2-Operating Research

را گرفته اند.

◀ کمک کردن به مدیریت به وسیله بررسی مسائل غیرساخت یافته (مسائلی هستند که نمی توانند صرفاً با تکنیکهای OR ساخت یافته حل شوند).

تفاوت بین سیستم های فبره و سیستم های پشتیبانی تصمیم گیری

اکثر افراد DSS و سیستم های خبره را باهم اشتباه می کنند. DSS از کاربرد and/or برای پردازش داده توسط تکنیک های OR پشتیبانی می کند. سیستم های خبره در این موارد خودشان می توانند تصمیمات را اتخاذ کنند. همچنین مشخصه های دیگری برای سیستم های خبره به شرح زیر وجود دارد:

◀ راه حل های خبره برای مسائلی که احتیاج به تکنیک های بهینه سازی ریاضی دارند، کاربرد ندارند ولی این راه حل هادرسیستم های پشتیبانی تصمیم گیری بکار می روند.

◀ سیستم های خبره می توانند در مسائلی که ذکر اهداف و مسائل در آنها مشکل است بکار روند.

◀ سیستم های خبره برای اخذ پیشنهادات متناوب جهت پردازش راه حل مسئله بسیار مؤثر هستند. این پیشنهادات ممکن است هر چه را که در دستور ورود و خروج نوشته شده

است دربرگیرد و این مسئله از قابلیت استدلال سیستم های خبره ناشی می شود.

سیستم های پشتیبانی تصمیم گیری (DSS) اغلب به انسان در تصمیم گیری در مورد تحلیل های کیفی یا کمی کمک می کنند، در حالی که سیستم های خبره سعی در جانشینی انسان متخصص دارند. تفاوت های بین DSS و سیستم های خبره در جدول 1-3 به صورت خلاصه ذکر شده است.

جدول 1-3 تفاوت های بین DSS و سیستم های خبره

مشخصات	(DSS) سیستم های پشتیبانی تصمیم گیری	سیستم های خبره
اهداف	کمک در تصمیم گیری	جایگزین توانایی های انسان
جهت یابی عمده	نصمیم گیری	انتقال تخصص
سازنده تصمیم	کاربر و سیستم AND/OR	سیستم
قلمرو مسئله	بزرگ - پیچیده	محدود
دیدگاه نیاز	باز	بسته
جهت گیری نیازمندی	نیازمندی های بشر	سیستم نیازمندی های بشر
عملیات روی داده	سمبلیک - عادی	سمبلیک
مدل های ریاضی	گسسته	-

تمرینات

1- سرویس های مدیریت در گذشته برای مسائل پیچیده از قبیل زمان بندی و برنامه ریزی به تکنیک های ریاضی پایبند بوده اند. چرا اکنون به جای آن تکنیک سیستم های خبره به کار برده میشوند؟

2- فکرمی کنید چه چیز نرم افزار کامپیوتر را هوشمند خواهد کرد. توضیح دهید.

3- درواثر منفی و مؤثر در توسعه تکنیک های هوش مصنوعی را بیان کنید.

4- چرا برنامه حل مسأله عمومی در این فصل در ابتدا مورد بحث قرار گرفت؟

خلاصه مطالب فصل اول

تحقیق در مورد هوش مصنوعی در دهه 1940 که مصادف با پدیدار شدن اولین نسل از کامپیوترها در مراکز تحقیقاتی بود، شروع شد.

یکی از عمده ترینو قابل توجه ترین برنامه های هوش مصنوعی منطق تئوری یا منطق علوم نظری نوول، شاو و سایمون در سال 1963 بود. این برنامه قادر بود 38 قضیه از 52 قضیه وایت هل راسیل (سال 1913) را در ریاضیات پایه اثبات کند.

بسیاری از سیستم های خبره که در سال 1970 میلادی ساخته شده بودند، آزمایشی بودند و اساساً به تحقیق دانشگاهی محدود می شدند.

سیستم XCON (مک در موت در سال 1982) از جمله اولین این سیستم ها بود. DEC این سیستم را در اوایل دهه 1980 میلادی ساخت که نمونه ای از یک سیستم خبره موفق است.

هم اکنون چندین پروژه هوش مصنوعی وجود دارند که جالبترین آنها Cyc است. Cyc از کلمه Encyclopaedia گرفته شده است. Cyc به ذخیره مقدار زیادی از دانش واقعی مبادرت می ورزد.

مزایای کامپیوتری

- ◀ انسان متخصص موقت است .
- ◀ انسان متخصص همیشه ثابت قدم نیست.
- ◀ انسان متخصص به راحتی قابل انتقال نیست و یک انسان بطور همزمان نمی تواند در دو مکان حضور داشته باشد ولی کامپیوتر نسبت به انسان راحتتر قابل انتقال است .
- ◀ هزینه انسان متخصص زیاد است .

مزایای انسان

- ◀ انسان ها خلاق هستند.
- ◀ انسان ها انعطاف پذیر هستند
- ◀ گرچه سیستم های خبره قابلیت یادگیری دارند اما در مقایسه با انسان قدرت یادگیریشان بسیار ضعیف می باشد.

فواید سیستم های فبره

- ◀ نگهداری دانش
 - ◀ توزیع دانش
 - ◀ آموزش
 - ◀ رقابت
 - ◀ کاهش قیمت
- مزایای فردی سیستم های خبره
- ◀ دسترسی به دانش
 - ◀ آموزش
 - ◀ سازگاری
- سیستم های پشتیبانی تصمیم گیری (DSS) یک شاخه از تحقیق در عملیات (OR) هستند که به استنتاج نیاز دارند و برای مسایل مدیریت پیچیده تکنیک های کمی را بکار برده ، به محاسبات طولانی و پیچیده نیز نیاز دارند.

تفاوت بین سیستم های فبره و سیستم های پشتیبانی تصمیم گیری

- ◀ راه حل های خبره برای مسائلی که احتیاج به تکنیک های بهینه سازی ریاضی دارند، کاربرد ندارند ولی این راه حل هادرسیستم های پشتیبانی تصمیم گیری بکار می روند.
- ◀ سیستم های خبره می توانند در مسائلی که ذکراهداف و مسائل در آنها مشکل است بکار روند.
- ◀ سیستمهای خبره برای اخذ پیشنهادات متناوب جهت پردازش راه حل مسئله بسیار مؤثر هستند.

سیستم های خبره چه هستند؟

یک سیستم خبره، برنامه ای است که سعی می کند از انسان متخصص، در استفاده در روش های استنتاج برای یک قالب معین از دانش تقلید کند. این قالب از دانش حوزه نامیده می شود اینک فرق بین داده ها، اطلاعات و دانش معلوم باشد بسیار مهم است. داده، چیزی بیشتر از یک دسته نشانه های الفبایی نیست. در زیر یک سری از داده ها را می بینیم.

a.6

b.6.0

c.-6

d.Oasis

لیست فوق یک گروه از نشانه ها می باشند که به تنهایی معنایی ندارند و فقط زمانی معنی می دهند که زمینه یا متنی به آنها اضافه شود که در این صورت اطلاعات نامیده می شوند.

آیتم های فوق در زیر، به اطلاعات تبدیل شده اند:

a.f6

b.6.0 جوایز مسابقات را یادآوری می کند.

c.-6 °c

d.Oasis یک گروه نوازنده معروف در انگلستان هستند.

هر آیتم در لیست فوق از یک سری مشخصه های الفبایی (داده) به یک سری اطلاعات معنی دار تبدیل شده اند. به این صورت که علامت f در کاراکتر a به مبلغی از پول اشاره می کند.

کاراکتر b علامت های داده، در بعضی مسابقات را نشان می دهد. کاراکتر c، به یک دمای خیلی سرد اشاره می کند. و آیتم های نهایی که یک گروه نوازنده معروف در انگلستان که Oasis نامیده می شود اشاره می کند. حال به آیتم های زیر توجه کنید. این آیتم ها دانش است که از راه تجربه و آزمایش بدست آمده اند. ملاحظه کنید که هر بخش ارتباط بین فرضیه ها و قواعد را برای عملیات روی آن ها، بیان می کند.

a. هوا بارانی است و به این دلیل من خیس خواهم شد.

b. دما 60°C - است و بنابراین سرد است .

c. موتور ماشین داغ است ، پس باید مورد استفاده قرار گرفته باشد.

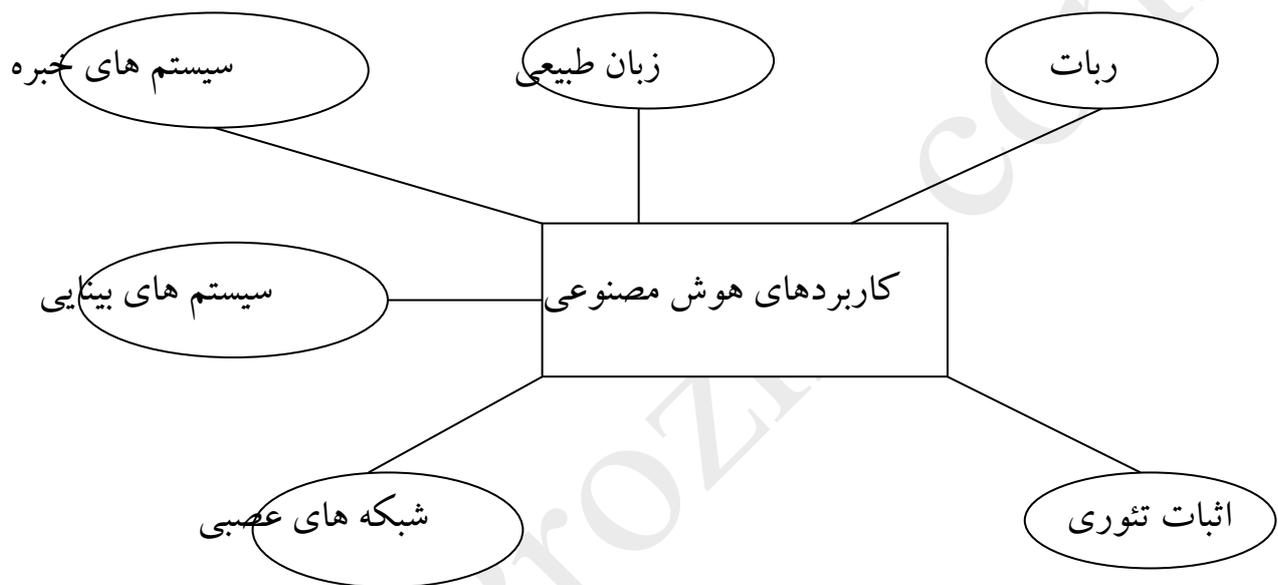
d. Oasis نوازنده هایی موفق هستند.

دانش با اطلاعات فرق دارد. اطلاعات غیر فعال بوده ، پویا نمی باشند. در حالی که مفهوم دانش ، فعال است به این معنا که با توسعه و پیشرفت چیزهایی جدید تولید می شود. برای مثال در بخش a از لیست فوق تجربه به ما می گوید که اگر در باران بیرون برویم خیس خواهیم شد و ما از این اطلاعات که "خیس خواهیم شد" نتیجه می گیریم که هوا بارانی است . این دانش بصورت قواعد نشان داده شده است . به همین صورت بخش d توضیح می دهد که Oasis نوازنده های معروفی هستند و از دانش این استنباط می شود که آنها میلیونرها هستند (تجربه زیر کانه ای وجود دارد مبنی بر اینکه گروه های نوازنده موفق معمولاً پولدار می شوند).

هوش مصنوعی

یکی از پیشگامان هوش مصنوعی ماروین مینسکی است . وی هوش مصنوعی را به این صورت تعریف می کند: "زمینه ای برای مطالعه است که سعی می کنیم سیستم هایی بسازد که اگر به

نتیجه رسید مردم به عنوان یک سیستم هوشمند به آن توجه کنند." (مینسکی در 1975). هوش مصنوعی زمینه گسترده ای است که محدوده های کاربردی مختلفی دارد. این کتاب فقط مربوط به مطالعات سیستم های خبره می باشد. کواوسین در سال 1998 میلادی بسیاری از زمینه های کاربردی هوش مصنوعی را به صورت جزئی مورد بررسی قرار دارد.



شکل 2-1: برخی کاربردهای هوش مصنوعی

هیوریستیک (کشفیات ذهنی)

سیستم های خبره به عنوان یک شاخه هوش مصنوعی مورد توجه قرار گرفته اند چون عمدتاً روش حل مسائل در آن براساس هیوریستیک (کشفیات ذهنی) است و این باروشی که الگوریتم ها برای حل برنامه ها بکار می برند متفاوت است.

الگوریتم یک روش گام به گام حل مسأله است. پردازش لیست پرداخت یک شرکت، الگوریتم را بکار می گیرد. استفاده از یک روش گام به گام روی داده و ورودی مثل ساعت کار یک کارمند، در صد اضافه کاری و غیره، خروجی مانند حقوق کارمندان را تولید می کند.

مراحل الگوریتم ، دستکاری مستقیم داده های عددی را برای تولید اطلاعات شامل می شود دستکاری مستقیم داده های عددی را برای تولید اطلاعات شامل می شود. اغلب برنامه های معمولی از روش های الگوریتمی برای حل مسأله استفاده می نمایند . از سوی دیگر هیوریستیک یک مسأله را با روش آزمون و خطا و با اتکا به تعدادی منبع برای یک هدف از پیش تعیین شده حل می کند . مثال تشخیص نقص کامپیوتر یک مورد برای آن است. همچنین مثال های فراوان دیگری نیز وجود دارد که ممکن است با آن ها مواجه شویم . به عنوان مثال، یک اتومبیل سواری برای جستجوی جای پارک در یک پارکینگ چند طبقه الگوریتمی را به کار نمی برد و هیچگونه تضمینی مبنی بر اینکه یک فضای پارک پیدا کند، وجود ندارد . راننده ممکن است تا بالاترین طبقه رانندگی کرده و هر سطح را جستجو کند، ولی باز هم هیچگونه تضمینی مبنی بر اینکه این روش جواب خواهد داد ، وجود ندارد .

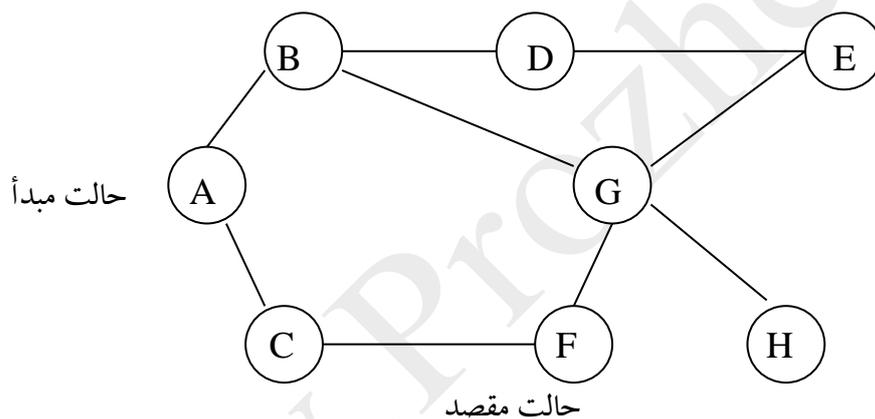
جستجوی فضای حالت

راه حل اینگونه مسائل به وسیله جستجو در بین حالات ممکن می باشد. در مورد مثال نقص کامپیوتر جستجوی کا شامل یافتن علت نقص می باشد. راه حل اینگونه مسائل به این صورت است که بدانیم چه طور از یک حالت اولیه به یک حالت نهایی یا هدف منتقل شویم. به عنوان مثالی دیگر بازی شطرنج را در نظر می گیریم که یک حالت اولیه را در بر می گیرد (ترکیب اولیه صفحه بازی) و یک حالت نهایی (مات کردن) . پیروزی در این بازی به دانش برای ترکیب صحیح جابجایی ها نیاز دارد تا به حالت نهایی برسد، که این یک جستجو در فضای حالت ممکن است . فضای جستجو به صورت ساختاری سلسله مراتبی که درخت نامیده می شود ارائه می گردد. مثال بعدی این نکته را روشن می کند.

مثال

فرض کنید که فروشنده دوره گردی بخواهد مسیری از حالت اولیه A تا حالت نهایی F از میان شبکه راههای نشان داده شده در شکل 2-2 پیدا کند. این حالات می توانند شهرهای انگلستان را نشان دهند. به عنوان مثال حالت اولیه (A) می تواند شهر شفیلد را نشان دهد و حالت نهایی (F) نیز شهر کاردیف باشد.

حالت واسط از قبیل B و C و غیره ممکن است شهرهایی باشند که در میان این شهرها قرار دارند مانند منچستر، بیرمینگهام و غیره. تمامی شهرها با حروف بزرگ A، B و C و غیره مشخص شده اند.



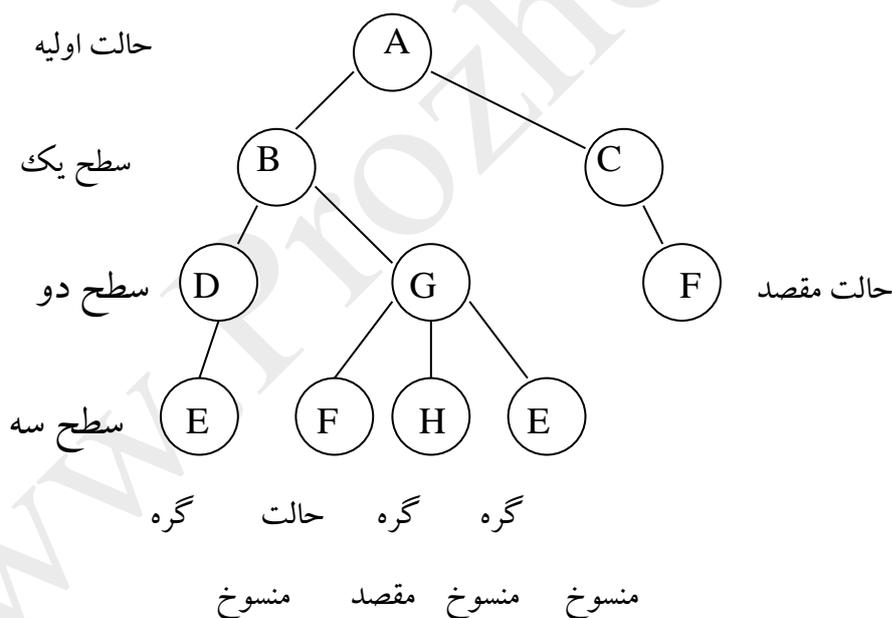
شکل 2-2: نمودار شبکه راه

از نمودار شبکه راه در شکل 2-2 یک نمودار درختی به شکل نمودار 2-3 می توان ایجاد نمود. یک درخت ساختاری، سلسله مراتبی است که شامل گره ها و شاخه هایی است که گره ها را به هم متصل می کند. از اینرو هر گره در نمودار درختی یک شهر را نشان می دهد و هر شاخه به عنوان یک راه نمایش داده می شود. مسیر راه حل از حالت اولیه در امتداد شاخه های درخت سیر می کند و در گره های علامت گذاری شده به حالت نهایی خاتمه می یابد. توجه کنید که در یک

مسأله هوش مصنوعی ممکن است بیشتر از یک راه حل داشته باشیم یعنی بیشتر از یک راه برای رسیدن به حالت اولیه به حالت نهایی وجود داشته باشد.

تکنیک های جستجو

درک این قضیه مشکل نیست که باور نماییم فضای جستجو به همان صورت که گره ها افزایش می یابند، سریعاً بزرگ می شود. این امر همیشه در مسائل عمده هوش مصنوعی مثل بازی شطرنج به مثابه مانع بزرگی بوده است. این پدیده افتخار ترکیبی نامیده می شود. افتخار ترکیبی باعث گسترش تکنیک های هیوریستیک شده است. تکنیک های جستجو به دو دسته تقسیم می شوند. جستجوی کورکورانه و جستجوی هیوریستیک.



شکل 2-3 : درخت جستجو برای شبکه راه فروشنده دوره گرد

جستجوی کورکورانه

با این تکنیک جستجو، یک طرح ترکیبی انتخاب شده و تا زمانی که یک راه حل یافت شود و یا فضای جستجو به اتمام برسد، مورد استفاده قرار می گیرد. در اینجا دو روش برای مدیریت یک

جستجوی کورکورانه وجود دارد روش اول عمق و روش اول پهنا. و در روش اول عمق جستجو از گره ریشه شروع می شود و در سطح پایین تر هر گره موجود را مورد بررسی قرار می دهد. بدین ترتیب جستجو به اتمام می رسد. وقتی یک گره منسوخ پیدا شود مکانیزم جستجو به عقب بر می گردد. بنابراین برای یک جستجو اول عمق در شکل 2-3 به طور پیاپی مسأله انتقال صورت گرفته است. شروع از گره A بوده است و سپس گره B سطح پایین بعدی به حساب آمده و هدف مسأله پیدا نشده است و همچنین گره سطح پایین تر بعدی که بررسی می شود، D می باشد که گره هدف نیست و این جستجو تا سطح پایین تر این گره نیز ادامه پیدا می کند و یک گره منسوخ می رسد (گره E، که آن نیز هدف نیست) و روش جستجو تا گره B عقبگرد می کند. دوباره فرآیند جستجو با پایین رفتن تا سطح بعدی در گره C ادامه می یابد. هیچ یافت نشده است. بنابراین گره بعدی که F است بازدید می شود. و گره هدف می باشد. اما ساختار روش اول پهنا به این صورت است که گره های درخت جستجو بوجود می آیند و سطح به سطح و تا اندازه ای عمق به عمق امتحان می شوند و براین اساس جستجو روی درخت شکل 3-2 صورت می گیرد.

گره B به دنبال گره A بررسی می شود، گره B هدف نیست، بنابراین گروه بعدی در این سطح جستجو می شود. گره بعدی گره C می باشد که این گره نیز هدف نیست و چون گره دیگری در این سطح وجود ندارد به سطح بعدی که با گره D شروع می شود می رویم و جستجو می کنیم و چون گره D هدف نیست گره بعدی موجود در این سطح که G است بررسی می شود و این نیز هدف نمی باشد. سپس گره F تولید شده که حالت انتهایی است و روش اول پهنا خاتمه می یابد.

فود آزمایی

تعداد گره هایی که توسط روش اول عمق و اول پهنا بررسی شده اند را برای مسأله شکل 2-2 بشمارید. کدام روش برای این مسأله بهتر است؟ آیا شما می توانید درخت ساختار یافته ای که کاربرد روش اول پهنا در آن از روش اول عمق بهتر باشد را پیشنهاد کنید؟

جستجوی هیوریستیک (ذهنی)

یک جستجوی کورکورانه هیچ استفاده ای از دانش (درمورد مسأله ای که مورد بررسی قرار می دهد) نمی کند. در مسائل پیچیده، جستجوها اغلب نیازمند تدابیر زیادی برای کنترل انفجار ترکیبی می باشند. جستجوهای هیوریستیک (ذهنی) محدودده خاصی از دانش را برای پیمایش فضای جستجو به کار می گیرند این تکنیک با تکنیک جستجوی کورکورانه که بدون هیچ توجهی به محدوده دانش به کار می روند تفاوت دارد. به عنوان مثال دریک بازی شطرنج، هیوریستیک می تواند درراهی حرکت کند که بیشترین تعداد مهره ها را از حریف بگیرد. اگر چه این قانون ممکن است بعنوان یک "قانون سرانگشتی" خوب باشد ولی همیشه موفق نیست. حالت نهایی در یک بازی شطرنج داشتن مهره های بیشتر از حریف نیست هرچند که داشتن مهره های بیشتر فرصت های پیروزی را بیشتر می کند. درمسأله فروشنده دوره گرد، یک روش ساده برای انتخاب مسیرها، استفاده از یک تابع ارزیابی است. تابع ارزیابی هرگره را ایجاد می کند و سپس مسیری که کمترین هزینه را دارد دنبال می کند. این هزینه ممکن است مجموعه هزینه حرکت از یک گره به گره بعدی باشد. نوعاً تابع ارزیابی گره ها هزینه را از ریشه تا گره خاصی که مورد آزمایش قرار گرفته است محاسبه می کند و با استفاده از مباحث هیوریستیک هزینه را از این گره تا گره هدف تخمین می زند. این روش به منظور تصمیم گیری در این مورد که آیا دریک مسیر

پیشروی و یا در مسیرهایی که قبلاً امتحان شده اند مجدداً سعی نمایم راهنمایی سودمندی می نماید.

کاربرد قوانین در نمایش دانش

سیستم های خبره به خاطر پردازش دانش با برنامه های رایج متفاوت هستند. این دانش در یک کامپیوتر به صورت قوانین نمایش داده می شود. قوانین روش های هیوریستیک یک انسان متخصص را در بردارند. به عنوان مثال قانونی که ممکن است توسط یک خبره برای رهن خانه مورد استفاده قرار گیرد عبارت اساز:

(IF) اگر متقاضی خانه را دوست دارد.

(AND) و خانه ارزش بازدید را دارد.

(Then) سپس متقاضی را برای استفاده از وام راهنمایی کنید.

قوانین معمولاً به صورت زیر بیان می شوند:

(IF) اگر شرط (ها)

(Then) سپس عمل (ها)

و عمل ها زمانی اجرا می شوند که شرط های قانون مورد قبول باشند. مجموعه قوانین ایجاد شده در این روش پایگاه دانش نامیده می شوند. بسیاری از سیستم های خبره از قوانین استفاده می کنند و به این دلیل سیستم های متکی بردانش نامیده می شوند. به هر حال همه سیستم های خبره تابع قوانین نیستند. مدل های دیگر نمایش دانش عبارتند از: قاب ها، شبکه های معنا و منطق که به صورت متداول مورد استفاده قرار می گیرند. روش های نمایش دانش در فصل 3 توضیح داده شده است.

استنتاج

هنر واقعی یک سیستم خبره استفاده از ظرفیتش جهت استنتاج است. این دقیقاً همان چیزی است که سیستم خبره را هوشمند می سازد. استنتاج نتیجه گرفتن از مراتبی است که از پیش بیان شده است. از اینروست که برای استنتاج باید شرایطی وجود داشته باشد و نتیجه آن نیز موجود باشد. وجود مختلفی از استنتاج وجود دارند، استنتاج قیاسی و استقرایی از مهمترین انواعی هستند که عموماً به کار می روند.

استنتاج قیاسی

قضایای (1) و (2) و (3) نشان دهنده یک مثال از استنتاج قیاسی هستند. نتیجه (3) در این مثال یک نتیجه منطقی است و می تواند قیاسی از فرضیات (1) و (2) باشد. در این مثال از اطلاعات داده شده، نتیجه استنتاج می شود. این استنتاج که یک نتیجه گیری از فرضیات است به نوعی از استدلال منجر می شود که عیناً به شکل قوانین ریاضی است. به این صورت که اگر مقدمات یا شرایط درست باشند پس می توان گفت که نتیجه نیز درست است.

استنتاج استقرایی

حال به قضایای زیر توجه کنید:

(4) همه حیوانات غذا می خورند.

(5) بنابراین همه کانگوروها غذایی خورند.

درنگاه اول ممکن است به نظر آید که نتیجه بالا (5) یعنی "همه کانگوروها غذایی خورند" به صورت قیاسی از قضیه شماره (4) بدست آمده است. که این تصور نادرست است چون داده های قضیه درباره اینکه آیا کانگورو یک حیوان است یا نه چیزی نمی گوید و انسان بر طبق دانشی که

از جهان واقعی دارد استنتاج می کند که کانگورویک حیوان است . از این رو استنتاج (5) به اندازه استنتاج (3) دقت ریاضی ندارد. برای اینکه استنتاج (5) صحیح باشد باید اینگونه فکر کنیم که “هر کانگورویک حیوان است ” تا به فرض قطعی (4) از طریق دانش حواس پنجگانه برسیم . این نوع از استنتاج که استنتاج استقرایی نامیده می شود، دقت ریاضی قیاس راندارد و با این حال در سیستم های خبره متداول است زیرا با استنتاج های بشری در جهان واقعی جور در می آید.

دانش رویه ای و اعلانی

روش حل مسائل هوش مصنوعی و سیستم های خبره قطعاً اعلانی است . یعنی همه مسائل با تعیین اینکه یک راه حل به چیزهایی نیاز دارد، حل می شوند. برای مثال به قضیه زیر توجه کنید:

(6) دیوید تنیس دوست دارد.

(7) هر چیزی که دیوید دوست دارد ژانیس نیز دوست دارد.

با استفاده از استنتاج قیاسی به این نتیجه می رسیم که ژانیس نیز تنیس را دوست دارد. قانون 7 و واقعیت 6 ما را به این استنتاج راهنمایی می کند، این یک دیالگرام اعلانی است . ملاحظه می کنید که نتیجه به وسیله استدلال بدست آمده و معلوم است که جواب با یک برنامه کامپیوتری بدست نیامده است . از جمله زبان هایی که از دیالگرام اعلانی استفاده می کنند زبان PROLOG است . کلمه پرولوگ از کلمات برنامه نویسی و منطق گرفته شده است (PROGRAMING) LOAIC که شامل یک منطق ساختاریافته است تا مسائل را بصورت اعلانی استدلال کند. این زبان با زبانهای الگوریتمی رایج که عمدتاً به چگونگی اجرای یک رویه می پردازند ، تفاوت دارد.

موتور استنتاج

حتی موقعی که قلمرو دانش رابا قوانین نمایش می دهیم بازهم یک فرد خبره باید مشخص کند که کدام قوانین رابرای حل مسئله خاصی بکارمی برد. علاوه براین باید مشخص کند که این قوانین را درچه رده ای بکارمی برد به طورمشابه یک سیستم خبره نیازخواهد داشت تا سیستم بگیرد که چه قانونی ودرچه مورد ورده ای باید برای ارزیابی انتخاب شود. برای اینکه کارصورت گیرد سیستم خبره یک موتوراستنتاج رابکارمی گیرد. موتوراستنتاج برنامه ای است که قوانین رادرپایگاه دانش تفسیرمی کند تا نتیجه را حاصل نماید.

دواستراتژی مهم برای سیستم های پایگاه قانون بکاربرده شده اند: حرکت روبه جلو وحرکت روبه عقب یک موتوراستنتاج ممکن است یکی یا هرردورا انتخاب کند. دریک موتوراستنتاج روبه عقب هدف تعیین می شود(شناسایی می شود) وسی می شود تا هدف با تأیید درستی همه شرایط اثبات شود به عنوان مثال به قانون زیرازسیستم خبره MYCIN توجه کنید:

(IF) اگرآلودگی وچرک بدن صددرصد منفی است .

(AND) وهیکل وریخت بدن قوی است .

(AND) وسیستم تنفس ناجوروبدون اکسیژن است .

(THEM) پس مدرکی وجوددارد که دلالت می کندبدن به اندازه 0/8 دارای باکتری است

اثبات این قانون ازحرکت روبه عقب بهره می گیرد موتوراستنتاج سعی می کند تا با تحقق

بخشیدن هرقضیه نتیجه هر قانون رابدست آورد این قضایا ممکن است خودشان نتایج قوانین

دیگرباشند . درحالت خاص MYCIN باید چنین عملی رابرای پردازش مشابه درمورد چنین

قضایایی انجام دهد ویا ارزش این قضایا به وسیله کاربر ازطریق مشاهدات تخصصی تأمین

شوددراین روش یک زنجیره از مراحل استنتاج دریافتن ارزش هدف ما را راهنمایی می کند.

درمقابل یک موتوراستنتاج روبه جلو از طرف دیگر شروع می کند و دانش را در پایگاه دانش می یابد و قوانینی را که از دانش می تواند برداشت شود پیدانموده و سپس نتایج آن قوانین را به پایگاه دانش می افزاید بعد از آن کل پایگاه دانش را می آزماید و تحلیلش را گزارش می نماید و می تواند تا زمانی که اطلاعات جدید اضافه می شوند پیشرفت کند. هر دو گونه استنتاج روبه عقب و روبه جلو در گیرنده یک زنجیره از مراحل است که می تواند به وسیله سیستم خبره ردیابی شود. این مسئله سیستم های خبره را قادر می سازد تا استدلالشان را توضیح دهند.

امکانات تفسیر

یکی از خصوصیات کلیدی سیستم های خبره توانایی توضیح استدلال پردازش می باشد. به هر حال برای بسیاری از سیستم های خبره توانایی توضیح، چیزی به غیر از ردیابی نتایجی که در طول مراحل یک استنتاج توسط سیستم حاصل می شود، نیست. به این معنی که امکان تفسیر پردازش ها کاربر را متوجه می سازد که سیستم چگونه به این نتایج رسیده است، یا اینکه سیستم چرا برای پاسخگویی به یک سؤال خاص در تکاپو می باشد. این امکانات تفسیر کاربر را در فهمیدن رفتار سیستم راهنمایی می کند و این مسئله مهم است، چرا که استنتاج بایک متخصص خبره نیز اغلب به مقداری توضیح نیاز دارد. بسیاری از مردم اکثر اوقات جوابهای یک خبره را بدون توضیح نمی پذیرند. به عنوان مثال از یک پزشک متخصص تشخیص علت بیماری فرد را انتظار داریم. وی نتایج را توضیح می دهد چرا که طبیعتاً تصمیم گیری نامطمئن او ممکن است جزئیات بیشتری را بخواهد. گرچه بیمار از هر گونه خطر احتمالی و درمان های پی در پی و غیره آگاه است. همین مسئله در تصمیم گیری مدیریت نیز اتفاق می افتد. توصیف صریح خطرات یا توضیح مراحل یک تصمیم

گیری یا یک پیشنهاد، موضوعاتی هستند که به عنوان ابزاری برای حمایت از مدیریت می باشند. در فصل های بعد موضوع تفسیر را با جزئیات بیشتری بررسی می کنیم .

ابزارهای ایجاد سیستم خبره

زبان های تک منظوره ای مانند PROLOG , Lisp جهت توسعه برنامه های کاربردی هوش مصنوعی استفاده می شوند. به هر حال ساخت سیستم های خبره با استفاده از این زبان ها احتیاج به دانش تفصیلی در مورد زبان و کارسخت دارد. به این دلیل برنامه هایی که پوسته سیستم خبره نامیده می شوند غالباً برای ساخت سیستم های خبره مورد استفاده قرار می گیرند. پوسته ها (shells) نقطه شروعی برای ساخت سیستم های خبره هستند. آن ها سیستم های خبره ای می باشند که قوانینی در آن ها وجود ندارد، به این معنی که توسعه دهندگان فقط بر ساختار پایگاه دانش تمرکز داشته و نگرانی بابت قسمت های دیگر مثل موتور استنتاج ندارند. اگرچه آنها می توانند خیلی مفید باشند، اما پوسته ها برای تبدیل یا تغییر مسیر کار با توجه به مکانیزم استنتاج و نمایش قابل انعطاف نیستند. AM برای 95 WINDOWS , OBJECT 5 LEVEL دو نمونه از متداول ترین پوسته هایی هستند که در انگلستان استفاده می شوند. دیگر ابزارهای ساخت سیستم های خبره، محیط های هوش مصنوعی یا جعبه ابزار (TOOLKIT) نامیده می شوند. آنها ابزارهایی هستند که توسط توسعه دهندگان با تجربه مورد استفاده قرار می گیرند. نموداری که در شکل 2-4 نشان داده شده است توزیع نرم افزار سیستم خبره به کاربرده شده در انگلستان بر اساس نوع ساخت را نشان می دهد (ادوارد 1990).

همانطور که نمودار نشان می دهد پوسته ها در انگلستان زیاد استفاده می شوند. برای اینکه بسیاری از سیستم های خبره ای که در انگلستان توسعه داده شده اند ، سیستم های کوچکی هستند

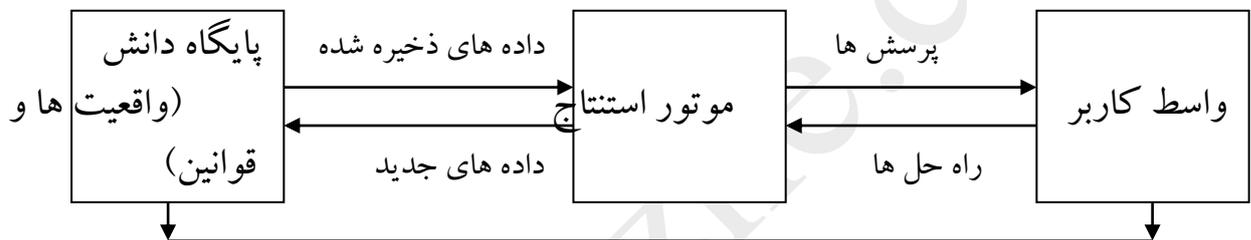
و بطور کلی زمان لازم برای توسعه دهندگان نرم افزار کمتر از 3 ماه می باشد. اما در آمریکا ، سیستم های خبره به برنامه های کاربردی مجتمع که بزرگتری باشند تمایل دارند و زمان لازم برای توسعه آنها 12 ماه یا بیشتر است و به همین دلیل در آنجا بیشتر از جعبه های ابزار (TOOLKIT) جهت تولید سیستم های خبره استفاده می شود.

شکل 2-4 : نمودار استفاده از ابزارهای مختلف برای تولید سیستم های خبره در انگلستان (ادوارد 1990)

واسطه کاربر

واسطه کاربریک سیستم خبره، طبیعتاً باید از قدرت تبدلی بالایی برخوردار باشد تا ساختار تبادل اطلاعات به شکل گفتگوی یک متقاضی و انسان خبره صورت گیرد. واسطه کاربر سیستم خبره نه تنها کاربر را قادر می سازد تا به سؤالات پاسخ دهد بلکه کاربر را مجاز می سازد عملیات اجرایی سیستم را با پرسش در مورد توضیحات داده شده قطع نماید. بنابراین وقتی که سیستم در حال پرسش یک سؤال از کاربر است، کاربر می تواند با پرسیدن یک سؤال از سیستم در مورد اینکه چرا سیستم این سؤال را پرسیده است ، کار را متوقف نماید.

برای مثال یک کاربر سیستم های خبره پزشکی ممکن است بخواهد بداند که چرا سیستم در مورد سن بیمار سؤال کرده است. این نوع از استفهام، تفسیر چرا نامیده می شود. نوع دیگری توضیح استفهامی، پرسش چطور است. از این کلمه معمولاً برای پرسش در مورد اینکه سیستم چگونه به یک تصمیم مشخص رسیده است استفاده می شود. برای مثال اگر به یک کاربر سیستم خبره پزشکی گفته شود که بیمار مننژیت دارد، کاربر ممکن است بخواهد بداند که سیستم چگونه به این نتیجه رسیده است.



شکل 2-5: طرح پایه برای یک سیستم خبره

معماری سیستم های خبره

روابط بین مفاهیم مختلف در سیستم های خبره در شکل 2-5 نمایش داده شده است این نمودار معماری پایه ای سیستم خبره را نشان می دهد.

قابلیت یادگیری

خصیصه دیگری که سیستم های خبره را از سیستم های رایج متمایز می سازد قابلیت آنها برای یادگیری است. برخی از آنها حتی بدون کمک کاربر از طریق پردازشی که قانون استقراء¹ نامیده می شود یادگیری می نمایند. این قانون داده های آماری را در یک مسأله تجزیه می کند با این هدف که قوانین جدید را بوجود می آورد. چنین تکنیک هایی در مورد مسائلی با دامنه های

تعریف شده ، تأثیر بسزایی خواهد داشت. بسیاری از ابزارهای ساخت سیستم خبره ، موتور استنتاجی را ایجاد می کنند که خود این موتور استنتاج قادر به ایجاد قوانین جدید از نمونه های داده شده می باشد این موضوع با جزئیات بیشتر در فصل 6 توضیح داده شده است.

مقایسه بین سیستم های فبره با سیستم های قراردادی رایج.²

ویژگی هایی که سیستم های خبره را از سیستم های قرار دادی متمایز می سازد در جدول 1-2 خلاصه شده است.

-
- 1.Rule Induction
 - 2.Conventional programs

جدول 1-2 : مقایسه سیستم های فبره با سیستم های قراردادی

ویژگی ها	سیستم فبره	سیستم قراردادی
نمونه های اساسی	هیوریستیک، معمولاً با استفاده از جستجو ارائه می گردد حرکت به سوی راه حل است (یعنی اینکه از طریق برنامه ریز مشخص نمی شود) و اگر راه حلی پیدا شود حتماً راه حل بهینه نیست معمولاً حل مسأله اعلانی است.	الگوریتمیک، راه حل بوسیله برنامه ریزی تعیین می شود. پاسخ های صحیح داده می شود. معمولاً حل مسأله رویه ای است.
شیوه عمل	دلایل دارای سمبل هستند مثل جمع بندی	کاملاً داده ها را دستکاری می

کند برای مثال مرتب کردن، محاسبه و ذخیره داده های پردازش شده برای سیستم پرداخت حقوق یک شرکت .	نتایج تشخیص بیماری یک مریض. موتور استنتاج برای تصمیم گیری در مورد قضایایی که ارزش یابی شده اند به کار می رود.	
داده، معمولاً به صورت آرایه یا رکوردها در زبانی مانند C یا Cobol نمایش داده می شوند. داده غیر فعال است و داده های بیشتری را ایجاد نمی کند.	دانش، ممکن است به صورت قانون نشان داده شود. دانش به صورت فعال عمل می کند به این معنی که سیستم خبره می تواند به وسیله دانش و از طریق داده دریافتی، دانش جدیدی را استنتاج کند.	واحد پردازش
داده یا اطلاعات و کنترل همیشه باهم مرتبطند.	موتور استنتاج همیشه از حوزه دانش جدا است.	مکانیزم کنترل
الگوریتم + داده	استنتاج + دانش	اجزاء اصلی
متغیر	دارای تبادل در سطح بالا و معمولاً به شکل پرسش و پاسخ	واسطه کاربر
خیر	بله - پیگیری واضح مراحل نهفته زنجیره استنتاج که به نوعی کاربر را مجاز می سازد تا بداند سیستم چگونه به نتایج خود رسیده است و یا اینکه سیستم چگونه به دنبال پاسخ یک سؤال خاص می گردد.	قابلیت تفسیر
خیر	آری - ولی محدود	قابلیت یادگیری

تمرینات

1- توضیح دهید معنی هیوریستیک چیست؟ چند مثال از هیوریستیک از دو حوزه کاربردی

متفاوت بیاورید.

2- جمله "سیستم های خبره ممکن است اشتباه کنند" درست است یا غلط؟

3- چرا سیستم های خبره یک شاخه از هوش مصنوعی تلقی می گردند؟

4- سیستم های خبره قادر به تفسیر روند استنتاج خود هستند. تفاوت میان یک تفسیر "چرا" و

"چگونه" را شرح دهید و از مثالی برای تشریح پاسخ خود استفاده کنید.

5- پنج حوزه کاربردی را که فکر می کنید سیستم های خبره می توانند در آن مفید باشند شرح

دهید.

6- در مورد اینکه قضایای زیر استنتاجی هستند یا خیر بحث کنید.

الف) ریانون استنباط کرد که گربه گرسنه است.

ب) اگر باران بیارد، من خیس می شوم.

ج) هیچ کس برنده نمی شود، پس کسی نباید بازی کند.

د) نور صحنه کم شد. سپس پرده افتاد.

ه) من فکر می کنم، بنابراین من وجود دارم.

7- استنتاج های زیر را امتحان کنید و صحت آنها را با منطق توضیح دهید.

الف) کاردیف درولز است بنابراین کاردیف در انگلستان است.

ب) توییتی پرنده است، هیچ پرنده ای نمی تواند آواز بخواند، بنابراین توییتی نمی تواند آواز بخواند.

خلاصه مطالب فصل دوم

یک سیستم خبره، برنامه ای است که سعی می کند از انسان متخصص، در استفاده در روش های

استنتاج برای یک قالب معین از دانش تقلید کند. این قالب از دانش حوزه نامیده می شود اینکه فرق

بین داده ها، اطلاعات و دانش معلوم باشد بسیار مهم است.

دانش با اطلاعات فرق دارد. اطلاعات غیرفعال بوده ، پویا نمی باشند. درحالی که مفهوم دانش ، فعال است به این معنا که با توسعه و پیشرفت چیزهایی جدید تولید می شود.

هوش مصنوعی

یکی از پیشگامان هوش مصنوعی ماروین مینسکی است . وی هوش مصنوعی را به این صورت تعریف می کند: "زمینه ای برای مطالعه است که سعی می کنیم سیستم هایی بسازد که اگر به نتیجه رسید مردم به عنوان یک سیستم هوشمند به آن توجه کنند".

سیستم های خبره به عنوان یک شاخه هوش مصنوعی مورد توجه قرار گرفته اند چون عمدتاً روش حل مسائل در آن براساس هیوریستیک (کشفیات ذهنی) است و این باروشی که الگوریتم ها برای حل برنامه ها بکار می برند متفاوت است .

سیستم های خبره به خاطر پردازش دانش با برنامه های رایج متفاوت هستند . این دانش در یک کامپیوتر به صورت قوانین نمایش داده می شود. قوانین روش های هیوریستیک یک انسان متخصص را در بردارند .

ابزارهای ایجاد سیستم خبره

زبان های تک منظوره ای مانند PROLOG , Lisp جهت توسعه برنامه های کاربردی هوش مصنوعی استفاده می شوند. به هر حال ساخت سیستم های خبره با استفاده از این زبان ها احتیاج به دانش تفصیلی در مورد زبان و کارسخت دارد.

واسطه کاربریک سیستم خبره، طبیعتاً باید از قدرت تبدلی بالایی برخوردار باشد تا ساختار تبادل اطلاعات به شکل گفتگوی یک متقاضی و انسان خبره صورت گیرد. واسط کاربر سیستم خبره نه

تنها کاربر را قادر می سازد تا به سؤالات پاسخ دهد بلکه کاربر را مجاز می سازد عملیات اجرایی سیستم را با پرسش در مورد توضیحات داده شده قطع نماید.

خصیصه دیگری که سیستم های خبره را از سیستم های رایج متمایز می سازد قابلیت آنها برای یادگیری است. برخی از آنها حتی بدون کمک کاربر از طریق پردازشی که قانون استقراء¹ نامیده می شود یادگیری می نمایند.

موضوعات

آنچه در این فصل خواهید آموخت :

- ◀ روشی مطلوب برای نمایش دانش .
- ◀ چگونگی نمایش دانش با استفاده از قوانین.
- ◀ چگونگی نمایش دانش به صورت موضوعات ساختاریافته .
- ◀ چگونگی نمایش دانش با استفاده از منطق.

◀ مقایسه بین روش های مهم نمایش دانش.

◀ معرفی شماهای مختلف نمایش دانش برای ساخت سیستم های خبره .

نمایش دانش

پیشگامان هوش مصنوعی نشان داده اند که رفتار هوشمندانه آنقدر که به دانش وابسته است به روش های استدلال وابسته نمی باشد.

برطبق آنکه برچ من (1988) گفته است:

“یک هدف شناخته شده در هوش مصنوعی ایجاد برنامه های نرم افزاری است که بتوانند با توانایی های انسان در محدوده های هوش مصنوعی مانند طراحی، نقشه کشی، درک زبان طبیعی، تشخیص ها و نظارت رقابت کنند.” به طور کلی بیشتر این کارها براساس این باور است که سیستم های هوشمند می توانند به صورت واضح و صریح از روی پایگاه های دانش ساخته شوند، که این مسئله موضوع مکانیزم های استدلال را پیش می کشید این فرض اساسی در هوش مصنوعی که استدلال و نمایش دانش سمبلیک می باشد، مهمترین بحث در این زمینه است.

دانش می تواند به طرق مختلفی نمایش داده شود. روش های نمایش دانش دربخش های بعدی توضیح داده شده است. توجه به برخی از خصوصیات روش های مختلف نمایش دانش بسیار مهم می باشد برخی از این خصوصیات عبارتند از :

◀ کامل بودن، به عبارت دیگر، روش نمایش دانش باید استنتاج از تمامی موارد دانش را

پشتیبانی کند .

◀ اختصار، اطمینان از اینکه نمایش ها مختصر هستند . یعنی روش نمایش باید ضمن اینکه استنتاج کافی را ایجاد می کند، دانش را به صورت خلاصه ذخیره نماید و همچنین راحتی قابل اصلاح و تغییر باشد .

◀ محاسبات دقیق و کافی (بازدهی خوب)، باید بتوان به سرعت و بدون انجام محاسبات بسیار زیاد از دانش استفاده نمود .

◀ شفافیت (وضوح)، یعنی رفتار دانش باید راحتی قابل درک باشد و اینکه دانش چطور به نتایج می رسد راحتی معلوم باشد.

◀ ساخت موارد مهم و پیشگیری از جزئیات اضافی، باید جزئیات در جایی در دسترس باشند تا امکان استفاده از آنها وجود داشته باشد.

یک روش نمایش دانش که دربرگیرنده همه مشخصات ذکر شده فوق باشد وجود ندارد. ثابت شده است که هر کدام از روشهای نمایش دانش که در این فصل بحث خواهند شد برای حوزه های کاربردی خاصی مناسب باشند. سیستم های تولید قوانین که به اختصار در فصل دوم راجع شان توضیح داده شد در این قسمت بصورت جزئی تر بررسی می شوند.

قوانین تولید

گروه بزرگی از سیستم های خبره از قوانین که گاهی اوقات "تولید"¹ نامیده می شوند، برای نمایش دانش استفاده می کنند. شهرت قوانین به سبب آسانی و راحتی آنها در کاربرد می باشد. قوانین معمولاً به شکل زیرنمایش داده می شوند.

عمل (ها) Then شرط (ها) if

قسمت if درست چپ یک قانون حل مسئله، شرط ها را که گاهی اوقات پیش فرض ها نامیده می شوند توصیف می کنند. پیش فرض ها باید درست باشند تا قانون بتواند عمل نماید.

قسمت Then درست راست ، مجموعه عملیات را، که نتایج نامیده می شوند توصیف می کند. اگر قانون قابلیت اجرا داشته باشد نتایج دنبال می شوند.

مثال

سرعت ماشین بسیار زیاد باشد if

پای خود را روی ترمز فشار دهید Then

مثال بالا نشان می دهد اگر شرط خاص "سرعت ماشین بسیار زیاد باشد" محقق شود آنگاه قسمت عمل انجام می شود. قوانین ممکن است شامل چندین شرط و یا چندین عمل باشند که می توان با بکاربردن AND که عطف (ترکیب عطفی) نامیده می شود و یا OR که انفصال (ترکیب فصلی) نامیده می شود با هم ترکیب شوند.

مثال

اتاق سرد است if

اتاق تاریک است and

1. Production

بهترین گیاه خانگی پیچک می باشد. then

مزایای قوانین

عوامل زیر فاکتورهایی هستند که سبب استفاده گسترده از قوانین در سیستم های خبره می شوند این عوامل عبارتند از:

◀ سادگی : شکل قوانین از مدل فیزیکی خوبی برای نمایش دانش برخوردار می باشد چرا که آنها بسیار نزدیک به استدلال انسان هستند. این ویژگی سیستم های مبتنی بر قوانین را در مقایسه با دیگر روش های نمایش دانش ، روشی ساده تر و راحت تر جلوه می دهد.

◀ واحدمندی¹: به این معنی که بلاکهای قوانین می توانند مستقل از هم نوشته شوند و سپس به پایگاه قوانین اضافه شوند و درستی آنها کنترل شود. بنابراین سیستم های خبره مبتنی بر قوانین به آسانی می توانند به قطعات قابل مدیریت شکسته شوند تا راحت تر ایجاد گردند. این خاصیت است که پایگاه های دانش را قادر خواهد ساخت مرحله به مرحله ساخته شده و قدم به قدم توسعه یابند.

◀ تشریح (توصیف) : سیستم های مبتنی بر قوانین امکانات تشریح (توصیف) ساده و شفاف را ایجاد می کنند.

◀ عدم قطعیت : یک سری تکنیک هایی ایجاد شده اند که دانش را مجاز می سازند تا با استفاده از قوانین ، عدم قطعیت (دانش غیر قطعی) را نیز نشان دهند.

قوانین همچنین به منظور نمایش انواع مختلف دانش استفاده می شوند. برای مثال دانش هیوریستیک، دانش محدوده (دامنه) و دانش رویه ای همه می توانند با استفاده از قوانین نشان داده

1. Modularity

شوند. برای هر یک آنها در زیر مثالی ارائه شده است.

قوانین هیوریستیک

هرقانون دریک پایگاه قانون می تواند برای نمایش دانش حل مسئله استفاده شود. یک هیوریستیک به صورت یک قانون شاخص تفسیر می شود، به عنوان مثال یک قانون شاخص برای تشخیص نقص فنی یک اتومبیل بصورت زیر است :

چراغ ها خاموش هستند if

باتری تمام شده است then

قوانین ممدوده (دامنه)

قوانین می توانند برای نمایش روابط بین اشیاء و موضوعات در دامنه استفاده شوند. به عنوان مثال:

اگر حیوانی یک ببر باشد if

پس آن حیوان گربه است then

این قانون بریک ارتباط بین گونه هایی از یک حیوان دلالت دارد و آن عبارت از این است که بریک عضواز گروه گربه سانان می باشد. به این معنی که قوانین به ارتباط بین کلاس های اشیاء دلالت دارند.

دانش رویه ای

قوانین همچنین می توانند بر ترتیب کارها دلالت کنند برای مثال:

اگر اتومبیل بسیار سریع حرکت کند if

پای خودتان را روی پدال ترمز بگذارید then

این قوانین یک ساختار را برای دنبال کردن یک عمل یا وضعیت جاری توصیف می کنند که چنین ساختاری "دانش رویه ای" نام دارد.

معایب سیستم های تولید قانون

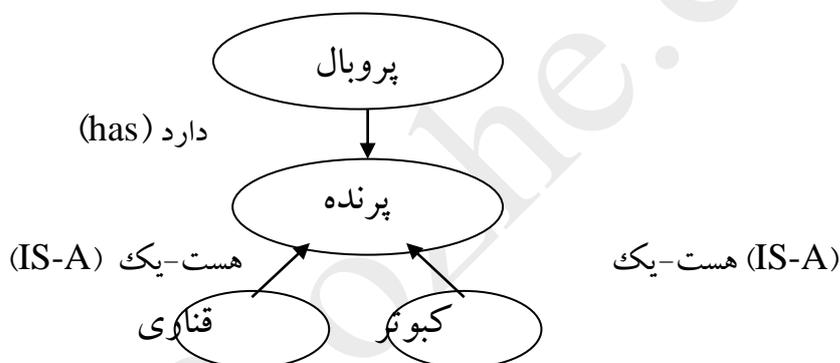
اگرچه قوانین تولید شده خاص می توانند شفافیت داشته باشند اما در سیستم هایی که شامل تعداد زیادی از قوانین هستند (بیشتر از 200)، ممکن است ترکیب نتیجه عملیات غیر شفاف باشد. به این دلیل که بعضی قوانین در یک پایگاه دانش ممکن است قوانین دامنه باشد و دیگر قوانین ممکن است قوانین کنترلی و یا دامنه ای باشند. راهی برای جداسازی و تفاوت بین گرامرها if.....then وجود ندارد و بنابراین همه نتایج ممکن است دنبال نشود. این مورد مخصوصاً ممکن است اشکال زدائی را با مشکل مواجه کند.

مشکل دیگر سیستم های مبتنی بر قوانین این است که در برخی از سیستم های خبره، دانش نمی تواند بطور کامل توسط قوانین به نمایش درآید. برای مثال در سیستم خبره پزشکی، تاریخچه بیمار بسیار مهم است. هر چند سیستم های مبتنی بر قوانین با تکیه بر پایگاه های داده وبه وسیله آنها داده هارا ذخیره می کنند. اما گاهی اوقات برقراری ارتباط موفق بین سیستم خبره مبتنی بر قوانین و پایگاه داده ها، کاری واقعاً مشکل است. در نهایت اینکه سیستم های مبتنی بر قوانین یک ناتوانی ذاتی در یادگیری دارند و نمی توانند خودشان را تغییر دهند، در حالی که این مشخصه برای هر سیستم نمایش دانش، ضروری می باشد.

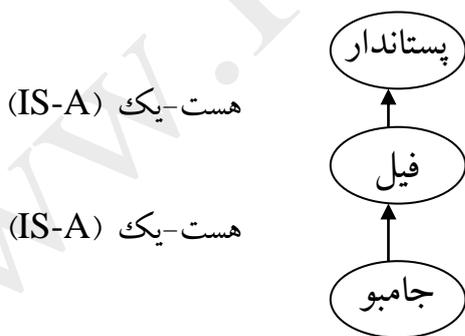
شبکه های معنایی

شبکه های معنایی یک روش گرافیکی برای نمایش دانش می باشند. این مفهوم ابتدا توسط رزکویلین در سال 1968 بیان شد. دانش می تواند بصورت اشیاء، مفاهیم، نتایج و روابط خاص بین آنها به نمایش درآید. در اصل شبکه معنا شامل یک مجموعه از گره ها و ارتباطات می باشد. گره ها اشیاء یا مفاهیم را نمایش می دهند و ارتباطات روابط بین آنها را مشخص می نمایند. یک مثال ساده از آن در شکل 3-1 نشان داده شده است.

این مثال نشان می دهد که کبوتر یک پرنده است و قناری نیز یک پرنده می باشد. یک شبکه سلسله مراتبی از ارتباطات اشیاء می تواند با استفاده از این نوع شبکه ایجاد شود. از آنجاییکه شیء پرنده مشخصاتی دارد آنگاه اشیاء فرزند مثل قناری و کبوتر که وارث نامیده می شوند نیز دارای همان خصوصیات و مشخصات هستند. به وسیله وراثت ، یک شیء می تواند بصورت اتوماتیک وار همه خصوصیات کلاس اصلی را داشته باشد. توارث مکانیزمی است برای اطمینان از اینکه یک موجودیت با گره ها و ارتباطات اضافی نشان داده نشود. مثالی دیگر از وراثت ، در شبکه های معنایی ، در شکل 3-2 نشان داده است .



شکل 3-1: یک شبکه معنای ساده



شکل 3-2: مثالی از کاربرد توارث در شبکه های معنایی

همانطوری که در شکل 3-2 می بینید جامبو یک پستاندار است . برای اینکه جامبو وارث خصوصیات فیل، و فیل نیز وارث خصوصیات پستاندار است .

مزایای توارث

در اینجا باید چندین نمونه آزمایای توارث بیان شده است:

◀ حافظه کمتری را مصرف می کند چون لازم نیست بسیاری از روابط به وضوح بیان شوند .

◀ توارث قادر است اشیاء شبیه به هم را گروه بندی نماید.

◀ هنگامی که یک گره بتواند از گره های قبلی صفات و ویژگی ها را به ارث برد، شبکه ها آسانتر به روزدرمی آیند.

شبکه های معنایی در عمل زیاد مفید نیستند چون آنها صرفاً قادر به نمایش دانش اعلانی بوده و نمی توانند دانش رویه ای را نمایش دهند. همچنین استاندارد دی برای شبکه های معنایی وجود ندارد. به این معنی که فرم آنها از یک سیستم به سیستم دیگر تغییر می کند.

قاب ها یک نمایش مبتنی بر اشیاء (شیء گرایی) می باشند که امروزه تقریباً جایگزین شبکه های معنایی در سیستم های کاربردی شده اند.

قاب ها

مینیکسی اولین شخصی بود که در سال 1975 مفهوم قاب را به این صورت تعریف کرد که: " یک قاب یک تکه ساختاریافته از اطلاعات، پیرامون خصوصیات ، مشخصات و یا رفتارهای یک شیء یک عمل ، یک موجودیت و یا یک رویداد می باشد".

یک قاب در واقع الگویی است که برای نگهداری اطلاعات یک موجودیت بکار می رود. هر قاب دارای قسمتهایی بنام اسلات (Slat) (شیار) می باشد که اسلاتها شامل صفاتی از آن موجودیت می باشند. به عنوان مثال قاب برای موجودیتی مانند حیوان دارای اسلایت هایی است که در شکل

3-3 نشان داده شده است . مقادیر اسلات در یک قاب می تواند بر سه نوع ذیل باشند:

FRAME animal:

NAME	FILLER VALUE
IS-A	sentient-being
NUMBER-OF-LEGS	default=4
IS-FURRY	default=true

شکل 3-3: الگوی قاب میوان.

1- اسلات ها ممکن است دارای مقادیر متفاوت باشند مثلاً در قاب animal، اسلات -NUMBER

of-legs دارای مقدار پیش فرض 4 می باشد و این مسأله از آنجا ناشی می گردد که اکثر حیوانات

چهارپا هستند و یا مقدار مربوط به اسلات IF-FURRY از قاب animal

2- اسلات ها در یک قاب ممکن است با هم مرتبط باشند. مانند مقدار مربوط به اسلات IS-A

در قاب animal، مقدار اسلات IS-A برای ارتباط اشیا بصورت سلسله مراتبی است. این

اسلات برای ارتباط بین قاب حیوان و قابی است که قاب موجودات نامیده می شود. در این

روش سلسله مراتبی از حیوانات می توانند بصورت قاب نشان داده شوند. شکل 3-4 را ببینید.

3- اسلات ها در یک قاب می توانند رویه ای باشند. برای مثال قاب سگ رادر شکل 3-5

در نظر بگیرید.

یک اسلات در این قاب برای مقدار غذای روزانه در نظر گرفته شده است که -DAILY-FOOD

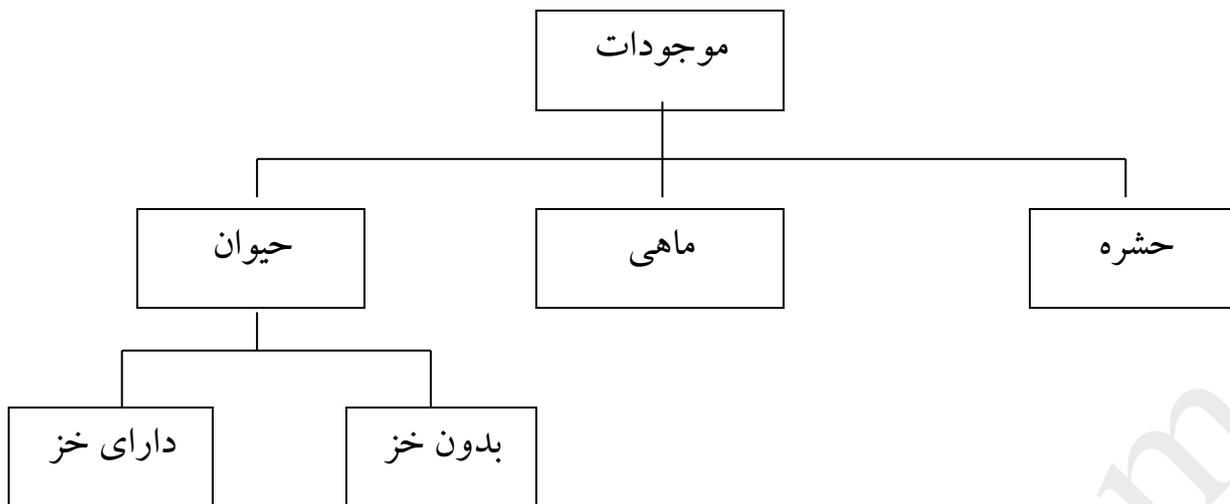
INTAKE نامیده می شود. مقادیر ممکن برای این اسلات زیاد، متوسط و کم می باشد که با توجه

به بزرگی و وزن سگ در نظر گرفته می شود. سگ های بزرگ عموماً به غذای بیشتری از سگ های

کوچک نیاز خواهند داشت. بنابراین مقدار این اسلات با محاسبه وزن و بزرگی سگ تعیین

خواهد شد. این چنین اسلاتی، رویه ای نامیده می شود. چرا که مقادیر مربوط به آن از طریق برخی

عملیات رویه ای بدست می آید.



شکل 3-4 : بخشی از سلسله مراتب قاب

FRAME dog:

SLOT	IFLLER VALUE
IS-A	animal
NAME	-
WEIGHT	-
HEIGHT	-
CAN-BARK	default=true
DAILY-FOOD-INTAKE	calculate from height and weight slots When added

شکل 3-5: قاب مربوط به سگ.

توجه کنید که اولین اسلات در قاب مربوط به سگ می گوید که سگ حیوان است. به این معنی که موجودیت سگ تمامی مقادیر مربوط به حیوان را به ارث می برد. و این یک مثال دیگر از توارث است. همانطوری که در قاب حیوان دیده می شود. سگ می تواند ویژگی چهارپا بودن را به ارث ببرد. برای اینکه مقدار NUMBE-OF-LEGS، 4 می باشد (در قاب حیوان). به صورت مشابه مقدار اسلات IS-FURRY (خزدار) مقدار True را دارد به این معنی که سگ به عنوان یک حیوان خزدار در نظر گرفته می شود.

نمونه های از اشیاء قاب

حال یک سگ خاص به نام Rover را در نظر بگیرید. قاب مربوط به سگ که در شکل 3-5 نشان داده شده است می تواند برای ذخیره مقادیر اسلات های Rover استفاده شود (شکل 3-6). از آنجایی که Rover یک سگ است پس او تمامی مقادیر اسلات های قاب سگ را به ارث می برد. به عنوان مثال برای Rover در نظر گرفته شده است که می تواند پارس اجرا کند چرا که در قاب مربوط به سگ اسلات CAN-BARK دارای مقدار True می باشد. از این رو هر زیر اسلات این مقدار را به ارث خواهد برد مگر اینکه با مقدار دیگری نوشته شده باشد. به طور مشابه Rover می تواند چهارپا بودن را از قاب حیوان به ارث برد.

زمانی که مقادیر وزن و اندازه به این قاب اضافه شده اند مقدار اسلات DAILY-FOOD-INTAKE به میزان "زیاد" در نظر گرفته شده است.

روابط سلسله مراتبی بین سگی که Rover نامیده می شود و قاب مربوط به سگ و قاب حیوان و قاب مربوط به موجودات در شکل 3-7 نشان داده شده است.

منطق¹

منطق بصورت گسترده ای در برنامه های هوش مصنوعی استفاده می شود. مهمترین بحث در منطق درستی یا نادرستی آرگومان ها می باشند.

1-Logic

نوعاً آرگومان ها شامل گزاره ها¹ و نتایج² می باشند و این پایه و اساس منطق گزاره ای می باشد که در بخش بعدی بررسی می شود.

Instance of frame Rover

SLOT

IFLLER VALUE

IS-A

Dog

NAME	Rover
WELGHT	2.20
HEIGHT	1.05
CAN-BARK	true=(inherited from default dog frame)
DAILY-FOOD-INTAKE	high:calculate from knowing the height and weight values

شکل 6-3: قاب مربوط به سگی بنام Rover

منطق گزاره ای

به منظوره فرموله کردن زبان منطق، لازم است توضیح دهیم که منظور از یک گزاره چیست؟ یک گزاره جمله ای است که می تواند ارزش درست یا غلط داشته باشد. برای مثال جملات زیر گزاره هستند:

(1) "کیت یک مردشاد است." (2) "همه گربه ها حیوانات خانگی خوبی هستند."

هر دو جمله (1) و (2) گزاره هستند و می توانند درست یا غلط باشند. جملات زیر گزاره نیستند.

(3) "حیوانات خانگیم." (4) "اوه - عزیزم."

عبارات در منطق گزاره ای معمولاً بصورت سمبلیک هستند. برای مثال استنتاج زیر را ببینید.

"اگر کیت مرد شادی است، آنگاه کیت معلم است."

1-Propositions
2-Conclusion

این جمله می تواند بصورت نمادین زیر بیان شود:

A. کیت مرد شادی است. B. کیت معلم است.

که می تواند بصورت گزاره منطقی زیر بیان شود:

if A then B

که این گزاره با استفاده از عبارات منطقی بصورت $A \rightarrow B$ نمایش داده می شود (به این معنی که گزاره A گزاره B رانتيجه می دهد).

این مثالی از یک قانون استنتاج است که قیاس استثنایی¹ نامیده می شود. این قانون می گوید، اگر گزاره A و قانون استنتاج $A \rightarrow B$ درست باشند پس B نیز گزاره درست خواهد بود.

گزاره ها همچنین می توانند به وسیله کلمات ربط منطقی به هم متصل شوند. به عنوان مثال: “اگر من یک موسیقی باخ گوش دهم و اتاق هم گرم باشد، احساس خواب خواهم نمود.” که بازنویسی سمبولیک آن بصورت زیر است.

گزاره A: من یک باخ گوش می دهم. گزاره B: اتاق گرم است.

گزاره C: من احساس خواب می کنم.

این عبارت می توانند در نوشتار منطقی بصورت زیر نوشته شوند.

$$A \wedge B \rightarrow C$$

علائم ارتباطی

علائمی که در جدول 1-3 نشان داده شده اند برای بیان بعضی از عمومی ترین علائم ارتباطی منطق گزاره ای کاربرد دارند.

1-Modus Ponens

جدول 1-3 : کاربرد علائم ارتباطی در منطق گزاره ای

علامت	معنی	تفسیر
$\sim A$	نقیض A	نقیض گزاره A که ارزش آن عکس ارزش گزاره A می باشد.
$A \wedge B$	B, A	ترکیب عطفی A, B: تنها در صورتی صحیح است، که A, B

		هر دو درست باشد.
AVB	A یا B	ترکیب فصلی A یا B: اگر A یا B درست باشد ارزش این عبارت درست است.
$A \rightarrow B$	A نتیجه می دهد B	اگر A درست باشد و $A \rightarrow B$ هم درست باشد آنگاه B درست است و اگر A نادرست باشد و $A \rightarrow B$ درست باشد بنابراین B می تواند درست یا غلط باشد وزمانی این عبارت جواب نمی دهد که A غلط باشد.

منطق محصولات¹

منطق گزاره ای برای حل برخی از مسائل کافی نیست، چراکه یک گزاره صرفاً ارزش درست یا غلط را مشخص میکند. منطق محصولات این مشکل را با تقسیم گزاره به دو قسمت حل کرده است. این دو قسمت آرگومان و اعلان می باشد. همچنین منطق محصولات اجازه استفاده از متغیرها را داده و علاوه بر این قوانین استنتاج بکاررفته و در منطق گزاره ای راپشتیبانی می کند.

برای مثال گزاره زیر را در نظر بگیرید.

“موهای امی قهوه ای است.” Amy has brown hair

این جمله را در منطق محصولات می توان به صورت زیر نوشت:

1- Predicate logic

HAS (Amy, brown hair)

(موهای قهوه ای و امی) دارد.

در مثال بالا “دارد” (Has)، محمول است و امی و موهای قهوه ای (Amy, brown hair) آرگومان

هستند. حال به گزاره زیر توجه کنید.

Minsky is a cat

مینسکی یک گربه است

IS-A (Minsky, cat)

در منطقه محمولات به این صورت نوشته

سورها در منطقه محمولات

منطق محمولات اجازه استفاده از سورها را می دهد به این معنی که زبان می تواند به وسیله جمله اعلانی توسعه یابد و هر کدام از این جملات به یک محدوده از یک متغیر اطلاق شود. به عنوان مثال جمله زیر را در نظر بگیرید:

“هر مردی یک زن را دوست دارد”

این جمله در منطق محمولات با استفاده از سورها به صورت زیر نوشته می شود:

+++++^

که خوانده می شود: برای هر موجودیت X در جهان ، اگر X مرد است پس موجودیت دیگری وجود دارد که Y می باشد که این Y زن است و X ، Y را دوست دارند.

سور \forall یک سور عمومی نامیده می شود که به همه موجودیت ها به طور عمومی ، مردان اشاره می کند . کمیت E+++ سور وجودی نامیده می شود که به حد اقل یک موجودیت در جمعیت زنان اشاره دارد. اکنون این جمله را در نظر بگیرید : “هر کسی که اهل ولز است یک مرد است” که این جمله در فرم منطقی به صورت زیر است:

+++++==

که خوانده می شود: برای هر X که اهل ولز باشد پس X مرد است.

از دو مورد فوق و با استفاده از قوانین استنتاج نتیجه می گیریم که جمله زیر باید صحیح باشد:

+++++

به این معنی که هر کس اهل ولز باشد یک زن را دوست دارد. این مثال ما را به یک نتیجه گیری واضح سوق می دهد. اما در مورد مثالهای دیگر شاید نتایج وضوح کمتری داشته باشند. مزیت منطق صوری این است که با استفاده از مجموعه قوانینی که قوانین استنتاج نامیده می شوند، می توان از واقعیت های صحیح برای ایجاد واقعیت های دیگر استفاده کرد. علاوه بر این صحت هر گزاره جدید می تواند بوسیله یکسری اصول شناخته شده در مورد آزمایش قرار بگیرد. به هر حال توجه به این نکته مهم است که استنتاج های منطقی تنها صحت نتایجی را تضمین می کند که مقدمات رسیدن به این نتایج خود نیز صحیح باشند.

نمایش ترکیبی

هیچ روشی برای نمایش دانش به تنهایی برای هر محدوده از مسائل مناسب نخواهد بود. هر شکل نمایش دانش مزیت هایی دارد و همچنین دارای معایب است، که این مزایا و معایب در جدول 3-3 خلاصه شده اند. بعضی اوقات یک نمایش ترکیبی بهتر از یک نمایش منفرد است. ابزارهای برنامه نویسی هوش مصنوعی قوانین هستند. قاب ها و منطق نیز توانایی نمایش دانش به شکل ترکیبی را از طرق مختلف دارند. این ابزارها اغلب در عمل به خوبی کار می کنند. برای مثال در قاب ها از اسلات می توانت برای نمایش محمولات استفاده نمود و بنابراین استنتاج منطقی می تواند در قاب کاربرد داشته باشد. همچنین در یک سیستم قاب می توان از سیستم های تولید (مبتنی بر قوانین) استفاده نمود که موجب استفاده از دانش رویه ای و هیوریستیک می شود.

خلاصه

همانطوریکه قبلاً گفته شد، در بسیاری مواقع یک ساختار نمایش منفرد برای تمامی محدوده های کاربردی مناسب نمی باشد.

جدول 3-3 مزایا و معایب روش های نمایش دانش.

معایب	مزایا	روش نمایش
نمایش دانش توصیفی ، همچنین مشکل جاسازی محدوده دانش حل مسأله وجود دارد.	واحدمند، انعطاف پذیر و برای بسیاری دامنه ها (موضوعات) مناسب هستند.	قوانین تولید
مبهم - مدل فیزیکی ضعیف - عدم قطعیت را نمایش نمی دهد.	دقت. به این معنی که اگر مقدمات درست باشد . صحت نتایج را تضمین می کند . برنامه ها را به سمت اعلانی سوق می دهد . مثل پرولوگ	منطق
نمی توانیم دو کلاس از یک شیء را متمایز کنیم و تشخیص دهی. برخلاف قاب ها نمایش دانش رویه ای میسر نمی باشد. نمایش ساختار دانش برای سیستم های پیچیده غیر ممکن است.	روش نمایش مبتنی بر شیء است. بنابراین برائت مجاز است.	شبکه های معنایی
مشکلات تئوری ناشی از استلات هایی که باید در یک قاب تعریف شوند که این امر مسأله ای غیر قابل اجتناب می باشد. به این معنا که هیچ چیزی به طور کامل مشخص نیست و نمی توان برای شیء تعریف کلی ارائه نمود.	نمایش مبتنی بر شیء استفاده مجدد را آسان می کند. قاب امکانات لازم برای مدیریت استثناها و رویه ها را عرضه می کند که این ویژگی ها در نمایش منطق و سایر روش های نمایش به سادگی محقق نمی شود.	قاب ها

برای جمع آوری قوانین خبره در هر زمینه باید در آن زمینه مسلط بود. سیستم های تولید (مبتنی بر قوانین) برای جمع آوری قوانین از شخص خبره ایده ال هستند منطق در نمایش دانش با ساختار جزئی موفق است و پایگاه های دانش شامل منطق و قاب برای نمایش اشیاء و موجودیت ها

بسیار مناسب هستند. در جدول 3-4 مشخصات اصلی هر کدام از چهار روش نمایش دانش آورده شده است.

جدول 3-4: خلاصه ای از ویژگی های روش های نمایش دانش.

روش های استنتاج	معماری پایه	روش نمایش
زنجیره حرکت روبه جلو یا روبه عقب، در زنجیره حرکت روبه جلو، مفسر در پایگاه داده جستجو می کند تا وقایع جورشده در سمت چپ قوانین را بیابد. در زنجیره رو به عقب مفسر سعی می کند شرایطی را بیابد که منجر به اثبات یک هدف گردد.	شامل مجموعه ای از قوانین و پایگاه داده ای که وقایع را می داند، با توجه به واقعیت های قوانین موجود مفسر سیستم حقایق جدیدی را کشف نموده به پایگاه داده اضافه می کند این که چه قانونی انتخاب شود بستگی به روش استنتاج دارد.	قوانین تولید
تفکیک پذیری	در قضایا داده می شود، از قوانین استنتاجی هم مانند قیاس استثنائی استفاده می کند. سیستم خروجی بر اساس درخواست ها می باشد	منطق
وراثت به وسیله ارتباطات و گره ها	نمایش مبنی بر شی از ارتباطات و گره ها برای نمایش دانش استفاده می کند.	شبکه های معنا
وراثت در قاب، به صورت سلسله مراتبی، مقدار اسلات می تواند تعریف شده باشد	نمایش مبنی بر شی گرایی، از اسلات ها برای ذخیره کردن صفات خلاصه استفاده می کند. مقادیر قراردادی می توانند به اسلات ها نسبت داده شوند، بنابراین وراثت را آسان می کند.	قاب ها

خلاصه مطالب فصل سوم

دانش می تواند به طرق مختلفی نمایش داده شود.

◀ کامل بودن

◀ اختصار

◀ محاسبات دقیق و کافی (بازدهی خوب)

◀ شفافیت (وضوح)

◀ ساخت موارد مهم و پیشگیری از جزئیات اضافی

گروه بزرگی از سیستم های خبره از قوانین که گاهی اوقات "تولید" نامیده می شوند.

مزایای قوانین

◀ سادگی

◀ واحد مندی

◀ تشریح (توصیف)

◀ عدم قطعیت

یک قاب در واقع الگویی است که برای نگهداری اطلاعات یک موجودیت بکار می رود. هر قاب دارای قسمتهایی بنام اسلات (Slat) (شیار) می باشد که اسلاتها شامل صفاتی از آن موجودیت می باشند.

منطق بصورت گسترده ای در برنامه های هوش مصنوعی استفاده می شود. مهمترین بحث در منطق درستی یا نادرستی آرگومان ها می باشند. نوعاً آرگومان ها شامل گزاره ها و نتایج می باشند.

مزیت منطق صوری این است که با استفاده از مجموعه قوانینی که قوانین استنتاج نامیده می شوند، می توان از واقعیت های صحیح برای ایجاد واقعیت های دیگر استفاده کرد. علاوه بر این صحت هر

گزاره جدید می تواند بوسیله یکسری اصول شناخته شده در مورد آزمایش قرار بگیرد. به هر حال توجه به این نکته مهم است که استنتاج های منطقی تنها صحت نتایجی را تضمین می کند که مقدمات رسیدن به این نتایج خود نیز صحیح باشند.

مهارت های مورد نیاز برای مهندسی دانش

دبناهم در سال 1919 4 مهارت اصلی برای مهندس دانش بیان کرد که عبارتند از:

- 1- توانایی اصلی دقیق و کامل دانش. باید در مورد صحت دانش ، بررسی صورت گیرد. تمامی موضوعات و مسائل ممکنه احتیاج به بررسی کامل دارند و همه خروجی های ممکن باید در نظر

گرفته شوند. برای مثال، در یک سیستم خبره پزشکی متخصص ممکن است یک بیماری خاص و نادرست را به خاطر عدم شناخت کافی از قلم بیاندازد.

2- توانایی ارائه و پردازش دانش به این منظور که تسلیم بعضی ابزارهای خاص توسعه سیستم خبره نشود. به این صورت که دانش اخذ شده از یک متخصص برای ورودی کامپیوتر باید مستقل از ابزارها باشد. در غیر این صورت خطر این وجود دارد که دانش بجای اینکه در محدوده کاربردش قالب بندی گردد برای تطبیق با ابزارهای بکاررفته ساختار بندی شود.

3- توانایی طراحی یک سیستم خبره برای نگهداری دانش. در یک سیستم خبره دانش باید به گونه ای که متخصص (خبره) آن را تفسیر می کند، ارائه شود. این موضوع ممکن است بعضی اوقات یک سری مشکلات برای نگهداری ایجاد کند. چرا که در سیستم های مبتنی بر قوانین تغییر یک یا چند قانون می تواند تأثیر غیر قابل پیش بینی روی دیگر قوانین پایگاه داشته باشد.

4- توانایی طراحی یک سیستم خبره به نحوی که بتواند با سیستم های قراردادی رایج هماهنگی و تطابق داشته باشد. کاربران به طور گسترده وابسته به داده های صفحه گسترده و یا پایگاه های داده ای بزرگ هستند. غالباً، متخصصین نیز باید (برای تصمیم گیری هایشان) به چنین داده هایی دسترسی داشته باشند. بنابراین برخی از سیستم های خبره که برای توسعه و گسترش پیشنهاد می شوند احتیاج دارند که به این داده ها دسترسی داشته باشند و با دیگر برنامه های قراردادی رایج همانند پایگاه های داده ای و صفحات گسترده و نظایر آنها ارتباط داشته باشند.

اکتساب و اخذ دانش

واژه های اخذ دانش و اکتساب دانش برخی اوقات بجای یکدیگر به کار می روند. یک اختلاف کوچک بین این دو واژه وجود دارد. اخذ دانش شامل مراحل است که دانش در قلمرو خود

توسط متخصصین به دست می آید و از طرفی دیگر منظور از اکتساب دانش شامل تمام مراحل است که طی آن دانش به فرم قابل استفاده در یک سیستم خبره تبدیل می گردد. راهنماهای مربوط به مهندسی دانش که توسط هالیس روت، واترمن و نیات در سال 1983 منتشر شد، به عنوان روشی منظم برای ساخت سیستم های خبره استفاده شده، ابزارهایی را که برای توسعه سیستم های خبره لازم است مشخص می نماید. همچنین اکتساب دانش را به عنوان موضوع مهمی در توسعه سیستم های خبره معرفی می کند.

به طور کلی مهندسی دانش راجع به اکتساب دانش و چهار فعالیت دیگر صحبت می کند که عبارتند از:

- 1- تحقیق در مورد اعتبار و صحت¹.
- 2- نمایش دانش.
- 3- استنتاج.
- 4- توضیح و تصمیم گیری.

اکتساب دانش

اولین وظیفه مهندس دانش آشنایی با محدوده کاربردی مورد نظر به وسیله درک مفاهیم پایه ای و فرضیه ها می باشد. این اطلاعات اغلب در کتاب ها، مراجع، مستندات و امثال آن یافت می شوند.

1. Validation and Verification (V & V)

اما از آنجایی که این منابع دانش به سرعت کهنه می شود (به روز نیستند) به مصاحبه با افراد متخصص نیز نیاز می باشد.

این مسأله که بخواهیم ساختار یک محدوده را بدون مصاحبه و گفتگو با یک متخصص به دست آوریم مشکل است، چون اطلاعات موجود در منابع و کتاب ها ممکن است پراکنده یا ناقص باشند.

متخصصان دائماً در حال یادگیری مطالب جدید هستند بدون آنکه لزوماً این دانش جدید را در قالب اسناد کتبی به روز در آورند.

مصاحبه

مصاحبه اولین مرحله کسب دانش توسط انسان است. یک مصاحبه موفق شامل طرح، آماده سازی، ثبت و ایجاد مستندات می باشد. این مراحل برای کلیه مصاحبه ها از جمله مصاحبه با متقاضیان کار صادق است. ایجاد طرح و آماده سازی عناصر مهمی برای مصاحبه به منظور اخذ دانش هستند. فراموش نکنیم که کیفیت یک سیستم خبره به تنهایی می تواند از طریق دانش جمع آوری شده در آن مشخص گردد. اگر متخصص به خوبی آمادگی نداشته باشد، ممکن است پاسخ های نادرستی به سؤالات بدهد. از طرف دیگر ممکن است سؤالاتی برای مصاحبه مطرح می شوند اشکالاتی داشته باشند. مرحله آماده سازی در مصاحبه در رابطه با مسائل زیر صحبت می کند:

- 1- توضیح در مورد اهداف و روند مصاحبه، که به منظور آماده نمودن متخصص می باشد، به طوریکه از نظر فکری برای مصاحبه آماده شود.
- 2- چه عواملی بر پیشرفت کار اثر گذار هستند؟ برای مثال مهندس دانش ممکن است بیان کند که چگونه درک و دریافت دانشی که خبره در آن تخصص دارد بر روند پیشرفت کار تأثیر گذار خواهد بود.
- 3- تجهیزات خاص که استفاده می شوند و باید تنظیم شوند. برای مثال باید مصاحبه روی نوار کاست یا ویدئو ضبط شود و یا یک دستیار در طول مصاحبه یادداشت برداری کند.

4- شرایط محیطی که مصاحبه در آن انجام می شود. برای مثال چه نوع اتاقی باید آماده شود. یا چه زمانی از روز برای مصاحبه در نظر گرفته شود.

تکنیک های مصاحبه مستقیم

بعد از آماده شدن شرایط مصاحبه مرحله بعدی انجام مصاحبه بصورت موفقیت آمیزی باشد. تکنیک های مصاحبه مستقیم شامل سه گروه عمده ذیل می باشد.

- مصاحبه جهت دهنده.

- مصاحبه ساختیافته.

- مصاحبه متفکرانه.

مصاحبه جهت دهنده

اولین بخش مصاحبه، جهت دهی یا گرایش و تمرکز بر روی مصاحبه می باشد. هدف از این نوع مصاحبه به دست آوردن یک دید کلی در محدوده کاربردی مورد نظر است. مهندس دانش به متخصص اجازه می دهد که آزادانه در مورد اطلاعات تخصصی در محدوده گفتگو صحبت کند و تنها با سؤالات کلی برای هدایت متخصص، دخالت می نماید. زمانی که متخصص بحثی را با واژه های سنگین و سطح بالا مطرح می کند به او گوشزد خواهد کرد. این تکنیک برای کسب دانش عمومی محدود مورد مطالعه بکار می رود.

در استفاده از این تکنیک مهندس دانش ممکن است چند سؤال کلی که باعث جهت دهی به مصاحبه شود را مطرح نماید.

مانند سؤالات زیر:

◀ “آیا فکرمی کنید با کلمات ساده و روان توضیح دهید که چه کاری انجام می دهید؟”
و یا “فکر می کنید، وظیفه اصلی شما چیست؟”

◀ مشکل اصلی شما چیست؟

مصاحبه ساختیافته

این نوع مصاحبه اساساً به عنوان مرحله دوم مصاحبه و با هدف تعمیق بیشتر در گفتگو و دستیابی به جزئیات بیشتر صورت می گیرد. مهندس دانش وقتی جهت دهی مصاحبه را انجام داد ممکن است از مصاحبه شوندگان بعنوان فرصتی برای درک مفاهیمی که هنوز مطرح نشده اند استفاده کند به همین دلیل گفتگو در این نوع مصاحبه بیشتر از یک پروسه دو مرحله ای می باشد. چرا که متخصص به سوالاتی که مهندس دانش می پرسد پاسخ داده و به طور ایده آلی مصاحبه کننده به متخصص اجازه می دهد که پیرامون پرسش مطرح شده صحبت کند و او را از انحراف از مبحث اصلی بازمی دارد. از متخصص ممکن است سؤالاتی به شرح ذیل پرسد.

◀ “شما چگونه این مسئله را بدست آوردید؟” یا

◀ “آیا شما می توانید این مسئله را با جزئیات بیشتری توضیح دهید؟” یا

◀ “درک نمی کنم که چرا شما این کار را انجام می دهید؟” یا

◀ “در اینجا چه اتفاقی می افتد؟” و غیره

مصاحبه متفکرانه

مصاحبه های ساختیافته بعضاً نمی توانند تمامی مطالب را که در حوزه علمی مربوطه لازم است، در خود بگنجاند. مصاحبه بصورت گفتگوی متفکرانه همانطور که از نامش پیداست متخصص را تشویق می کند به اینکه در هنگام فکر کردن صحبت نماید (فیرج و هنلر در سال 1991). با استفاده

از این تکنیک، به متخصص کاری مشابه به کسی که در یک محیط حل مسئله دارد، واگذار می شود. از متخصص خواسته می شود تمامی افکاری را که در مورد کارهایش دارد بیان کند. روش مصاحبه به صورت گفتگوی متفکرانه بعد از مصاحبه ساخت یافته انجام می شود. بنابراین استراتژی حل مسئله توسط خبره و دانشی که از مصاحبه قبلی بدست آمده می تواند مورد ارزیابی قرار گیرد.

فلاسه ای از تکنیک های مصاحبه

تکنیک های اصلی که برای مصاحبه بکار می روند در جدول 1-4 خلاصه شده است.

جدول 1-4: تکنیک های مصاحبه.

روش مصاحبه	هدف و کاربرد	برفی مشکلات ممکن	برفی راه حل های ممکن
جهت دهنده	برای اخذ دانش در یک حوزه، در مراحل اولیه اخذ دانش کاربرد دارد.	1- متخصص برای جمع آوری اطلاعات مشکل دارد. 2- متخصص کلمات تخصصی و غیر قابل فهم رابه هنگام مصاحبه بکار میبرد.	باید دید دیگری به دامنه گفتگو بخشید. ممکن است متخصص احساس راحتی نداشته باشد. باید تمهیدات لازم را قبلاً برای مصاحبه فراهم نمود.
ساخت یافته	برای کسب جزئیات بیشتر و عمیق تر بکار می رود. مصاحبه کننده ممکن است توضیحات دقیق تر از مباحثی که در مساحبه جهت دهنده مطرح شد، لازم داشته باشد. مصاحبه های	1- دانش غلط یا ناقص. 2- دانش غیر قطعی.	لازم است مصاحبه کننده سؤالات بیشتری از متخصص پرسد. مثل این پرسش که: "اگر یک مفهوم ناشناخته باشد چه اتفاقی می افتد؟" یا اینکه "آیا این شرط، شرط دیگری را که خودش مقادیری می خواهد طلب می کند؟" و مانند اینها

	ساختاری در فاز دوم صورت میگیرد		
متفکرانه	برای تأیید دانش و فرضیاتی است که در مراحل قبلی به دست آمده و در گام سوم مصاحبه بکار می رود	1- شبیه سازی عملکرد متخصص به عنوان یک روش مفید، مشکل مییابد. 2- روش مصاحبه متفکرانه میتواند بدین نحو کار متخصص تأثیر بگذارد.	متخصص ممکن است احساس ناراحتی کند، مصاحبه رابه طریق دیگر مطرح کنید.

سایر تکنیک های استخراج دانش

یک متخصص همیشه نمی تواند صرفاً با استفاده از روش های مصاحبه مستقیم جزئیات دانش و چگونگی کاربرد آن را ابراز نماید. به عنوان مثال اخذ یک تصمیم ممکن است به عوامل ترکیبی متعدد بستگی داشته باشد که اظهار نمودن تمامی آنها مشکل بوده و زمان زیادی می طلبد. تکنیکهای دیگری نیز برای استخراج دانش مورد استفاده قرار می گیرند. برخی از آنها به طور خلاصه در زیر توضیح داده شده اند.

شبکه فهرست¹

یک روانشناس به نام جی کلی در سال 1955 مفهوم شبکه فهرست را مطرح کرد. این طرح در واقع نمایشی از نظرات یک متخصص راجع به یک مسأله خاص می باشد. در این روش یک سری عناصر بر روی یک محور بر روی یک صفحه دوبعدی نمایش داده می شوند. این عناصر برای خبره موجودیت هایی آشنا هستند. مانند لیستی از اشخاص یا اشیاء همانند اتومبیل ها. سپس ویژگی

های این موجودیت بر روی محور دیگر به آن نسبت داده می شود مانند رنگ، وزن و... یک رویه عمومی در این روش این است که اطلاعات در رابطه با مجموعه ای از موجودیت ها را به ترکیب های مختلف دسته بندی نموده از متخصص در رابطه با تشابهات و تفاوت های گروه های مختلف سؤال پرسیده شود. به این طریق تمامی ترکیبات ممکن بررسی شده تفاوت هایی که ممکن است بین موجودیت های مختلف مطرح باشد روشن می گردد.

تکنیک های بصری

یک روش دستیابی به این هدف که متخصصان چه کاری انجام می دهند این است که هنگام مسأله به کار آنها نگاه کرد. یک سری از این تکنیکهای قابل دسترسی، تحلیل پروتکل نامیده میشوند.

تملیل پروتکل²

این روش بر اساس تکنیک مصاحبه متفکرانه پایه گذاری شده است. از متخصص هنگامیکه در حال انجام کارش می باشد سؤال می شود تا توضیح دهد که چه می کند. یک نوار کاست یا ویدئو تهیه شده و سپس مطالب روی آن تفسیر و تحلیل می شوند. مزیتی که این روش دارد این

-
- 1.Repeytory grid
 - 2.Protocol analgsis

است که فعالیت ها کاملاً بر طبق روال طبیعی بررسی و تکمیل می شوند. دیگر تکنیک های موجود برای اکتساب دانش عبارتند از:

◀ مشاهده ▶ مطالعات موردی ◀ نقش بازی¹

مشاهده

یکی از عمده ترین مشکلاتی که برای مصاحبه پیش می آید درخواست وقت از متخصص می باشد. این مشکل برای روش مشاهده مطرح نیست. چون مشاهده متخصص در حین انجام کار هیچ وقت اضافی را نمی طلبد. با مشاهده یک متخصص هنگام کار، یک مهندس دانش می تواند بوضوح کاربرد آن دانش را هم ببیند. مشاهده همچنین می تواند با بخش های قبلی (مصاحبه) ترکیب شود تا مهندس دانش قادر شود فرضیات دامنه موجود را تحقق بخشد یا اصلاح نماید. از معایب روش مشاهده این است که این روش وقت گیر می باشد و ممکن است فقط موارد محدودی از مصاحبه را پوشش دهد. همچنین یک متخصص هنگامی که بداند دیگران بر عملکرد او نظارت دارند ممکن است احساس ناراحتی و یا حتی خجالت نماید و بنابراین غیر طبیعی رفتار کند. تکنیک های مشاهده موفق نیاز به تحلیل دقیق محیط عملیاتی دارند و به این دلیل مهندس دانش باید این بخش را تا حد امکان محرمانه انجام دهد.

مطالعات موردی

از جمله مشکلاتی که در روش مشاهده ممکن است وجود داشته باشد مسائل غیر قابل پیش بینی هستند که ممکن است رخ دهند. و این مسأله باعث اتلاف وقت مهندس دانش که در حال مشاهده عملیات است، شود. با استفاده از مطالعات موردی می توان براین مشکل فائق آمد. در اینجا

1. Role play

مهندس دانش می تواند مواردی از توضیحات متخصص را انتخاب نماید. مانند روش مشاهده، مطالعات موردی بعضاً بسیار وقت گیر و پرهزینه هستند. به دلیل اینکه متخصص باید همه مواردی که نتایج و عواقب کار را پوشش می دهد در نظر داشته باشد.

نقش بازی

این تکنیک نیز می تواند وقت گیر باشد و ممکن است تمامی موارد ممکن را پوشش ندهد. نقش بازی، همچنین می تواند در عمل بسیار مشکل باشد و نیاز به یک مهندس دانش دیگر است، تا کارایی بیشتر شود.

مشکلات اکتساب دانش

مشکلات اکتساب دانش اغلب در زمینه توسعه سیستم های خبره بوجود می آید. جدول 4-2 برخی از این مشکلات را به همراه راه حل های ممکن نشان می دهد.

جدول 4-2: مشکلات اکتساب دانش.

مشکل	راه حل ممکن
مهندس دانش برای دریافت قوانین از متخصص مشکل دارد. روابط بین آنها مستلزم صرف وقت است.	سعی کنید که یک متخصص فعال و آشنا به کامپیوتر انتخاب کنید. این مسأله با تحقیق و جستجو در مورد عقاید و نظرات متخصصین محقق می شود. همچنین به یاد داشته باشید که متخصص باید خوش برخورد باشد.
متخصص حوزه مربوطه وقت کافی برای این پروژه ندارد.	این مشکل کلی است و راه حل آن این است که باید از شروع پروژه از همراهی متخصص تا پایان کار مطمئن شد.
قوانین ارائه شده توسط متخصص کوتاه و ساده بوده و در موارد پیچیده دقت ندارند.	از کاربردن مسائل ساده پرهیز کنید و سعی کنید با استفاده از مسائل واقعی متخصص را درگیر نمایید و ازداده حقیقی استفاده کنید. برای مثال تست های آزمایشگاهی و غیره سپس نوع دیگری از مسائل را مطرح نمایید.
متخصص ممکن است در مورد این پروژه بی میل باشد و زمانی که برای کار روی پروژه اختصاص می دهد بعد از مدتی کم شود.	مطمئن شوید که مهندس دانش و متخصص بر اساس یک برنامه منظم (حداقل هفتگی) همدیگر را ملاقات می کنند. همچنین باید متخصص را ملزم به ایجاد راه حل های ممکن نمود.

<p>متخصص با کامپیوتر آشنا نیست و فوائد سیستم های خبره را نمی داند</p>	<p>تدریجاً عملکرد یک سیستم خبره را معرفی کنید. در این گونه موارد مدل سازی می تواند مزایای سیستم های خبره را برای متخصص آشکار سازد.</p>
<p>از تعداد زیادی متخصص استفاده شده است و مهندس دانش زمان کافی برای بررسی عمقی مطالب آنها را ندارد.</p>	<p>دروغله اول سعی کنید از خبرگان زیادی استفاده نکنید. همچنین مهم است که یک رابطه خوب کاری با متخصصین برقرار شود. بخاطر داشته باشید عملیات اکتساب دانش نیاز به این دارد که متخصص چیزهایی را در مورد سیستم خبره بیاموزد و مهندس دانش مسائلی را در محدوده دانش یاد بگیرد.</p>

تمرینات

- 1- مشکلات احتمالی موجود در روش مصاحبه متفکرانه چیست؟ چگونه می توان بر این مشکل فائق شد؟
- 2- با تحقیق و جستجو در مراجع دو سال گذشته، سیستم های خبره ای را که در تجارت کاربرد داشته است به عنوان مثال ذکر نمایید. سیستم را مختصراً توضیح داده و آنرا نقد کنید.
- 3- به اختصار توضیح دهید که یک مهندس دانش چه کاری انجام میدهد و چه نیازی به وجود اوست؟
- 4- دوشیوه کلی اکتساب دانش برای یک سیستم خبره را شرح دهید.
- 5- تحلیل پروتکل چیست؟ توضیحی همراه با یک مثال ارائه دهید و یک مزیت و یک عیب این روش را مطرح کنید.

6- به نظر شما برای اخذ دانش ارزشمند به چه چیزی نیاز است؟

7- یک شرکت خصوصی و کوچک حفاری معدن تصمیم دارد که یک سیستم خبره برای ساخت

ماشین حفاری بسازد در این سیستم خبره اهداف زیر باید در نظر گرفته شوند: اول اینکه ماشین

حفاری باید کمترین زمان پرت رداشته باشد. علاوه بر این سیستم باید طوری طراحی شده

باشد که وظیفه تکنسین را کاهش داده و زمینه را برای آموزش تکنسین های جدید فراهم

کند. سیستم باید قادر باشد تا داده ها را از صفحات گسترده بگیرد و تشخیص دهد که آیا

بارگذاری روی ماشین حفاری بیشتر از حد معمول بوده است یا خیر. نهایتاً سیستم باید

دیاگرامهایی برای تعیین موقعیت خطا و همچنین ارائه راه کارها ایجاد کند.

الف) مزایا و معایب بکارگیری روش های زیر چیست؟

1) مشاهده . 2) مطالعه موردی . 3) نقش بازی.

ب) چه روش یا روش هایی را برای اخذ دانش در این مورد ترجیح می دهید؟ چرا؟

یکی از سه روش فوق را برای پاسخگویی به سوالات انتخاب کنید.

8- بعد از مطالعه یک گزارش در مورد سیستم های خبه پروفیسور نووال تصمیم به ساخت یکی از

سیستم ها گرفت . طرح او ساخت یک دستگاه نوشتاری با استفاده از کلام بود. این دستگاه در واقع

یک ماشین تایپ هوشمند است که می تواند صحبت هایی که از طریق میکروفن دریافت

می کند را تایپ کند.

قدم اول او پیدا کردن چند شنونده و تایپیست متخصص بود. به این دلیل که منشی ها باید هم نت

بردارند و هم حروف را تایپ کنند که افراد کمی قادرند این کار را انجام دهند و این کار به دانش

نیاز خواهد داشت. او از منشی های سازمان خود استفاده کرد و با آنها مصاحبه نمود تا بفهمد که چگونه این کار را انجام می دهند.

الف) پروفیسور نوال برای رسیدن به هدفش به چه دانشی نیاز دارد؟ پیشنهادات خود را در قالب روش های زیر ارائه دهید.

◀ مهارت های تایپی. ▶ مهارت های شنیداری.

◀ مهارت های نکات دستوری زبان، حروف اضافه و علامات خاص دستور زبان.

◀ مهارت های پردازش لغات. ▶ مهارت های ایجاد مستندات.

◀ مهارت های بحث و بررسی.

برای هر مهارت به طور خلاصه بگویید که چه مؤلفه هایی از دانش مورد نیاز است؟ آیا منشی ها این مهارت را دارند؟ چگونه این دانش در چنین دستگاهی هوشمندی کاربرد دارد؟

ب) آیا این منشی ها، دانش فوق را دارند؟ آیا این دانش با روش های مصاحبه و تحلیل پروتکل قابل دستیابی است؟

خلاصه مطالب فصل چهار

دب‌نهام در سال 1919 4 مهارت اصلی برای مهندس دانش بیان کرد که عبارتند از:

1- توانایی اصلی دقیق و کامل دانش.

2- توانایی ارائه و پردازش دانش به این منظور که تسلیم بعضی ابزارهای خاص توسعه سیستم خبره نشود.

3- توانایی طراحی یک سیستم خبره برای نگهداری دانش.

4- توانایی طراحی یک سیستم خبره به نحوی که بتواند با سیستم های قراردادی رایج هماهنگی و تطابق داشته باشد .

اخذ دانش شامل مراحل است که دانش در قلمرو خود توسط متخصصین به دست می آید و از طرفی دیگر منظور از اکتساب دانش شامل تمام مراحل است که طی آن دانش به فرم قابل استفاده در یک سیستم خبره تبدیل می گردد .

اولین وظیفه مهندس دانش آشنایی با محدوده کاربردی مورد نظر به وسیله درک مفاهیم پایه ای و فرضیه ها می باشد. این اطلاعات اغلب در کتاب ها، مراجع ، مستندات و امثال آن یافت می شوند. اما از آنجایی که این منابع دانش به سرعت کهنه می شود (به روز نیستند) به مصاحبه با افراد متخصص نیز نیاز می باشد.

مصاحبه اولین مرحله کسب دانش توسط انسان است . یک مصاحبه موفق شامل طرح ، آماده سازی ، ثبت و ایجاد مستندات می باشد.

مرحله آماده سازی در مصاحبه در رابطه با مسائل زیر صحبت می کند:

- 1- توضیح در مورد اهداف و روند مصاحبه
- 2- چه عواملی بر پیشرفت کار اثر گذار هستند
- 3- تجهیزات خاص که استفاده می شوند و باید تنظیم شوند.
- 4- شرایط محیطی که مصاحبه در آن انجام می شود.

تکنیک های مصاحبه مستقیم

- مصاحبه جهت دهنده.
- مصاحبه ساختیافته.

- مصاحبه متفکرانه.

یکی از عمده ترین مشکلاتی که برای مصاحبه پیش می آید درخواست وقت از متخصص می باشد. این مشکل برای روش مشاهده مطرح نیست. چون مشاهده متخصص در حین انجام کار هیچ وقت اضافی را نمی طلبد. با مشاهده یک متخصص هنگام کار، یک مهندس دانش می تواند بوضوح کاربرد آن دانش را هم ببیند.

مشکلات اکتساب دانش اغلب در زمینه توسعه سیستم های خبره بوجود می آید.

استنتاج

درفصل دوم دیده شد که اجزاء اصلی یک سیستم خبره، موتوراستنتاج، پایگاه دانش و واسط کاربر می باشند که تمامی این اجزاء مفهوم واقعی ارتباط بایک سیستم خبره را ایجاد می کنند. این فصل

جزئیات استنتاج رامورد بررسی قرارداد، مکانیزم های استنتاج راکه در ساختاریک سیستم خبره مورد استفاده قرار می گیرند مورد بحث قرار می دهد. استنتاج مبحث مهمی است زیرا شامل تکنیک هایی است که سیستم خبره توسط آنها مسائل راحل می کند. واژه استنتاج در فرهنگ لغت به این صورت تعریف شده است: “ رسیدن به دانش به وسیله استدلال ”. در سیستم های مبتنی بر قانون این استدلال بر اساس نتیجه گیری از مقدمات و فرضیات صورت می گیرد. این امر به وسیله موتور استنتاج قابل تحقق می باشد.

عملکرد موتور استنتاج

در سیستم های مبتنی بر قانون، موتور استنتاج به این صورت کار می کند که یک قانون را برای تست انتخاب می کند و بررسی می کند که آیا شرایط این قانون صحیح هستند یا خیر. این شرایط ممکن است از طریق سؤال از کاربر بررسی شود و یا ممکن است از واقعیت هایی ناشی شود که در طول مصاحبه بدست آمده اند. وقتی شرایط مربوط به یک قانون صحیح باشند آنگاه نتیجه آن قانون نیز درست خواهد بود. پس این قانون فعال¹ شده و نتیجه آن به پایگاه دانش افزوده می شود. همچنین ممکن است این نتیجه به عنوان اطلاعات در واسط کاربر نمایش داده شود. برای مثال قوانین زیر را که مربوط به یک سیستم خبره مشاور در مورد گیاهان خانگی است در نظر بگیرید.

قانون 1: اگر اتاق سرد باشد و نور اتاق کم باشد آنگاه بهترین گیاه پیچک می باشد.

1. Fired

قانون 2: اگر دما کمتر از 55 باشد آنگاه اتاق سرد است.

اگر موتور استنتاج سعی در اثبات قانون داشته باشد آنگاه مقادیری راکه در شرایط این قانون آورده شده است احتیاج دارد که این شرایط عبارتند از: “ اتاق سرد است ” و “ نور کم باشد ”. شرط “ اتاق

سرداست ” با اثبات قانون دوم بدست می آید برای اینکه این شرط از قانون دوم می باشد و بنابراین به وسیله اثبات این قانون حاصل می شود. به این معنی که شرایط برای قانون دوم یعنی ” دمای کمتر از 55 درجه ” باید درست باشد. شرط ” نور کم است ” باید از کاربر پرسیده شود چرا که نتیجه قانون دیگری نیست و بنابراین ارزش آن فقط با پرسش از کاربر حاصل می گردد.

در مثال فوق موتور استنتاج می تواند برای یافتن بهترین گیاه از قانون 1 شروع کند و با اثبات فرضیات به هدف برسد. همچنین موتور استنتاج می تواند از قانون 2 شروع کند و سپس به طرف حل صحیح مسأله پیش برود.

استراتژی های استنتاج

معمولاً سه روش استنتاج توسط متخصصین بکار برده می شود: استنتاج قیاسی ، استنتاج استقرایی . استنتاج انتزاعی .

استنتاج قیاسی

استنتاج قیاسی به اختصار در فصل دوم بررسی شد. مثال شامل سه قضیه زیر بود:

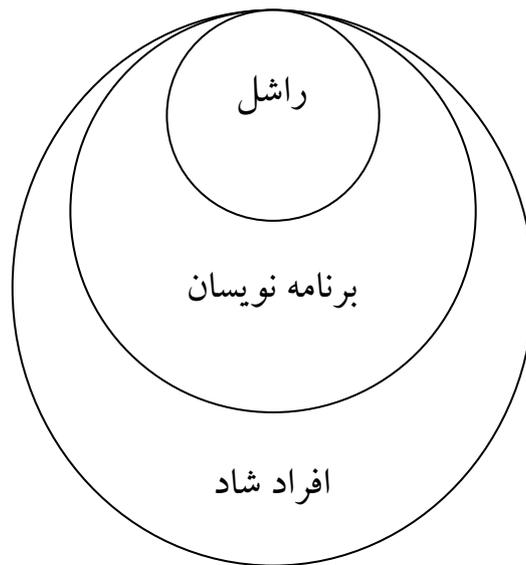
(1) راشل یک برنامه نویس است .

(2) همه برنامه نویس ها شاد هستند.

(3) بنابراین راشل شاد است .

نتیجه (3) یک نتیجه منطقی و یا قیاسی از قضایای (1) و (2) می باشد. در این مثال نتیجه می تواند از اطلاعات داده شده استنباط شود و این یک مثال از استنتاج قیاسی است .

جمله 1 قضیه اصلی نامیده می شود و جمله 2 قضیه فرعی نامیده می شود. نتیجه بدست آمده 3 حاصل منطقی قضایای 1 و 2 است که در نمودارون در شکل 5-2 نمایش داده شده است .



شکل 5-2 : نمودارون برای مثال ارائه شده

مزیت استنتاج قیاسی این است که این نوع استنتاج از نظر منطق ریاضی کاملاً درست است. به این معنی که اگر قضایای پیش فرض درست باشند نتیجه بدست آمده نیز مطمئناً درست است.

استنتاج استقرایی

جملات یا قضایای زیر را در نظر بگیرید:

4- مینسکی یک گربه است.

5) بنابراین مینسکی چهار پا دارد.

ابتدا ممکن است چنین تصور شود که نتیجه (5) که عبارت است از "مینسکی چهار پا دارد" به وسیله روش قیاسی از پیش فرض (4) بدست آمده است ولی این نتیجه گیری غلط است، چون جمله (4) چیزی راجع به اینکه یک گربه چهار پا دارد نمی گوید.

افراد بوسیله دانشی که از جهان طبیعی دارند و بر اساس تجربه خود این فرض را (گربه 4 پا دارد) می دانند و به نتیجه (5) می رسند. بنابراین نتیجه (5) از نظر ریاضی، دقت و صحت نتیجه (3) را ندارد. برای اینکه نتیجه (5) درست باشد (گربه 4 پا دارد) باید دانش ناشی از حواس پنج گانه

انسان به فرض (4) اضافه شود. به این دلیل است که چنین نتایجی، استنتاج استقرایی نامیده می شود. این چنین استنتاجی دقت ریاضی ندارد و همیشه این احتمال وجود دارد که نتایج اشتباه باشند. استنتاج استقرایی در سیستم های خبره عمومیت دارد چرا که با استنتاج های انسان در جهان حقیقی هماهنگی و سنخیت دارد.

استنتاج انتزاعی

استنتاج انتزاعی از معلول به علت می رسد و از این نظر با استنتاج استقرایی که از علت به معلول می رسد تقاضا دارد. برای مثال قانون "اگر هوا بارانی شد آنگاه چمن باغ خیس می شود" را در نظر بگیرید درستی مقدمه درستی نتیجه را در بر دارد و استنتاج انتزاعی بکار رفته برای این قانون درستی را به صورت معکوس در نظر می گیرد و به این صورت که "اگر چمن خیس باشد پس باران باریده است". این قانون به صورت کلی پذیرفته شده است حتی اگر تضمینی برای درستی آن وجود نداشته باشد.

ممکن است که چمن بدلیل آب پاشی و یا دلایل دیگر خیس شده باشد. بنابراین استدلال انتزاعی از نظر ریاضی به اندازه استدلال استقرایی هم صحت و دقت ندارد. اگر چه افراد متخصص غالباً استنتاج انتزاعی را بکار می برند. جدول 5-1 خصوصیات هر یک از استراتژی استنتاج را به طور خلاصه نشان می دهد.

فود آزمایی

استنتاج های زیر انتزاعی، قیاسی یا استقرایی می باشند؟ پاسخ خود را توضیح دهید.
الف) هیچ کس برنده نمی شود بنابراین هیچ کس نباید بازی کند.

ب) همه پرندگان می توانند پرواز کنند، چلچله ها همه پرنده هستند. بنابراین همه چلچله ها می توانند پرواز کنند.

جدول 5-1 مقایسه استراتژی های مختلف استنتاج.

استراتژی استنتاج	مشخصات
استنتاج قیاسی	تنها اطلاعات داده شده را برای حل مسائل به کار می گیرد و فقط به آن نیز نیاز دارد. استدلال ضرورتاً از عام به خاص صورت می گیرد. مفهوم ضرورت به این معنا است که اگر فرض درست باشد آنگاه نتیجه نیز درست است. به عبارت دیگر، نتیجه صحیح حاصل می شود اگر فرضیات درست باشند.
استنتاج استقرایی	جمع آوری دانش در این روش بیشتر به تجربه نیاز دارد. استدلال از خاص به عام می باشد و همیشه این احتمال وجود دارد که نتایج نادرست باشند.
استنتاج انتزاعی	استدلال از نتیجه و اثر (معلول) به علتی صورت می گیرد بنابراین این احتمال ایجاد نتایج نادرست وجود دارد.

استدلال مبتنی بر موارد (استدلال موردی)

استدلال مبتنی مورد¹ (CBR) با انتخاب یک مورد از مواردی که قبلاً در پایگاه اطلاعات ذخیره

1. Cases Based Reasoning

شده اند کار می کند. به نحوی که این مورد بهترین و بیشترین شباهت را به خصوصیات مسأله جاری تحت بررسی، دارا باشد. این روش مشابه روشی است که افراد متخصص هنگام تلاش برای حل انواع مسائل استدلال می کنند. به عنوان مثال تصمیم یک پزشک برای تعیین مقدار داروی خاصی جهت یک بیمار، که دردی دارد، ممکن است یادآور بیماری مشابه باشد که تجربه چنین

دردی را در چند هفته پیش داشته است. اگر میزان دارو قابل قبول بوده باشد آنگاه به نظر می رسد که تصمیم اتخاذ شده توسط پزشک مشابه تصمیم قبلی وی باشد.

CBR روشی از استنتاج است که اصاصاً با دیگر روش های استنتاج متفاوت است. CBR بجای تکیه بر دانش عمومی قلمرویک مسأله با ارتباط بین فرضیات مسأله و نتایج آن، قادر است تا دانش مخصوص به تجربه های قبلی را بجای وضعیت ها و حالات مسأله اصلی بکاربرده و مسائل را به هم پیوند دهد. این مسائل اصطلاحاً موارد¹ نامیده می شوند.

یک مسأله جدید با پیدا کردن یک مورد مشابه قبلی و استفاده مجدد آن به جای وضعیت جدید، حل می شود. در صورت لزوم ممکن است موردی هم باشد که برای راه حل ها مستقیماً بکار رود یا برای تغییر شکل مشکلات مسئله جدید استفاده شود. یک مورد (case) توضیحی از یک مسئله به همراه جزئیات عملکردش می باشد که پاسخی به حل مشکلات خواهد بود. بطور کلی حل یک مسئله با استفاده از CBR مراحل زیر را شامل می شود:

1- ثبت جزئیات مسئله فعلی.

2- تطبیق این جزئیات با موارد ذخیره شده به منظور یافتن وضعیت های مشابه.

3- انتخاب موارد (cases) ذخیره شده ای که با مسئله جاری بیشترین ارتباط دارند.

1.Cases

4- سازگار کردن (وفق دادن) راه حل ذخیره شده با مسئله جاری.

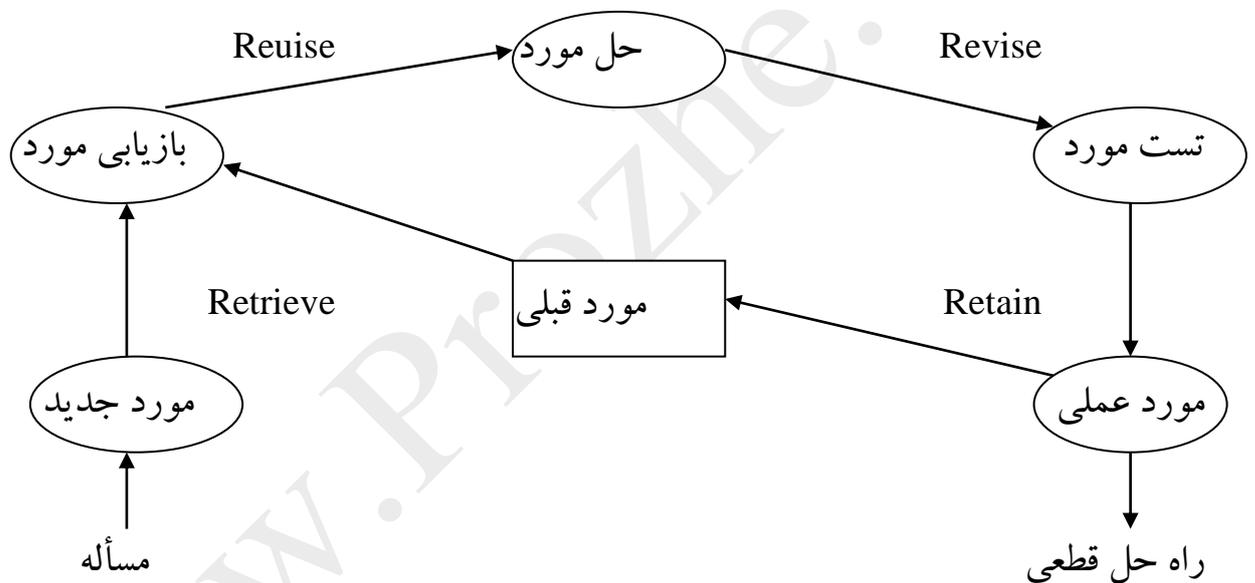
5- ارزش دهی به راه حل جدید و ذخیره جزئیات مورد (case) جدید.

این مراحل حل مسئله، برگرفته از چرخه یک نوع سیستم CBR که در شکل 3-5 نشان داده شده است، می باشد. 4 مرحله کلی که در شکل 3-5 نشان داده شده است به عنوان چهار R معروف می باشد که عبارتند از:

1- بازیابی¹ موارد مشابه به منظور بررسی.

2- استفاده مجدد² از اطلاعات و دانش، در آن مورد، برای حل مسئله.

3- تجدید نظر (بررسی مجدد) راه حل پیشنهادی.



شکل 3-5 چرخه حل مسأله CBR.

1.Retrieve

2.Reuse

4- نگهداری¹ بخش هایی از این تجربه، که احتمالاً برای حل مسائل آینده بکار برده می شوند.

یک مسأله جدید در CBR، باززیابی یک یا چند مورد از قبل ذخیره شده، استفاده مجدد مورد برای مسأله تحت بررسی، تجدید نظر راه حلی که براساس این مورد بنا شده است، و نگهداری مورد جدید با الحاق آن به دانش ذخیره شده، حل می شود.

زمانی که یک مورد (Case) باززیابی می شود، سیستم CBR از راه حل پیشنهادی براساس مورد باززیابی شده استفاده می کند. در بعضی موارد این راه حل کافی نخواهد بود، در چنین مواردی، سیستم CBR باید با استفاده از راه حل ذخیره شده در مورد باززیابی شده نیازهای مورد جاری را برآورده سازد. چندین روش برای ایجاد سازگاری و تطبیق در CBR بکاربرده شده است (واتسن در سال 1997).

کاربرد CBR

CBR در حالات زیر احتمالاً موفق باشد:

- ◀ زمانی که آزمایش یا تجربه به جای تئوری، منبع اولیه دانش است.
- ◀ هنگامی که راه حل ها بتوانند به جای آنکه تنها برای یک مسأله واحد به کار برده شوند، مورد استفاده مجدد قرار بگیرند.
- ◀ زمانی که به جای یک راه حل کاملاً تضمین شده، هدف بهترین راه حل ممکن باشد.

کاربرد CBR

نمونه های موفق کاربرد CBR شامل نرم افزار کمک راهنما¹، شناخت بیماری، و دامنه های کاربردی آموزش می باشد. CBR برای کاربردهای کمکی راهنما مناسب است. چرا که دانش

1.Retain
2.Helpdesk

کمک راهنما به وسیله شناخت تئوریک یک حوزه براحتی ایجاد نمی شود. بعنوان مثال کمک راهنما برای سخت افزار کامپیوتر برای شناسایی دیسک های سخت ، پردازشگرها و غیره به کار می رود که بطور ثابت در حال تغییر می باشند و بنابراین طرح یک مدل تئوری در این زمینه غیر ممکن است.

یک مثال دیگر کاربرد شناخت بیماری با استفاده از CBR ، CASEY می باشد (کوتون در سال 1989) ، که یک سیستم بهداشتی و سلامتی است و شناخت بیماری افراد را براساس شناخت افراد بیمار قبلی انجام میدهد. CASEY از قانون هایی استفاده می کند تا به مرحله تطبیق برسد.

برای مثال CASEY می تواند دانش خودش را به این صورت تغییر دهد که بی نظمی های قلبی و عروقی ارتباط با فشار خون بالا و تصلب شرایین دارد. به این طریق CASEY می تواند راه حل های مناسبی برای این موارد ارائه دهد.

از دیگر کاربردهای تشخیص پزشکی CBR جلوگیری از پیشرفت بیماری ایدز می باشد. سیستمی که بدین منظور به کار گرفته می شود AIA خوانده می شود که به منظور ارزیابی مقدماتی بیماری ایدز از آن استفاده می شود. این سیستم برای افرادی کاربرد دارد که در معرض خطر بیماری ایدز قرار دارند. در اینجا با استفاده از تکنیک CBR، از موارد قبلی جهت هدایت مراحل حل مسئله استفاده می شود. به این معنا که در این CBR ، موارد قبلی که شبیه به مورد جدید تحت بررسی هستند، فراخوانی گشته و سپس با دخالت افراد متخصص عملیات تطبیق و بررسی پیشگیری می شود.

ابزارهای CBR

بسیاری از ابزارهای (توسعه) تجاری CBR در حال حاضر موجود می باشند که توسط مؤسسات استنتاج توزیع می شوند. بسیاری از ابزارهای CBR می توانند روی کامپیوترهای پنتیوم تحت

سیستم عامل ویندوز اجرا شوند. برخی ابزارهای CBR دارای کاربرد های خاص هستند. برای مثال CASEY POINT در کاربرد های مربوط به به کمک راهنما بسیار مناسب است. شکل 5-4 ابزار دیگری را نشان می دهد که CBR Express نامیده می شود. این شکل نشان می دهد که این ابزار چه طور می تواند برای حل مشکلات یک چاپگر مورد استفاده قرار گیرد.

CBR در مقابل سیستم های سنتی

در فصل دوم دیدیم که بسیاری از سیستم های خبره، دانش را با استفاده از قوانین ذخیره می کنند. همانگونه که قبلاً دیدیم این سیستم های مبتنی بر قانون با ترکیب قوتنی در یک مکانیزم زنجیره ای که معمولاً زنجیره روبه عقب یا روبه جلو است، مسائل راه حل می نمایند. این روش به دانش حل مسئله و دامنه تکیه می کند. اما روش CBR به قلمرو دانش مسئله نیازی ندارد. چرا که CBR تنها باید تشخیص دهد که آیا مشکل مشابهی قبلاً حل شده است یا خیر. مزایای ذیل در استفاده از CBR وجود دارد:

شکل 5-4: حل مشکلات یک چاپگر با استفاده از CBR Express

◀ اکتساب دانش کاهش می یابد چون نمونه واضحی از عملیات در این زمینه لازم نمی باشد.

◀ قابلیت نگهداری دانش افزایش می یابد چون تغییرات حوزه دانش می تواند با موارد حل شده جدید به روز درآورده شود.

◀ قابلیت یادگیری به وسیله موارد جدید و بدون اضافه کردن قانون های جدید یا اصلاح قانون های موجود صورت می گیرد.

جدول 5-2 سیستم های خبره سنتی را با CBR مقایسه می کند.

هیوریستیک یا دانش سطحی

اکثر سیستم های خبره ، دانشی را که بر اساس ذهنیات پایه گذاری شده است به کار می برند. به چنین دانشی ، سطحی گفته می شود. به این خاطر که نمایش محدوده مسئله به به وسیله آنها سطحی است. برای مثال قوانین در نظر گرفته شده برای یک سیستم خبره تشخیص در اتومبیل را در نظر بگیرید:

اگر ماشین سوخت زیادی مصرف کند.

و موتور به موقع روشن نشود = بله

آنگاه شمع اتومبیل را عوض کنید = بله ،

قانون بالا دانش هیوریستیک سطحی را نشان می دهد که این فرضیه های ساده و عملی نتایج را سریعاً ایجاد می کنند . این مسأله در بخش بعدی بیشتر توضیح داده می شود.

جدول 5-2 مقایسه سیستم های فبره سنتی با CBR.

سیستم های فبره سنتی	CBR
محدود، مدل دامنه قابل درک است	مدل دامنه ضعیف است
قوانین <i>if-Then</i> ، قاب ها، منطق و غیره	موارد
زنجیره روبه عقب یا روبه جلوی قوانین، استفاده از روش های دیگر در قاب ها و غیره.	انتخاب موردی از موارد قبلی که تا اندازه ای شبیه به مشخصات مسئله جاری می باشند.
بسیار محدود	خوب، با اکتساب از موارد انجام می گیرد.
قابلیت یادگیری	

استدلال عمقی (علی العمق)

بسیاری از محققان (چاندرسکاران، تانزو جوزفسون در سال 1988) ارزش استدلال سیستم های خبره را از اولین اصلشان (اصول اولیه) دانسته اند. این مسئله استدلال عمقی نامیده می شود. برای مثال، یک سیستم خبره برای حفظ و نگهداری چمن های باغ ممکن اسن شامل قانون زیر باشد:

(1) اگر باران ببارد، آنگاه سبزه ها سریعتر رشد خواهند کرد.

این قانون در واقع مثال دیگری از استدلال سطحی می باشد. به این معنی که در استدلال سطحی مفهوم دقیقی بین علت و اثر وجود ندارد، چون زنجیره استنتاج ایجاد شده سطحی مفهوم دقیقی بین علت و اثر وجود ندارد، چون زنجیره استنتاج ایجاد شده است. این مثال یک نمونه هیورستیک است. یک قانون مانند آنچه ارائه شد ممکن است به شکل فشرده ای برای توضیحات پیچیده در برخی از پدیده های شیمیایی مورد استفاده قرار گیرد. مثلاً علت علمی این قانون با جزئیات بیشتر می تواند به صورت ذیل عنوان گردد:

اگر باران ببارد آنگاه سبزه‌ها سریعتر رشد خواهد کرد.

چون خاک مرطوب تر خواهد شد.

این قانون توسعه یافته می تواند به یک زنجیره عمیقی ، تبدیل شود . یکی از کاربردهای مدل عمقی دامنه، فراهم آوردن توضیحات با کیفیت بهتر می باشد.

یک مدل عمقی از دامنه می تواند توضیحات عمیق تری از رفتارهای سیستم ارائه دهد . چرا که مدلی واضح و کامل از زنجیره استدلال ایجاد خواهد شد.

چاندرسکاران، تانرو جزفسون در سال 1988 مدل های عمقی را برای گسترش توضیحات، پیرامون دانش دامنه های خاص با موفقیت بکاربرند. اگر چه، در بسیاری از دامنه ها ایجاد دانش عمقی و نیز معنادار شدن آن خودداری مشکلات است. برای مثال سیستم خبره ای توسط انجمن پزشکی انگلیس¹ ایجاد شده که علم اخلاق و قوانین ارتباط را بیان کرده ، چگونگی رفتار پزشکان با بیماران و سلوک پزشکی را روشن می سازد به وضوح مشخص است که ساخت یک مدل عمقی برای این نوع از دامنه مقدور نمی باشد. این سیستم را برای کمک به آموزش پرستاران ، پیراپزشکان و دانشجویان پزشکی بکار می برند. مثال دیگری در این رابطه southwick می باشد که یک سیستم خبره در زمینه حقوق است .

فصل پنجم مطالب فصل پنجم

استدلال مبتنی مورد¹ (CBR) با انتخاب یک مورد از مواردی که قبلاً در پایگاه اطلاعات ذخیره

شده اند کار می کند. به نحوی که این مورد بهترین و بیشترین شباهت را به خصوصیات مسأله

جاری تحت بررسی ، دارا باشد. این روش مشابه روشی است که افراد متخصص هنگام تلاش برای

1-(British Medical Association) BMI

حل انواع مسائل استدلال می کنند .

اکثر سیستم های خبره ، دانشی را که بر اساس ذهنیات پایه گذاری شده است به کار می برند. به چنین دانشی ، سطحی گفته می شود. به این خاطر که نمایش محدوده مسئله به به وسیله آنها سطحی است.

یک مدل عمقی از دامنه می تواند توضیحات عمیقتری از رفتارهای سیستم ارائه دهد. چرا که مدلی واضح و کامل از زنجیره استدلال ایجاد خواهد شد.

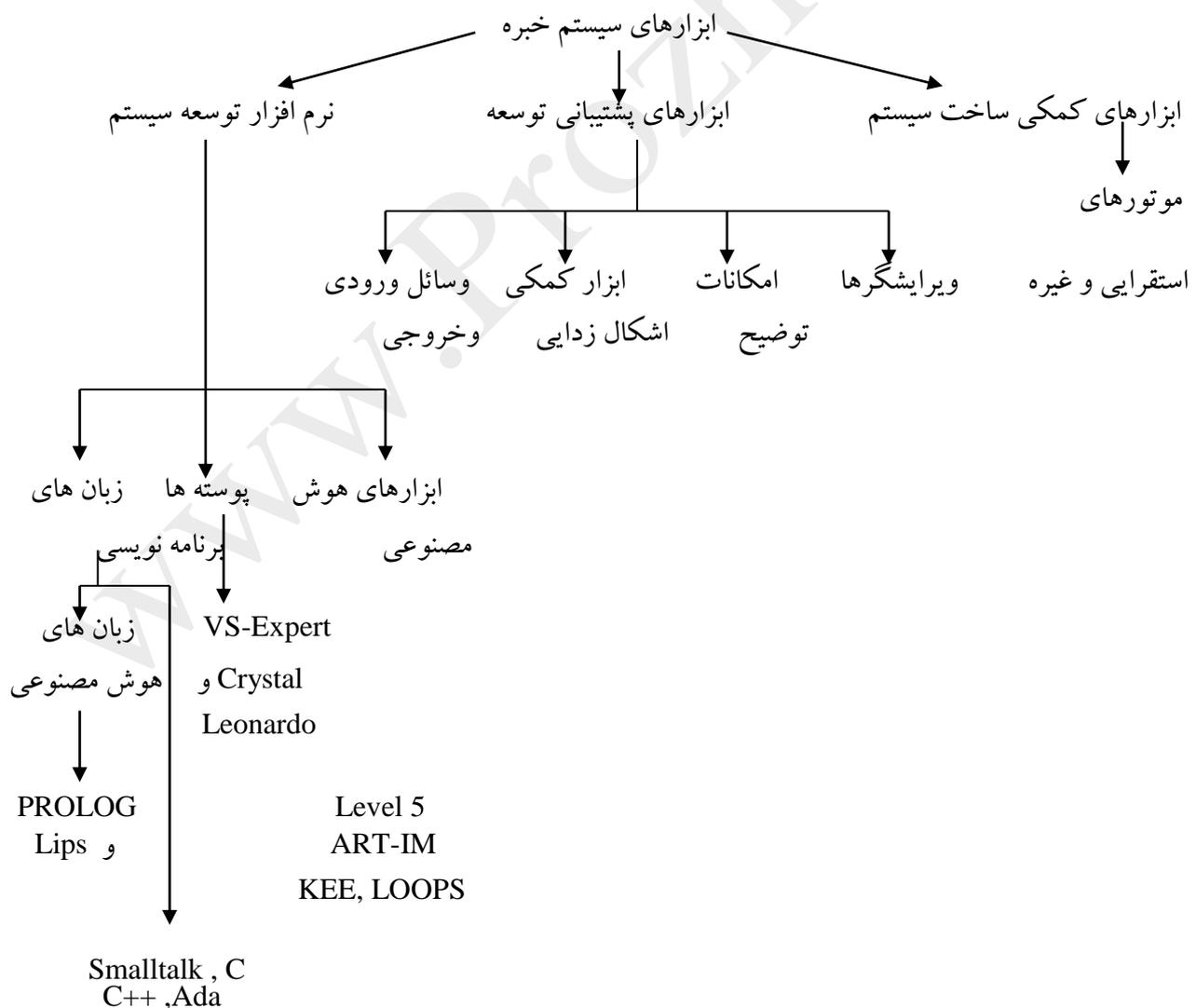
اصطلاح سیستم های ابزارهای خبره ، نرم افزاری را توصیف می کند. که برای ساخت یک سیستم خبره بکار می رود. این ابزارها از برنامه هایی تشکیل شده اند که در ساخت سیستم های خبره کاربرد دارند و شامل برنامه هایی هستند که به اکتساب دانش کمک می کنند. شکل 6-1 ارتباط میان ابزارهای عمومی سیستم های خبره را نشان می دهد.

ابزارهای توسعه سیستم های خبره

اصلی ترین ابزارهای نرم افزاری برای توسعه سیستم های خبره بصورت زیر دسته بندی می شوند:

◀ زبان های برنامه نویسی . ▶ پوسته های سیستم خبره . ▶ ابزارهای هوش مصنوعی .

شکل 6-1: سلسله مراتب ابزارهای توسعه سیستم خبره.



زبان های برنامه نویسی

- زبانهایی که برای توسعه سیستم های خبره بکار می روند به دو دسته تقسیم می شوند:
- 1- زبان های قراردادی .
 - 2- زبان های هوش مصنوعی .

زبان های قراردادی

زبان های قراردادی به نام زبان های مسئله گرا (مبتنی بر مسئله) نیز نامیده می شوند. مثل C , Ada , COBOL, Small talk . یکی از فوائد اصلی استفاده از زبان های قراردادی قابلیت دسترسی واسط ها به نرم افزارهای قراردادی مانند پایگاه های داده و صفحات گسترده است. اگرچه بسیاری از ابزارهای توسعه سیستم خبره ، هم اکنون از نظر تجاری در دسترس هستند و این امکانات را دارا می باشند، اما ممکن است که برای ساخت یک سیستم خبره از یک زبان برنامه نویسی قراردادی مثل C یا پاسکال حتی کوبول هم استفاده شود . همانطوریکه می توان از یک قیچی هم برای بریدن چمن استفاده کرد. این زبان ها برای ساخت سیستم های خبره مناسب نیستند. چرا که قدرت اداره ساختار دانش (که قبلاً بحث آن گذشت) را ندارند. برای مثال کوبول برای پردازش داده ها طراحی شده است نه برای نمایش و کنترل دانش. با این وجود از زبان هایی مانند زبان C که مزیت اصلی آن سرعت می باشد در طراحی سیستم خبره استفاده می شود.

زبان های هوش مصنوعی

زبان های هوش مصنوعی ، زبان های سمبلیک¹ نامیده می شوند. چون آنها برای کاربرد های هوش مصنوعی ایجاد شده اند. رایج ترین این زبان ها Lisp و PROLOG هستند. اولین گروه از سیستم های خبره اغلب از Lisp استفاده می کردند. ساخت سیستم های خبره با زبان های سطح

1.Symbol-manipulation languages

بالا بسیار وقت گیر است و اغلب چندین سال طول می کشد. چرا که سازنده مجبور است واسط
کاربر، پایگاه دانش و موتور استنتاج را با استفاده از ساختارهای موجود در زبان توسعه دهد. یک
مثال از چنین سیستم هایی DENDRAL می باشد که عبارت است از یک سیستم خبره تحلیل
شیمیایی، که برای استنتاج ساختارهای مولکولی ناشناخته به کار می رفت. PROSPECTOR مثالی
دیگر از سیستم خبره استخراج معدن می باشد که مانند یک مشاور فنی به زمین شناسان در
استخراج معدن کمک می کند. مثال دیگر Xcon می باشد که DEC از آن برای سفارشات
مشتریان استفاده می کند.

شکل 6-2 نمودار زمانی توسعه سیستم های خبره.

شکل 6-2 زمان تقریبی ایجاد هر یک از سیستم های فوق و نیز سیستم خبره LINK MAN که در
سال 1990 توسعه یافته است، را نشان می دهد.

این شکل به وضوح نشان می دهد که زمان توسعه سیستم خبره تا 30 سال قبل دارای پیشرفت چندانی نبوده اما در 10 سال گذشته این زمان تقلیل و بهبود یافته است، که علت آن توسعه ابزارها می باشد. عمومی ترین ابزارهایی که امروزه برای توسعه سیستم های خبره به کار می رود پوسته های سیستم خبره نام دارند.

پوسته های سیستم خبره

پوسته های یک خبره شروع آسان و راحت برای ساخت سیستم های خبره ایجاد می کنند چرا که کاربرد آنها ساده می باشد. در واقع آن سیستم های خبره ای هستند که خالی از قوانین می باشند. به این معنی که توسعه دهندگان می توانند روی ورودی پایگاه دانش متمرکز شوند بدون اینکه مجبور باشند چیزی مانند موتور استنتاج و یا واسط کاربر را بسازند. حتی خبرگانی که برنامه نویسی را نیز بلد نیستند می توانند به سرعت با پوسته ها آشنا شوند. همچنین بسیاری از پوسته های سیستم های خبره شامل امکاناتی است که اکتساب دانش را نیز راحت می سازند. چندین نمونه پوسته عموماً در دسترس هستند عبارتند از:

AM برای ویندوز (که قبلاً تحت عنوان Crystal شناخته می شد) و Leonardo و EXSYS که همه این ها روی سخت افزارهای کامپیوترهای شخصی تحت ویندوز کار می کنند. نمایی از محیط پیشرفته Leonardo در شکل 6-3 نمایش داده شده است.

پوسته ها در زمینه های تخصصی

هم اکنون ابزارهای متعددی برای توسعه زمینه های کاربردی خاص در دسترس می باشند. برای مثال پوسته هایی برای سیستم های تشخیصی، سیستم های وضعیتی، سیستم های برنامه ریزی، سیستم های کمکی راهنما و غیره وجود دارد. یک مثال از این ابزارهای خاص که در فصل قبلی

ذکر شد ، CBR Express نام دارد. یک CBR خصوصاً برای کاربرد در ابزارهای کمکی راهنما بسیار مناسب است. پوسته زمینه های خاص ، بسیار گرانتر از پوسته های معمولی می باشد.

شکل 6-3 : محیط پوسته سیستم خبره Leonardo.

پوسته های توسعه CBR در دسترس می باشند. هارمون (در سال 1992) به بازنگری در مورد چهار نمونه از این پوسته ها پرداخت. آن ها عبارتند از : ReMind برای سیستم های تشخیصی، CBR Express/ART-IM از محصولات استنتاج، Esteem از شرکت نرم افزاری Esteem و Casepower برای راه حل های استقرایی.

جعبه های ابزار هوش مصنوعی¹

ابزارهای دیگر برای سیستم ها خبره جعبه های ابزار هوش مصنوعی یا محیط ها² می باشند. این ابزارها ابزارهای پیشرفته ای هستند که نوعاً شامل ساختارهای مختلف کد، برای مجموعه ای از

1.AI toolkits
2.Environment

کارهای سیستم خبره می باشند. آنها از قوانین، قاب ها، برنامه نویسی شی گرا (OOP) و شبکه های معنایی یا منطق استفاده می کنند و ممکن است از زنجیره های روبه عقب یا روبه جلو CBR و یا تکنیک های متنوع توارث نیز استفاده نمایند. جعبه های ابزار هوش مصنوعی پیش رفته تر از پوسته ها (shell) هستند. بنابراین آنها میزان تولید و بهره‌وری را افزایش می دهند. گرچه به علت پیچیدگی، جعبه های ابزار هوش مصنوعی بیشتر از زبان های برنامه نویسی یا پوسته ها نیاز به مهارت دارند. نهایتاً می توان گفت که پوسته ها برای کاربردهای کوچک مناسب هستند و ابزارهای هوش مصنوعی نیز بیشتر برای کاربردهای بزرگ (همانند گروه های کاری Client/Server) مناسب هستند. دو نمونه از رایج ترین جعبه های ابزار هوش مصنوعی عبارتند از:

1- ART-IM که یک جعبه ابزار هوش مصنوعی بر پایه لیست و برای کامپیوترهای شخصی بوده که در محیط ویندوز قابل اجراست. ART-IM شامل 4 قسمت اصلی، قوانین (اصولاً برای دانش رویه ای بکار می رود) واقعیت ها، قاب ها، نظریه ها (برای دانش اعلانی) می باشد. ART برای کاربردهای پیچیده و مخصوصاً کاربردهای بلادرنگ مثل تفسیر هوشمند داده های دریافتی از حس گرها¹ در محیط کارخانه مناسب می باشد.

2- Level 5 Object که یک جعبه ابزار کامپیوترهای شخصی می باشد و برای کاربردهای Client/Server مناسب می باشد. همانند ART-IM Level 5 Object نیز مدل های مختلف دانش، شامل: قوانین، قاب ها و دامنه را حمایت می کند. همچنین شامل قابلیت های استدلال ارزنده ای می باشد. محیط توسعه شامل ویرایشگر قانون، ویرایشگر اشیاء و ویرایشگر نمایش برای طراحی

واسط کاربرد محیط ویندوز می باشد. ویرایشگر قانون جهت نگهداری قوانین و نمایش درختی دانش استفاده می شود. درخت دانش به معنای نمایش گرافیکی دانش بر مبنای درخت تصمیم گیری می باشد. ویرایشگر شی اجازه می دهد که توسعه دهنده کلاس های اشیاء را ایجاد کند و ویرایش نماید (شکل 4-6 را ببینید). امکانات ویرایشگر به توسعه دهنده اجازه ایجاد ویرایش موضوعات مختلف را می دهد. امکان کنترل جعبه ها، دکمه های رادویی، پنجره های متن، ناحیه ها و ابزارهای توسعه ویندوز همگی امکاناتی است که در اختیار توسعه دهنده قرار دارد.

ابزارهای کمکی سافت سیستم

ابزارهای کمکی ساخت سیستم، برنامه های مختلفی هستند که کلیه مراحل توسعه را شامل می شوند. آنها در عملیات مختلف مربوط به اخذ و نیزنمایش دانش خبره به کمک مهندس دانش می آیند. این ابزارها با امکانات مختلف ترسیم فلوچارت و غیره به طراحی سیستم خبره توسط مهندس دانش کمک می نمایند.

شکل 4-6؛ عملیات (روی کلاس ها در Object 5 Level).

شکل 6-5: ابزارهای کمکی ساخت سیستم فبره.

یک نمونه ابزار کمکی موجود جهت ساخت سیستم عبارت است از : Xpert Rule که Attar Software ایجاد شده است. شکل 6-5 نمایی از کمک های ساخت سیستم توسط این ابزار را نشان می دهد که به صورت درخت های تصمیم گیری می باشند.

ابزارهای اکتساب دانش

روش های مختلف اکتساب دانش بسیار گران و وقت گیر هستند. تلاش های زیادی صورت گرفته شده است تا ابزاری ایجاد شود که اکتساب دانش را آسان کند. یک موتور استقرایی رایج ترین وسیله ای است که قادر می باشد قوانین را از مثال های موجود استخراج نماید. هرچند که موتور استقرایی بخش ضروری یک سیستم خبره نمی باشد ، با این حال یک بخش مفید ضمیمه شده است که توسط بسیاری از پوسته های سیستم خبره مورد استفاده قرار می گیرد. موتور استقرایی

مجموعه ای از نمونه ها و مثال های مختلف را که یک حوزه خاص وجود دارد بررسی نموده ارتباطات بین مفاهیم مختلف را تحلیل می نماید. سپس موتور استقرایی قوانین که این مثال ها را به یکدیگر پیوند می دهد ایجاد می کند و آن ها در پایگاه دانش جای می دهد. مثال ها اغلب به شکل جدولی ترسیم می شوند. به عنوان مثال جدول 6-1 سه نمونه بدست آمده از یک خبره در بیمه عمر را نشان می دهد.

جدول 6-1: مثال هایی از یک فبیره بیمه عمر

مثال	سن	سیگاری	جنس	خطر
1	جوان	خیر	زن	پایین
2	پیر	بله	مرد	بالا
3	میانسال	خیر	مرد	پایین

جدول مثال ها، میزان خطر مرگ افراد را با توجه به فاکتورهای نشان داده شده مشخص می کند. موتور استقرایی می تواند قوانین را از مثال های موجود در جدول 6-1 ایجاد کند. البته باید توجه کرد که ضرورتاً یک تناظر یک به یک بین مثال های جدول و قانون های تولید شده وجود ندارد و قانون تولید شده می تواند وابسته به عوامل و فاکتورهای متفاوت دیگری نیز باشد. قوانین شکل 6-6 نشان می دهد که چطور تکنیک های استقرایی قوانین عمومی را از مثال های خاص تولید می کنند.

قانون 1	اگر شخص پیر باشد	قانون 2	اگر شخص میانسال
باشد			
	و جنس مرد باشد		و جنس مرد باشد

و سیگاری نباشد	و سیگاری باشد
آنگاه خطر پایین است.	آنگاه خطر بالاست.
	اگر جوان باشد
	و جنس زن باشد
	و سیگاری نباشد
	آنگاه خطر پایین است

شکل 6-6 : قوانین القا شده از مثال ها.

انتخاب ابزارهای توسعه

انتخاب ابزار مناسب برای ساخت سیستم های خیره مشکل تر از سیستم های قراردادی می باشد چراکه بسیاری از این ابزارها برای انواع خاصی از مسائل مناسب نیستند و یک ابزار انتخاب شده ممکن است پس از مدتی نیاز به تغییر داشته باشد.

علت این امر معمولاً به دلایل زیر است :

- ◀ سازنده از محدودیت های ابزار انتخاب شده ، آگاهی ندارد.
- ◀ بعضی از عوامل مهم ممکن است در طول تجزیه و تحلیل نادیده گرفته شوند.
- ◀ در صورت ضرورت ابزاری برای ساخت نمونه انتخاب می شود تا با تغییرات در نمونه ، سیستم توسعه یابد.

پرسش های زیادی هنگام انتخاب یک ابزار به وجود می آید که باید پاسخ داده شود از جمله اینکه:

- ◀ آیا ابزار انتخابی، خصوصیات پیشنهاد شده را با توجه به نیازهای مسئله و نوع کاربرد دارا

می باشد یا خیر؟

برای مثال قابلیت ایجاد عوامل قطعیت یا استنتاج بیزین وجود دارد یا خیر؟

◀ آیا ابزار امکانات تست، عیب یابی، اشکال زدایی و غیره را دارد یا خیر؟ شکل 6-7 اجزای یک محیط رانشان می دهد.



شکل 6-7 : محیط پشتیبانی توسعه

ارزیابی پوسته های سیستم فبره

معیارهای کلی انتخاب یک پوسته خاص برای توسعه سیستم خبره عبارتند از:

سهولت استفاده

ابزارهای انتخاب شده برای استفاده باید آسان باشد و نیز نمونه سازی را با سرعت انجام دهد.

توانایی تکنیکی

پوسته ها به طرق مختلف به نمایش دانش می پردازند و استراتژی های مختلفی را برای استنتاج بکار می برند. برای مثال VP-Expert کاملاً بر مبنای قوانین است و دیگر پوسته ها مثل Leonardo قاب ها یا شیء گرایی را پشتیبانی می کنند. قابلیت های تکنیکی باید توسط توسعه دهنده پس از درک کامل از ماهیت حوزه کاربرد شناسایی گردد.

محیط پشتیبانی توسعه

ساختاریک پوسته خبره که محیط را پشتیبانی می کند حداقل باید شامل یک ویرایشگر به منظور ایجاد پایگاه دانش، ابزار اشکال زدایی و ردیابی و نیز یک واسط کاربر دارای امکانات توضیح باشد.

امکانات واسط کاربر

پوسته ها باید شامل ابزاری برای توسعه کیفیت واسط کاربر باشند. بطوری که کاربر بتواند به هنگام اجزا از امکانات مختلفی از قبیل مسیریابی متن های به هم پیوسته و دستیابی به اطلاعات مختلف استفاده کند. همانند امکانات دسترسی به صفحات وب و یا استفاده از فرامتن.

واسط خارجی

بیشتر پروژه های سیستم خبره نیاز به درجه ای از ارتباط با خارج از محیط سیستم دارند. از قبیل: نیاز به فایل های صفحه گسترده، پایگاه های داده و یا حتی روال های ++C. برای مثال در یک سیستم خبره پزشکی ممکن است به اطلاعاتی در رابطه با بیمار نیاز داشته باشیم که این اطلاعات در یک پایگاه داده ای ذخیره شده باشد.

مجازها و پشتیبانی فروشنده

درواقع این پارامتر چگونگی توسعه یک محصول را از لحاظ کاربردی بررسی می نماید. مثلاً برنامه روی نرم افزارها و سخت افزارهای مختلف بدون هزینه های اضافی (افزایش قیمت) قابلیت توسعه

و توزیع داشته باشد (قابلیت حمل). به عنوان مثال یک سیستم خبره توسعه یافته بر روی یک کامپیوتر پنتیوم تحت ویندوز 98 ممکن است روی یک کامپیوتر 486 تحت DOS اجرا نشود. درخت شکل 6-8 نتایج ارزیابی های بیان شده و ارتباط هر کدام با دیگری را نشان می دهد.

نتایج

در این فصل مجموعه ابزارهای توسعه سیستم خبره مورد بررسی قرار گرفت. پوسته های سیستم خبره یک وسیله سریع و کم هزینه را برای آشنایی با تکنولوژی مهیا می سازد. اگرچه برای محدوده های خاصی در سیستم های پیچیده استفاده از پوسته های سیستم خبره میسر نمی باشد. اما با تمام این محدودیت ها، پوسته ها بطور زیادی مورد استفاده قرار می گیرند.

در بعضی موارد آنها در ابتدای چرخه یعنی در مراحل اولیه کار، به منظور ایجاد درک بهتر از نیازمندی های مسأله استفاده می شوند و در سطوح بالاتر کنار گذاشته می شوند.

قانون زیر به اختصار چگونگی انتخاب یک ابزار برای توسعه سیستم خبره را بیان می کند:

اگر می توانید پوسته ها را بکاربرید. در جایی که مجبورید از یک جعبه ابزار استفاده کنید و وقتی که چاره ای ندارید، یک زبان هوش مصنوعی را بکارگیرید.

خلاصه مطالب فصل ششم

ابزارهای توسعه سیستم های خبره

- ◀ زبان های برنامه نویسی .
- ◀ ابزارهای هوش مصنوعی .
- ◀ پوسته های سیستم خبره.

زبان های برنامه نویسی

- 1- زبان های قراردادی .
- 2- زبان های هوش مصنوعی .

زبان های قراردادی به نام زبان های مسئله گرا (مبتنی بر مسئله) نیز نامیده می شوند.

زبان های هوش مصنوعی ، زبان های سمبلیک نامیده می شوند.

پوسته های یک خبره شروع آسان و راحت برای ساخت سیستم های خبره ایجاد می کنند چرا که

کاربرد آنها ساده می باشد. در واقع آن سیستم های خبره ای هستند که خالی از قوانین می باشند.

هم اکنون ابزارهای متعددی برای توسعه زمینه های کاربردی خاص در دسترس می باشند.

ابزارهای دیگر برای سیستم ها خبره جعبه های ابزار هوش مصنوعی یا محیط ها² می باشند. این

ابزارها ابزارهای پیشرفته ای هستند که نوعاً شامل ساختارهای مختلف کد، برای مجموعه ای از

کارهای سیستم خبره می باشند.

ART-IM-1 که یک جعبه ابزار هوش مصنوعی بر پایه لیست و برای کامپیوترهای شخصی بوده

که در محیط ویندوز قابل اجراست.

ویرایشگر قانون جهت نگهداری قوانین و نمایش درختی دانش استفاده می شود. درخت دانش به

معنای نمایش گرافیکی دانش بر مبنای درخت تصمیم گیری می باشد.

روش های مختلف اکتساب دانش بسیار گران و وقت گیر هستند. تلاش های زیادی صورت گرفته

شده است تا ابزاری ایجاد شود که اکتساب دانش را آسان کند. یک موتور استقرایی رایج ترین

وسیله ای است که قادر می باشد قوانین را از مثال های موجود استخراج نماید.

تأثیر متقابل انسان و کامپیوتر

تأثیر متقابل انسان و کامپیوتر¹ (HCI) موضوع بسیار مهمی در طراحی سیستم های خبره محسوب می شود. همانطوریکه برای سایر سیستم های رایج نیز اینگونه است. در این باره باید بحث کرد، چرا که تأثیر این مسأله در طراحی سیستم های خبره بسیار زیاد است. اکلی در سال 1980 اظهار داشت جای تردید نیست که خروجی یک سیستم خبره ممکن است درست باشد یا نباشد. به این معنی که کاربر می تواند سیستم را مورد بررسی قرار دهد و به خروجی نیز شک کند و سؤالاتی را طرح نماید.

سیستم های خبره موفق باید به جانب توسعه HCI اطمینان نمایند. چنین پیش بینی می شود که در آینده توضیحات (برای خروجی) کیفیت بهتری پیدا کنند و ساختار ورودی و خروجی نیز بهبود یابد.

نیازمندی های واسط کاربر در سیستم های خبره

بسیاری از طرح های HCI وجود در سیستم های خبره، همان هایی هستند که در سایر سیستم ها نیز وجود دارند. طبیعت یک فرآیند تأثیر متقابل در یک سیستم خبره به عنوان یک نیاز حیاتی، موجب پیدایش یک سری خواسته های متفاوت و اضافی در واسط ها می شود (هندلر و لویس در سال 1992). برخلاف برنامه های معمولی یک سیستم خبره تنها وسیله ای برای انجام و تکمیل یک پردازش نیست، بلکه بیشتر به صورت نمایش آن پردازش می باشد.

علاوه بر این بسیاری از این پردازش ها دارای مفاهیمی هستند که در آنها قضاوت هایی توسط سیستم انجام می گیرد که نتایجی همچون دنیای واقعی دارند. داوری هایی که غالباً استنتاج هایی

1. Human-Computer interaction

همراه با عدم قطعیت می باشند . واسطه کاربر تنها برای نمایش نتایج نمی باشد . بلکه باید نشان دهد که چگونه این نتیجه حاصل شده است. برخی از واسطه های کاربر برای مهندس دانش و کاربر موارد استفاده فراوانی دارند.

واسطه مهندس دانش و واسطه کاربر

تفاوت آشکاری بین واسطه مهندس دانش و واسطه کاربر وجود دارد. اولی دسترسی به ابزارهای ساخت سیستم های خبره مانند ویرایشگر، ابزار تست، امکانات اشکال زدایی و امکانات تفسیر به منظور توانمند کردن مهندس دانش برای بررسی صحت برنامه و غیره را مهیا می کند. اما دومی دسترسی به عملیات سیستم خبره را مهیا می کند و کاربر را قادر می سازد با سؤالاتی که می پرسد و جواب هایی که سیستم به آنها می دهد، عملکرد سیستم نمایش داده شود. مانند ابزار توضیح و تفسیر و امکانات کمکی بر خط (Online) و غیره . روشن است که متخصص نیاز دارد در طول فاز توسعه سیستم خبره برای تأییدی درستی و صحت عملکرد برنامه به واسطه مهندس دانش دسترسی داشته باشد. علاوه بر این کاربر نیز نیازمند به تست سیستم خبره به منظور پذیرش پاسخ های آن می باشد. ارتباط بین افراد در این حوزه که شامل بر متخصص ، کاربر و مهندس دانش می باشند با هم و نیز با سیستم خبره در شکل 1-7 نشان داده شده است.

شکل 1-7 : رابطه میان افراد مختلف در سیستم های خبره.

وسائل ورودی و خروجی برای سیستم های فبره

برای برقراری ارتباط بین کاربر و سیستم خبره وسائل ورودی و خروجی بسیاری وجود دارد. متداولترین وسائل ورودی عبارتند از: ماوس، صفحه کلید، قلم نوری، صفحات حساس به لمس و انواع ورودی صوتی (مانند میکروفون). خروجی ها نیز عبارتند از: صفحه نمایش، چاپگر، پلاترو... عملکرد تقابلی یک سیستم خبره معمولاً شامل طرح تعدادی پرسش می باشد که برای آن باید از یک یا چند ابزار ورودی استفاده نمود و معمولاً متعاقب آن از یک یا چند وسیله خروجی نیز استفاده می شود.

پیغام هایی که سیستم می دهد ممکن است به صورت متن و یا خروجی گرافیکی بر اساس دامنه طبیعی سیستم خبره باشد.

برای مثال ممکن است نشان دادن شکل یک ماشین توسط تصویر برای کاربر، قابل فهم تر باشد در حالی که فرم خروجی به صورت متن برای سیستم های خبره در زمینه حقوق مناسب تر است.

روش های مناسبیت تأثیر انسان در سیستم های فبره

انواع مختلف واسطه های کاربر برای توسعه دهندگان و کاربران سیستم های خبره وجود دارد. بسیاری از این روش ها برگرفته از سیستم های نرم افزاری رایج می باشند. متداولترین روش ها در زیر آمده است.

به سیستم خبره (Housing Benefit) خوش آمدید.

1- نام شما چیست؟ فردفیلیت سون.

2- آیا اکنون شاغل هستید؟ خیر.

3- تا کنون چقدر پول پس انداز کرده اید؟ 7500

شکل 2-7 نمونه سوالات مطرح شده در سیستم فبره Housing Benefit

سؤال 2: آیا اکنون شاغل هستید؟ (لطفاً بله یا خیر را انتخاب کنید)

بله

خیر

شکل 7-3: یک پرسش بصورت منو.

استفاده مستقیم از صفحه کلید

این روش به این صورت است که کاربر به یک پرسش با تایپ جواب توسط صفحه کلید پاسخ می دهد، که یکی از قدیمی ترین روش هاست. یک مثال از این نوع عملکرد در شکل 7-2 نشان داده شده است. این مثال از سیستم خبره housing benefit آورده شده است.

(توجه: سوالات سیستم خبره به فرم نرمال نوشته می شود و پاسخ های کاربر بصورت ایتالیک نمایش داده می شود).

واسطه های کاربر بصورت منو

اگر سوال هایی که در شکل 7-2 آمده است بطور دقیق مورد توجه قرار گیرند می بینیم که روش ورودی مستقیم با صفحه کلید تنها برای پاسخ به سوالات 1 و 3 است. چون شمارش همه انتخاب های ممکن برای این دو سوال غیر ممکن است. اما برای سوال دو بهتر است شیوه ارتباط واسطه بصورت منو مورد استفاده قرار گیرد. چنین سیستمی کاربر را مجاز می سازد تا از میان چندین موردی که بر روی صفحه نمایش است، یکی را انتخاب کند و کاربر نیز توسط ماوس یا صفحه کلید گزینه مورد نظر خود را انتخاب می کند. برای مثال در شکل 7-3 نمونه ای از این شیوه نشان داده شده است. این شیوه معمولاً نسبت به شیوه ورودی مستقیم به صفحه کلید ارجحیت دارد چون

موجب صرفه جویی در وقت می شود و اشتباهات تایپی را کاهش داده و کاربر را مجاب می کند تا انتخاب خود را از یک منو که در دسترس است انتخاب کند.

نکاتی در مورد طراحی منوی واسط

- ◀ سعی کنید مواردی که ارتباط منطقی باهم دارند در یک منو گروه بندی کنید.
- ◀ سعی کنید منوها را بر اساس کاربرد، توالی و اهمیت طبقه بندی کنید.
- ◀ پاسخ های مورد نظر را ذکر نمایید و آن را با انتخاب ها ربط دهید.
- ◀ عنوان منو را بر اساس عملکردش انتخاب کنید.
- ◀ کاربر را در مورد سطوح منو، اشتباهات موجود و غیره آگاه سازید.
- ◀ برای کاربر راه خروج و سیر عبور قرار دهید.
- ◀ پاسخ ها را تضمین کنید. برای مثال اگر انتخاب ها از شماره 1 تا 7 هستند و صفر راه خروج است. اطمینان حاصل کنید که هر کلید دیگری اشتباه بوده و کار نمی کند.

زبان طبیعی

واسط زبان طبیعی یک روش ارتباطی است که کاربر توسط عبارات یک زبان طبیعی مثل زبان انگلیسی تقاضای خود را وارد می کند.

فرم ها

واسط هایی هستند که کاربر را قادر می سازند که اطلاعات را با تایپ کردن و یا شیوه های دیگر پیش بینی شده روی صفحه نمایش وارد نماید.

رابط تصویری (آیکون)

درواسط های سمبلیک ، عبارات مورد نظر کاربر و پاسخ های سیستم به جای کلمات بصورت سمبل و یا اشکال تصویری نمایش داده می شود. این نوع روش واسط در محیط های تحت ویندوز بسیار متداول است .

دستکاری مستقیم

اشاره به شیوه ای از واسط ها دارد که در آن کاربر از طریق فشار دادن کلید و حرکت دادن یک وسیله نشانگر مثل یک ماوس ، نمایش گرافیکی اطلاعات را دستکاری می کند.

میزان رضایت کاربر

موفقیت یک سیستم خبره بستگی به این دارد که کاربر تا چه اندازه در کار با آن سیستم احساس راحتی نماید . بنابراین بسیار تلاش شده است که واسط های سیستم خبره را توسعه دهند. نکات کلیدی در طراحی واسط های کاربر در زیر جمع آوری شده است:

◀ طراحی واسط ، باید بر روی کاربر متمرکز باشد. به این معنی که یک واسط باید با کاربران ارتباط متقابل داشته و در ارتباط با نیازهای آنان منطقی و سازگار باشد. همچنین شامل امکاناتی برای کمک به کاربران در مورد سیستم بوده، اشتباهات آنان را نشان داده و حل نماید.

◀ طراحی یک واسط بصورت استعاره اغلب راه خوبی برای کمک به کاربران یک سیستم می باشد. فرهنگ وبستر در تعریف لغت استعاره می گوید “نوعی از گفتار که در آن یک لغت و یا عبارت منسوب به عملی می شود که موجب ارتباط و یا شباهت آن با لغتی دیگر میگردد.” استعارات به کاربران در درک یک مفهوم جدید کمک می کند.

◀ انتخاب یک روش واسط مناسب . مثلاً شیوه واسط WIMP¹، اگر کاربر به یک واسط

گرافیکی

1.Windows Icons Mouse and Tointers

مانند ویندوز دسترسی داشته باشد، مناسب به نظر می رسد. سیستم های منوی نیز برای کاربران عادی مناسب است چراکه آگاهی آنها به نسبت کم می باشد. همچنین وقتی تعداد گزینه ها بسیار زیاد باشد کاربران خسته خواهند شد. واسط های زبان طبیعی برای کاربران متخصص و با تجربه تنظیم شده است چون دستورات پیچیده می توانند ایجاد شوند و همچنین سریعتر از منوها عمل می کنند. نمایش گرافیکی اطلاعات باید زمانی استفاده شود که نیاز به این داشته باشیم که اطلاعات را به صورت تخمینی و نموداری عرضه می کنیم. نمایش دیجیتالی نیز تنها هنگامی باید استفاده شود که زمان مورد نیاز است.

◀ کمک های برخط (on-line) چندی است که در نرم افزارها مورد استفاده قرار می گیرد و استفاده از آن لازم نیز می باشد که این مورد برای کاربران سیستم های خبره نیز صدق می کند. کاربران باید بدانند که چگونه امور مختلفی را انجام دهند. مثل تغییر پاسخ یک پرسش و یا داشتن اطلاعاتی در مورد حوزه عملکرد.

◀ طراحی مناسب واسط باید اصلاح اشتباهات را نیز شامل شود. پیغام های اشتباه نباید توهین آمیز بوده و موجب سرزنش کاربر شود. به کاربران باید پیشنهاداتی برای رفع اشتباه و کمک در کار با سیستم ارائه شود.

◀ رنگ ها باید به دقت بکاربرده شوند تا موجب سردرگمی کاربر نشود. ترکیب رنگ ها باید ثابت و مناسب باشد. برای مثال رنگ زمینه برای پیغام اشتباه در ویندوز قرمز می باشد، که این رنگ باید در طراحی به طور ثابت استفاده شود.

اصول طراحی آیکون

نکات زیر باید در طراحی واسط مبتنی بر آیکون در نظر گرفته شود.

- ◀ نحوه عرضه آیکون به کاربران آزمایش شود.
- ◀ در صورت امکان از استعارات در حوزه کاربر استفاده شود.
- ◀ تاجایی که ممکن است، سمبل، واقعی انتخاب شود.
- ◀ از سمبلی استفاده شود که مفهوم آن از قبل مشخص شده باشد.
- ◀ هنگام نشان دادن دستورات از نمایش واقعی استفاده شود.
- ◀ برای وضوح بهتر، سمبل ها به صورت واضح و مشخص کنارهم قرار داده شوند.

امکانات تشریح

در فصل دوم دیده شد که سیستم های خبره توانایی تشریح فرآیندهای استدلال خود را دارند. امکانات تشریح در متقاعد ساختن کاربران نکته سنج در تأیید قابلیت های یک سیستم خبره بسیار مؤثر است. بعضی از محققین معتقدند که تسهیلات تشریحی باعث تفهیم بهتر کاربران مبتدی از دامنه کاری خود می شود. عدم موفقیت در تغییر سیستم خبره پزشکی MYCIN جهت تبدیل آن به یک برنامه آموزشی برای دانشجویان، در همین نکته نهفته است.

نیاز به امکانات تشریح

توانایی تشریح فرآیندهای استدلال برای سیستم های خبره بعنوان یک مشخصه کلیدی محسوب می شود. امکانات تشریح، کاربر را از چگونگی رفتار سیستم آگاه می کند. تحقیقات نشان می دهد که کاربران سیستم های خبره از امکانات تشریح استفاده بهینه می نمایند. این امر به این خاطر اهمیت دارد که مشاوره با یک انسان متخصص نیز اغلب نیازمند به برخی تشریحات می باشد. بسیاری افراد پاسخ های ارائه شده از یک متخصص را نیز بدون چنین توجیهاتی قبول نمی کند.

بعنوان مثال یک بیمار چاق که به او توصیه شده تحت رژیم درمانی ویژه ای قرار گیرد، ممکن است سؤال کند: علت این توصیه چیست؟

دانش عمقی و سطحی

عدم توانایی در تهیه دانش قابل توجیه به این دلیل اتفاق می افتد که مدلی از دانش استفاده شده توسط خبرگان برای استدلال وجود ندارد. این مسأله دانش عمقی نامیده شده و متخصصین به چنین دانشی دسترسی دارند. هرچند، وقتی یک متخصص با مشکلی روبه رو می شود سعی می کند با استفاده از روش هیوریستیک یا همان راه های تجربی با مشکلات مواجه شود که این حالت دانش سطحی نامیده می شود اما متخصصین برای حل سریع مسائل به حافظه و محفوظات خود نیز رجوع می نمایند. برای مثال یک پرستار، چشمان مریضی را در مورد تأثیر دارو چک می کند تا مقدار مصرف آن را تعیین کند. این دانش ساده از دانش فیزیولوژیکی انسان در لایه های عمیق تر ناشی می شود و همچنین دانش است که اغلب در سیستم های خبره دیده می شود چرا که می توان به طور بسیار آسان و راحت از آن برای حل مسائل استفاده کرد. یک نمایش عمقی اغلب برای بهبود امکانات تشریحی لازم است.

سیستم XPLAIN

سوارت اوت نیز در سال 1986 متوجه شد که سیستم های مبتنی بر قوانین قادر به اثبات آن چه انجام می دهند نمی باشند. چون دانش لازم برای ایجاد توجیهات در سیستم وجود ندارد. سیستم XPLAIN او یکی از اولین مواردی بود که به استفاده از تکنیک های برنامه نویسی اتوماتیک برای اثبات طرح ها در پس ساختار تشریحات سیستم خبره تأکید داشت. به هر حال استفاده از این رویه و متد نیز برای اجرا با مشکلات زیادی توأم بود.

دانش استراتژی

مجدداً پایگاه دانش گیاهان خانگی را در نظر بگیرید. اگر قانون 4 اجرا نمی شد. آنگاه موتور استنتاج سعی در اثبات قانون بعدی می کرد که قانون 5 بود. دستوری که موتور استنتاج سعی در انتخاب بهترین گیاه خانگی دارد کاملاً براساس قواعد است. بنابراین دانش توضیح دهنده دستورات حل مسئله، قابل رویت نیست.

کلنسی در سال 1983 این مفهوم را ثابت کرد. وقتی که او متوجه شد GUIDON قادر به توضیح استراتژی سیستم خودش نمی باشد به مانند سیستم گیاهان خانگی او ثابت کرد که استراتژی جستجو در قوانین مطلق می باشد.

کلنسی می گوید: در طی برنامه GUIDON دریافتیم که روش تشخیص و درک قواعد و قوانین توسط خبره به وضوح نمایش داده نشده است و GUIDON نمی تواند قوانین را بیان نماید. چون MYCIN مشخص نمی کند که چگونه مفاهیم در یک قانون باهم جور درمی آیند. GUIDON بطور واضح و کامل نحوه حل مشکلات و مسائل توسط MYCIN را بر نمی شمارد چون ساختار فضای جستجو و استراتژی عملکرد آن صرفاً در حوزه مفاهیم قواعد واضح و قابل اجرا می باشد. بنابراین تغییر یک سیستم مبتنی بر دانش به یک سیستم کامپیوتری کمک برنامه های آموزشی نیاز به یافتن جزئیاتی در مورد پایگاه قوانین و تغییرات و تحولات زیادی دارد. سیستمی را که کلنسی طراحی کرد، تحت عنوان NEOMYCIN نام گرفت (1983). این سیستم توانایی های MYCIN را در تشریح استراتژی افزایش داد. دانش استراتژی، دانشی در رابطه با چگونگی دسترسی به قوانین مختلف برای حل یک مسأله می باشد. جدول 7-1 انواع مختلف دانش را که برای تشریح لازم است بصورت خلاصه عرضه می کند.

در NEOMYCIN دانش استراتژیک بشکل قوانین پیشرفته درمی آید که درجایی به غیراز دامنه دانش طبقه بندی شده اند.

جدول 7-1: دانش مورد نیاز برای توضیحات

دانش مورد نیاز	هدف و تأثیر تشریح	روش های ممکن در سیستم های فبره مبتنی بر قوانین
حل مسئله	نمایش زنجیره ای استدلال انجام شده توسط سیستم، با ایجاد یک ردیابی از روش حل مسئله، که در نتیجه مسئله باراه حل پیوند می خورد.	همیشه در دسترس است. بدون اینکه تلاش اضافی صورت گیرد.
توجیه درستی	برای تأیید دانش موجود در سیستم خبره است. نتیجه آن این است که کاربران به صحت پایگاه دانش اطمینان بیشتری می نمایند.	اگر یک مدل عمقی نیاز باشد می تواند مورد استفاده قرار گیرد.
استراتژی	نشان می دهد که چطور یک خط استدلال از خطوط دیگر ناشی می گردد و نتیجتاً به کاربران اطلاعاتی رادر زمینه روش های بکاررفته برای (حداقل) جستجوی یک راه حل ارائه می کند.	می توان با استفاده از قوانین فراگیر ¹ مسائل را حل نمود

متن فشرده²

متن فشرده، در برخی سیستم های امروزی عملکرد موفقی داشته است و همانطور که از نامش

-
1. Meta-Rule
 2. Canned Text

پیدا است از متونی که قبلاً آماده شده است برای ارائه تشریح استفاده می کند. یکی از عمده ترین موارد استفاده از متن فشرده ارائه پیغام های خطا در کامپیوتر می باشد. با پیش بینی پرسش هایی که کاربران ممکن است پرسند، سیستم پاسخ های احتمالی را که به صورت عبارات یک زبان طبیعی طراحی می کند بطوریکه دسترسی به آنها از روش ردیابی قوانین سریعتر می باشد.

در عین حال طراح سیستم، باید به منظور طراحی پاسخ های مناسب، همه سؤالات احتمالی کاربران را پیش بینی نماید. متن فشرده بر خلاف ردیابی قوانین به صورت بسیار وابسته و غیر مستقل عمل می کند که این مسأله موجب می شود استفاده از آن در سیستم های دیگر بسیار مشکل شود. همانگونه که اسپارک جونز (1984) مطرح کرد، پیش بینی متن فشرده موجب انحراف کاربر در تخمین قابلیت های سیستم می شود. چون کاربران ممکن است فکر کنند که بازیان طبیعی و عادی با سیستم در حال گفتگو هستند.

موارد استفاده فرا رسانه ای

در سال های اخیر توان و انرژی زیادی صرف ترکیب سیستم های خبره و سیستم های فرارسانه ای شده است که بیشتر مورد قبول کاربران باشد. سیستم فرارسانه ای مجموعه ای از متن، شکل و صداست که از مکانیزم نشانه گذاری استفاده می کند. سیستم فرامتن به صورت ویژه ای از متن استفاده می کند. کاربرد این سیستم را می تواند در صفحات وب موجود در اینترنت مشاهده کرد. سیستم فرارسانه ای ابزاری برای مدیریت اطلاعات است که متن، شکل و صدا را به روش خاصی باهم پیوند می دهد. استفاده از این روش بکاربر اجازه می دهد در یک سیستم به طور غیر خطی به کاوش پردازد نرم افزارهایی مثل Microsoft Internet Explorer Netscape Navigator به

کاربر اجازه می دهند تا با کلیک کردن بر روی یک کلمه یا یک تصویر به اطلاعات بیشتری دست یابد. سیستم فرا متنی به صورت موفق در بسیاری از سیستم های خبره استفاده می شود.

شکل 5-7: مثالی از کاربرد فرامتن در برنامه Knowledgepro

شکل 5-7 قطعه ای از یک سیستم مشاوره آماری را نشان می دهد که از فرامتن برای توضیحات اضافی در مورد آزمون های آماری استفاده می کند. این سیستم با استفاده از برنامه Knowledgepro در محیط ویندوز نوشته شده است. در شکل 5-7 در بالای پنجره زیر عبارت ضریب مطابقت kendalls خط کشیده شده است این عبارت نشان دهنده وجود یک فرامتن می باشد که کاربر می تواند با کلیک بر روی آن به توضیحات اضافی دست پیدا کند. پایین شکل 5-7 متن نهفته در مورد عبارت بالا را نشان می دهد که شامل توضیحات لازم می باشد. توجه داشته باشید عباراتی که زیر آن در بالا خط کشیده شده است در عنوان پنجره پایینی آمده است.

سیستم فرامتنی به کاربران اجازه می دهد تا باتفکیک میزان تخصص خود در استفاده از سیستم در سطوح مختلف از سیستم فرامتن استفاده کنند.

واسط های گرافیکی

برخی از سیستم های خبره پیشرفته وجود دارند که چند شکل از واسط های گرافیکی را دارا هستند. اینها عمدتاً برای کاربردهای ویژه ای هستند که نمیتوان آنها را تعمیم داد.

STEAMER (هولان ، هاجنیز، ویتزمن در سال 1984) معروف ترین سیستم مبتنی بر گرافیک است که برای شبیه سازی فشار بخار در نیروی دریایی ایالات متحده آمریکا بکار می رود.

سیستم اولیه شامل مدل ریاضی برای فشار بخار بود که STEAMER یک واسط تصویری شبیه سازی شده به آن اضافه کرد. STEAMER چندین تصویر از سیستم فراهم می نماید که با استفاده از آن می توان سیستم را در سطوح مختلف کاری بررسی نمود. به هر حال STEAMER با هدف جایگزینی انسان متخصص بنا نشده است. برعکس هدف از طراحی آن ایجاد یک سیستم خبره در حوزه خاص بوده که به انسان در درک یک دامنه پیچیده کمک کند.

نتیجه گیری

موفقیت یک سیستم خبره اغلب به چگونگی کیفیت واسط کاربر بستگی دارد. به نظر نمی رسد که مشکلات موجود در طراحی واسط کاربر برای سیستم های خبره به دلیل ملاحظات تجاری باشد. بلکه مشکلات ، در مورد چگونگی تطابق نیاز های کاربران و واسط ها می باشند. (استلزنر- ویلیام 1992) به این معنا که کاربران سیستم های خبره نه تنها از نظر تخصص و آگاهی در سطوح مختلفی قرار دارند ، بلکه نیازهای آنان نیز دارای تفاوت هایی می باشد. بسیاری خواهان

وجود توضیحات در سیستم خبره اند. با این حال همانطوریکه در این فصل بررسی شد، چنین تسهیلاتی گاهی محدودند و بدون کمک طراحان حرفه ای دستیابی به آن امکان پذیر نمی باشد.

خلاصه مطالب فصل هفتم

بسیاری از طرح های HCI کوجود در سیستم های خبره، همان هایی هستند که در سایر سیستم ها نیز وجود دارند. طبیعت یک فرآیند تأثیر متقابل در یک سیستم خبره به عنوان یک نیاز حیاتی، موجب پیدایش یک سری خواسته های متفاوت و اضافی در واسط ها می شود (هندلر و لویس در سال 1992). برخلاف برنامه های معمولی یک سیستم خبره تنها وسیله ای برای انجام و تکمیل یک پردازش نیست، بلکه بیشتر به صورت نمایش آن پردازش می باشد.

علاوه بر این بسیاری از این پردازش ها دارای مفاهیمی هستند که در آنها قضاوت هایی توسط سیستم انجام می گیرد که نتایجی همچون دنیای واقعی دارند.

تفاوت آشکاری بین واسط مهندس دانش و واسط کاربر وجود دارد. اولی دسترسی به ابزارهای ساخت سیستم های خبره مانند ویرایشگر، ابزار تست، امکانات اشکال زدایی و امکانات تفسیر به منظور توانمند کردن مهندس دانش برای بررسی صحت برنامه و غیره را مهیا می کند. اما دومی دسترسی به عملیات سیستم خبره را مهیا می کند و کاربر را قادر می سازد با سؤالاتی که می پرسد و جواب هایی که سیستم به آنها می دهد، عملکرد سیستم نمایش داده شود.

برای برقراری ارتباط بین کاربر و سیستم خبره وسائل ورودی و خروجی بسیاری وجود دارد. متداولترین وسائل ورودی عبارتند از: ماوس، صفحه کلید، قلم نوری، صفحات حساس به لمس و انواع ورودی صوتی (مانند میکروفون).

انواع مختلف واسطه های کاربر برای توسعه دهندگان و کاربران سیستم های خبره وجود دارد. بسیاری از این روش ها برگرفته از سیستم های نرم افزاری رایج می باشند.

موفقیت یک سیستم خبره بستگی به این دارد که کاربرتا چه اندازه در کاربا آن سیستم احساس راحتی نماید. بنابراین بسیار تلاش شده است که واسطه های سیستم خبره را توسعه دهند.

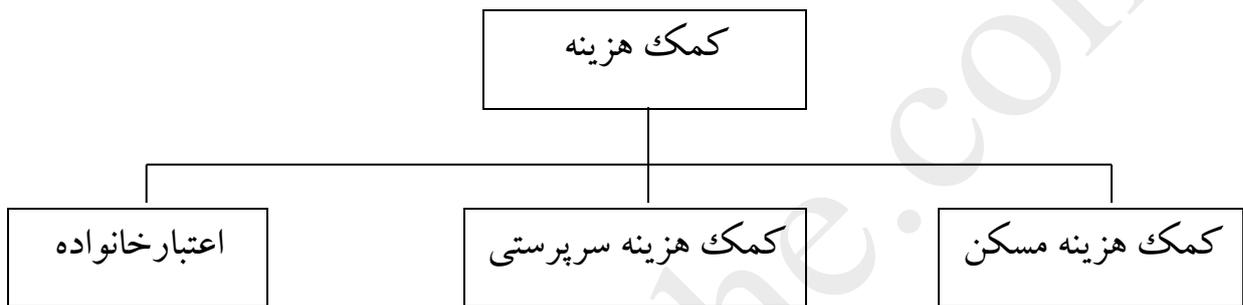
نکات زیر باید در طراحی واسطه مبتنی بر آیکون در نظر گرفته شود.

- ◀ نحوه عرضه آیکون به کاربران آزمایش شود.
- ◀ در صورت امکان از استعارات در حوزه کاربر استفاده شود .
- ◀ تاجایی که ممکن است، سمبل، واقعی انتخاب شود.
- ◀ از سمبلی استفاده شود که مفهوم آن از قبل مشخص شده باشد.
- ◀ هنگام نشان دادن دستورات از نمایش واقعی استفاده شود.
- ◀ برای وضوح بهتر، سمبل ها به صورت واضح و مشخص کنارهم قرار داده شوند.

مراحل موجود در طراحی سیستم های فبره

مراحل اصلی طراحی یک سیستم خبره عبارتند از:

- 1- تعیین یک بخش از دامنه برای نمونه اولیه.
- 2- تصمیم گیری برای نمونه اولیه.
- 3- ایجاد نمودار موکل.
- 4- ایجاد جداول تصمیم گیری.
- 5- کد نمودن با استفاده از پوسته مناسب.



شکل 8-1 : نمودار بلوکی عملیات اصلی سیستم کمک هزینه.

روش کار

برای توضیح اینکه مراحل نامبرده چگونه در عمل بکاربرده می شوند، مثالی را در زیر مطرح می کنیم. در اینجا سیستم خبره ای برای استفاده یک سازمان خیریه در یک حوزه محلی ایجاد می شود تا توصیه نامه های کمک هزینه را تنظیم کند. به عبارت دیگر به عنوان یک سیستم پشتیبانی تصمیم گیری برای بررسی اعتبار خانواده، کمک هزینه خانه سازی و کمک هزینه سرپرستی بکار گرفته می شود. شکل 8-1 نمودار بلوکی عملیات اصلی کمک هزینه را نمایش می دهد.

حال به تشریح مراحل اصلی طراحی می پردازیم.

تعیین یک فضا از دامنه برای نمونه اولیه

اگر پیچیدگی اولیه دامنه سیستم خبره زیاد باشد، آسان تر این است که با تمرکز به روی یک بخش از دامنه، نمونه اول را شکل دهیم. در نمودار بلوکی شکل 8-1 فضای مورد نظر برای ساخت نمونه اولیه، بخش کمک هزینه خانه سازی است (Housing benefit).

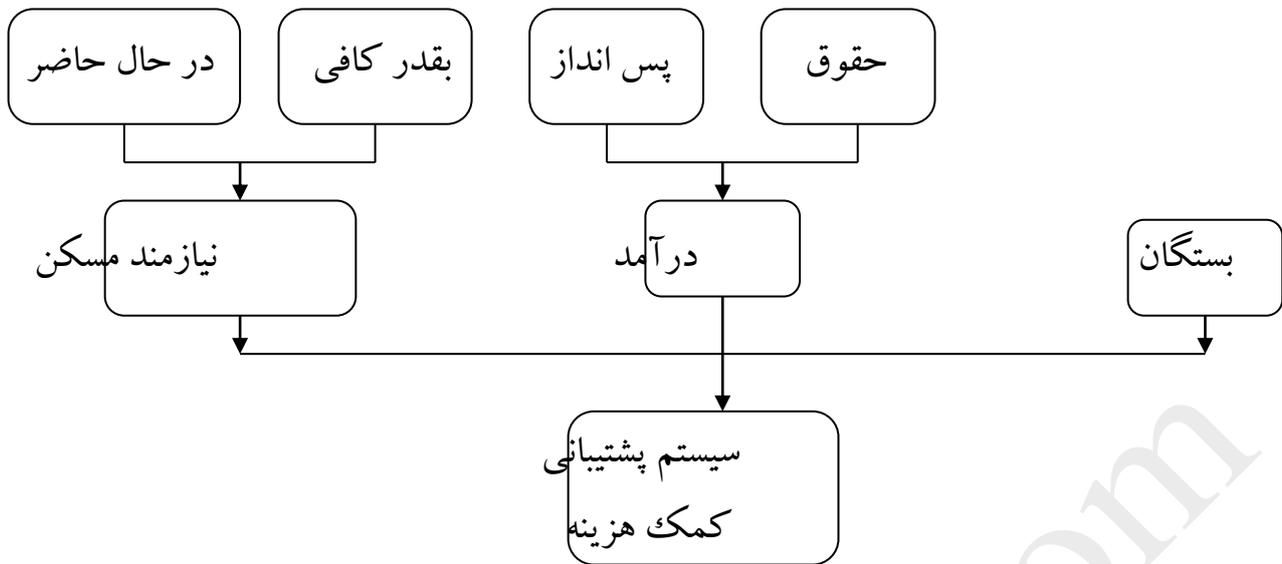
تصمیم گیری برای تعیین نمونه اولیه

انتخاب یک بخش از دامنه، گام بعدی تصمیم گیری برای ایجاد نمونه اولیه است. در اینجا باید پارامترهای مختلف تأثیرگذار در هدف مسأله را با توجه به میزان تأثیرشان مورد بررسی قرار داد. (در مثال فوق یک سیستم پشتیبانی تصمیم گیری برای شخص متقاضی کمک هزینه خانه سازی ایجاد نمود.

نمودار بلوکی شکل 8-2 روی این زیر مجموعه از دامنه تمرکز می کند و فاکتورهای تأثیرگذار بر تصمیم گیری را تعریف می کند.

◀ نیازمند مسکن (House need) آیا متقاضی واقعاً نیازمند کمک هزینه مسکن است؟ دو عامل در پاسخ به این سؤال تأثیرگذار می باشند: آیا متقاضی یک مسکن ثابت دارد؟ و اینکه آیا این جای ثابت پاسخگوی نیازهای او می باشد یا خیر؟ ارزش این دو مورد توسط پاسخ هایی که به این سؤال در زمان اجرا داده خواهد شد، مشخص می شود.

◀ درآمد (Income) آیا متقاضی بدون در نظر گرفتن وام دریافتی برای خانه سازی درآمد کافی برای گذراندن زندگی دارد؟ در اینجا نیز دو عامل در پاسخ به این سؤال مؤثرند: اولاً، آیا متقاضی پس اندازی دارد و ثانیاً حقوق متقاضی چقدر است؟ ارزش این دو پرسش نیز از پاسخ هایی که کاربر در طول زمان اجرا خواهد داد بدست می آید.



شکل 8-2: نمودار بلوکی تصمیم گیری برای نمونه نسخه اولیه

◀ وابستگان (Dependants). آیا متقاضی وابستگان نزدیک یا فرزند و غیره دارد یا خیر؟ پاسخ این سؤال کاملاً واضح می باشد و به هیچ عاملی بستگی ندارد و از طریق پاسخ به پرسش ها در زمان اجرا مشخص می شود. با توجه به پاسخ های داده شده، متقاضی می تواند خدماتی به صورت زیر دریافت کند:

- ◀ پشتیبانی سطح 1- متقاضی حداکثر کمک هزینه را دریافت می کند.
- ◀ پشتیبانی سطح 2- متقاضی حداکثر 80٪ کمک هزینه را دریافت می کند.
- ◀ پشتیبانی سطح 3- متقاضی حداکثر 50٪ کمک هزینه را دریافت می کند.
- ◀ پشتیبانی سطح 4- متقاضی حداکثر 20٪ کمک هزینه را دریافت می کند.
- ◀ هیچ مقدار کمک هزینه دریافت نمی کند.

نمودار موکلر

نمودار بلوکی شکل 8-2 یک روش مفید برای توضیح و توصیف ارتباطات بین عوامل و هدف است. اما برای ساخت سیستم خبره کافی نمی باشد، چون شامل جزئیات نیست. در اینجا نیاز به نموداری خواهیم داشت که از طریق سؤالات ورودی هدف را نمایان نموده، قوانین و توصیف هایی که توسط نمونه اولیه ساخته شده اند، را نشان دهد. یک نمودار موکلر یا نمودار وابستگی برای این کار مناسب است. برای ایجاد یک نمودار موکلر ابتدا نمودار 8-2 را به صورت افقی پرخانید. سپس برای هر فاکتور اصلی که در گام دوم مشخص شده یک جعبه مثلث شکل بکشید. در این مقطع فاکتورها عبارتند از: نیاز به مسکن (House need) وابستگان (Depndants) و درآمد (Income). از آنجایی که مورد وابستگان مستقیماً به توصیه نامه ها مربوط است. جعبه مربوط به آن می تواند از نمودار حذف شود. آنگاه روی هر خط مستقیم که به یک مثلث وارد می شود، کلمه یا عبارتی نوشته شود که بهترین توصیف برای مواردی باشد که از فاکتور اصلی خارج شود. این مورد یک متغیر نامیده می شود.

شکل 8-3: نمودار کامل موکلر برای کمک هزینه مسکن.

زیر هر خط مستقیم تمامی مقادیر ممکن که یک متغیر می تواند اختیار کند لیست نموده، همچنین اسامی برای مقادیری که فاکتور اصلی نمایش می دهد انتخاب شود. همین کار برای توصیه نامه ها نیز باید صورت گیرد. شکل 8-3 نمودار موکلر را نشان می دهد.

جدول تصمیم گیری

مرحله نهایی کار ما قبل از تولید قوانین در پایگاه دانش، ساخت جدول تصمیم گیری است. این جدول برای نمایش روابط داخلی مقادیر با خروجی هر فاز سیستم، مورد نیاز است. برای هر فاکتور در نمودار موکلر یک جدول تصمیم گیری جداگانه مورد نیاز است. بدین گونه که برای فاکتورهای، کمک هزینه، نیاز به مسکن و درآمد باید یک جدول تصمیم گیری ساخته شود. برای ساخت جدول تصمیم گیری از تصمیم اصلی شروع نموده و کار را به صورت عقب گرد از بالا به پایین انجام می دهیم. هر جدول تصمیم گیری یک مجموعه قوانین تولید می کند و هر مجموعه قوانین باید شماره گذاری شده و در مثلث مشخصی در نمودار موکلر نوشته شود. قبل از رسم هر جدول تصمیم گیری تعداد سطرهای مورد نیاز برای جدول را محاسبه می نمایم.

نوشتن قوانین برای پایگاه دانش (VP-Expert)

بیشتر لایه های سیستم خبره مبتنی بر قانون (از جمله VP-Expert) دانش را با استفاده از قوانین if...then نشان می دهند. این بدان معنی است که یک قانون به یک کلمه کلیدی if آغاز می شود و بوسیله شرط هایی ارزیابی می شود (یک سری شرط ها را بدنبال دارد). شرط ها می توانند توسط عملگرهای منطقی and و یا or به هم متصل شوند. برای تبدیل سطرهای جدول تصمیم تغییر یافته به قوانین، به راحتی می توان هر سطر را مانند آنچه در شکل 8-4 نشان داده شده است با استفاده از عملگرهای and و or به فرمت if...then تبدیل نمود. کلمه کلیدی if را قبل از اولین

ستون شرط در جدول قرار دهید. باقی شرط ها را با استفاده از عملگر and متصل نموده و کلمه کلیدی then را پیش از مقدار نتیجه درج می نماییم. قوانین برای مجموعه سطرهای جدول 2-8 همانطور که در شکل 4-8 نشان داده شده است (مجموعه قانون I) کد می شوند. توجه کنید که همیشه بین سطرهای جدول و قوانین تولید شده در شکل 4-8 تناظر یک به یک مستقیم برقرار نیست، مثلاً سطرهای B5 و B6 با استفاده از دستور OR برای این جدول خاص به یک ردیف تبدیل شده اند. نتیجه کارهای انجام شده این است:

B5 gives if house-need=yes and Sakary=full then Support=no-benefit

B6 giver if house-need=no then Support=no-benefit

RULE1

IF house-need=ye

 Dependints=yes AND

 Income=none

THEN support=level-1;

RULE2

IF house-need=yes AND

 Dependants=yes AND

 Income=some

THEN support=level-2;

RULE3

IF house-need=yes AND

 Dependants=no AND

 Income=some

THEN support=level-4;

RULE4

IF house-need=yes AND

 dependants=no AND

 Income=none

THEN support=level-3;

RULE5

IF house-need=no OR

 Income=full

THEN support=no-benetit;

شکل 4-8: مجموعه قانون 1- قوانین هدف اصلی.

RULE6

IF fixed-abode=yes AND adequate = yes THEN house-need=no;

RULE7

IF fixed-abode=yes AND adequate = no THEN house-need=yes;

RULE8

IF fixed-abode=yes adequate = no THEN house-need=yes;

شکل 5-8 : مجموعه قانون 2- نیاز مسکن.

این دو قانون با استفاده از عملگر OR ترکیب می شوند:

if salary=full or houses-need=no then support=no-benefit;

این فرآیند برای مثلث های بعدی نمودار موکلر نیز تکرار می شود تا همانگونه که در شکل 5-8

نشان داده شده است مجموعه قانون 2 ایجاد شود. سرانجام، این فرآیند برای سومین مثلث نمودار

موکلر نیز تکرار می شود که نتیجه آن قانون 3 است و در شکل 6-8 نشان داده شده است.

RULE9

IF salary=no AND savings=low THEN income=none;

RULE10

IF salary=no AND savings=high THEN income=some;

RULE11

IF salary=yes THEN income=full;

شکل 6-8 : مجموعه قانون 3- درآمد.

تبدیل قوانین به فرمت VP-Expert لایه مقصد (هدف)

مجموعه قوانین تولید شده خودشان به تنهایی برای سیستم مشاوره کافی نیستند زیرا هدف معین

نشده است. یعنی سیستم خبره احتیاج دارد که بداند قوانین برای انجام چه کارهایی تدارک دیده

شده اند. VP-Expert این کار را کلمه کلیدی FIND انجام می دهد، که گرامر کامل آن بصورت «FIND support» و به VP-Expert دستور می دهد تا مقداری برای پشتیبانی در طول زمان اجرای برنامه پیدا (FIND کند). این دستور همیشه در قسمت ACTIONS از یک برنامه VP-Expert قرار می گیرد.

ساختار برنامه VP-Expert

در VP-Expert یک برنامه به سه بلوک ساختار بندی شده است:

- 1) فعالیت ها¹.
- 2) قوانین².
- 3) سؤال ها³.

یک بلوک فعالیت مجموعه ای از موارد مورد بحث را جمع آوری می کند. این قسمت اولین بلاک در برنامه VP-Expert می باشد که کلمه کلیدی ACTIONS آغاز می شود. فعالیت ها نوعاً شامل نمایش (DISPLAY) تعدادی پیغام های اصلی زمان اجراء دستوری برای FIND و دستوری برای نمایش (DISPLAY) مقادیر این هدف می باشند. شکل 7-8 یک بخش فعالیت ACTIONS برای سیستم خبره کمک هزینه را نشان می دهد.

ACTIONS	
DISPLAY	“به سیستم مشاور کمک هزینه مسکن خوش آمدید”
FIND Support	
DISPLAY	“{Support}: کمک هزینه در نظر گرفته شده”

شکل 7-8: یک فعالیت (ACTIONS) برای سیستم خبره کمک هزینه.

یک برنامه VP-Expert با بلاک ACTIONS آغاز می شود، با بلاک RULES ادامه پیدا می کند

- 1.Actions
- 2.Rules
- 3.Questions

و در آخر با بلاک QUESTIONS تمام می شود. پرسشی که باید در برنامه VP-Expert پرسیده شود را می توان با مراجعه به نمودار موکلر (شکل 8-3) پیدا کرد. از آنجا که تمام متغیرهای سمت راست نمودار به چیز دیگری وابسته نیستند. بنابراین آنها سؤالاتی خواهند بود که توسط کاربر پایانه ای جواب داده می شود. بنابراین در سیستم کمک هزینه مسکن ، جواب های کاربر به سؤالاتی از قبیل : محل سکونت ثابت، رفاه، وابستگان، پس انداز و حقوق خواهد بود. شکل 8-8 یک بلاک سؤال مناسب را برای سیستم خبره کمک هزینه نشان می دهد. توجه داشته باشید که در گرامر کد VP-Expert بلاک پرسش ها با کلمه کلیدی ASK آغاز می شود. CHOISES که به دنبال برخی از پرسش ها می آید اختیاری است و تنها زمانی در گرامر سؤالات VP-Expert قرار می گیرد که نتیجه سؤالات از کاربر در انتخاب از یک لیست فهرستی محدود شده باشد. کد کامل برنامه VP-Expert در شکل 8-9 لیست شده است.

ASK fixed-abode: "Has the applicant a fixed abode or address?";

CHOICES fixed-abode:yes,no;

ASK adequate:"Has t the applicant accommodation?";

CHOICES adequate:yes,no;

ASK dependants: "Has the applicant any children or dependants?";

CHOICES dependants:yes,no;

AKS savings: "Has the applicant any savings?";

CHOICES savings: high, low;

AKS salary: "Has the applicant a regular sufficient salary?";

CHOICES salary:yes,no;

شکل 8-8 بلاک پرسش برای سیستم خبره کمک هزینه.

فلاصه مطالب فصل هشتم

مراحل اصلی طراحی یک سیستم خبره عبارتند از:

- 1- تعیین یک بخش از دامنه برای نمونه اولیه.
- 2- تصمیم گیری برای نمونه اولیه.
- 3- ایجاد نمودار موکلی.
- 4- ایجاد جداول تصمیم گیری.
- 5- کد نمودن با استفاده از پوسته مناسب.

اگر پیچیدگی اولیه دامنه سیستم خبره زیاد باشد، آسانتر این است که با تمرکز به روی یک بخش از دامنه، نمونه اول را شکل دهیم.

انتخاب یک بخش از دامنه، گام بعدی تصمیم گیری برای ایجاد نمونه اولیه است.

مرحله نهایی کار ما قبل از تولید قوانین در پایگاه دانش، ساخت جدول تصمیم گیری است.

برای ساخت جدول تصمیم گیری از تصمیم اصلی شروع نموده و کار را به صورت عقب گرد از بالا به پایین انجام می دهیم.

بیشتر لایه های سیستم خبره مبتنی بر قانون (از جمله VP-Expert) دانش را با استفاده از قوانین if...then نشان می دهند. این بدان معنی است که یک قانون به یک کلمه کلیدی if آغاز می شود و بوسیله شرط هایی ارزیابی می شود.

در VP-Expert یک برنامه به سه بلوک ساختار بندی شده است:

- 1) فعالیت ها¹.
- 2) قوانین².
- 3) سؤال ها³.

مقدمه

توسعه نرم افزارهای قراردادی در اغلب مراحل چرخه عمر قابل رویت می باشند. به این دلیل که شامل مراحل مشاهدات تا تولیدنهایی محصول می باشند. برای مثال یک شرکت تولید نرم افزارهای تجارتي که در نظر دارد بسته های نرم افزاری ایجاد کند باید در آغاز کار امکان سنجی را کاملاً منطبق با تجارت روز انجام دهد. اگر مطالعه امکان سنجی انجام شود و پاسخ آن مثبت باشد، آنگاه مراحل بعدی که شامل تجزیه و تحلیل جزئیات مسئله است باید انجام گیرد مرحله بعدی طراحی سیستم جدید می باشد و غیره.

بطور اتفاقی روش توسعه نرم افزار سیستم خبره در بعضی مواقع دارای شباهت هایی با سیستم های قراردادی می باشد. اختلاف های این دو سیستم در جدول 9-1 آمده است. این جدول تفکیک مراحل برای سیستم های قراردادی و سیستم های مبتنی بر اختلافات بصورت ویژه در فاز دوم نمایان است.

سیستم مبتنی بر دانش	سیستم قراردادی
1- مطالعه امکان سنجی	1- مطالعه امکان سنجی
2- مهندسی دانش	2- تجزیه و تحلیل سیستم
3- طراحی	3- طراحی
4- پیاده سازی	4- پیاده سازی
5- تست	5- تست
6- نگهداری	6- نگهداری

جدول 9-1: مراحل چرخه عمر سیستم های خبره و سیستم های قراردادی.

روش مرحله ای

در حال حاضر تفاوت اساسی بین سیستم های خبره و سیستم های قراردادی در فاز مهندسی دانش می باشد. مرحله طراحی ، پیاده سازی و تست همگی از این فاز تبعیت می کنند. مراحل موجود در این روش در شکل 9-1 نشان داده شده است .

روش نمونه سازی

تجزیه و تحلیل سیستم های قراردادی یک فعالیت خوش تعریف می باشد. جزئیات و مدت هر کاری تواند مستند شده ، نمایش داده شده، مدیریت و پشتیبانی می گردد(در صورتی که پروژه با منابع کافی زمانبندی شده باشد). اما مهندسی دانش بسیار متفاوت است . نیازهای مختلف سیستم خبره غالباً به سختی مشخص می گردد، برای اینکه نوعی چکیده از دانش محسوب می شود. در سیستم های خبره هدف های نهایی نوعاً بصورت واضح تعریف شده نیستند و این مسأله اختلاف اصلی و عمده بین یک سیستم خبره و یک سیستم قراردادی می باشد. مهندسی دانش به بسته بندی ، نمایش و ساختار دانشی که در طبیعت خلاصه می شود متمایل است . و نهایتاً بر روی پردازش داده متمرکز می شود. نیازهای مختلفی در سیستم های خبره ای که پایگاه دانش پیچیده ای دارند احساس می شود. بنابراین اصلاح پایگاه دانش ممکن است لازم باشد و این نشان می دهد که مهندس دانش باید سیستم را مجدداً بررسی نموده، با اصلاح آن موافقت کند. این عمل تکراری برای ایجاد توسعه سیستم خبره با موفقیت انجام می شود و این کار نمونه سازی نام دارد. توجه کنید که حلقه باز خوردی از مراحل مهندسی دانش تا مرحله تست متغیر است . در عمل نمونه سازی اغلب شامل تولید سریع یک نمونه است که هدف از آن ، ایجاد یک سکوی پرتاب برای کاربران و متخصصین می باشد. همچنین با هدف تشریح عملکرد سیستم برای مدیران ، که در مورد استفاده

از تکنولوژی سیستم خبره ممکن است دچار شک و تردید شده باشند. این امر ضروری به نظر می‌رسد. نمونه سازی در واقع عبارت است از: تغییرات اساسی در فرایند توسعه (نومان و جن کنیز 1982).

نمونه از تعداد نامحدودی نتایج باز خوردی و ارزیابی کاربران و متخصصین تشکیل شده است. نمونه سازی مزایای دیگری نیز دارد.

به منظور درک این مسأله که نمونه سازی چه طور ممکن است در اولین مرحله توسعه یک برنامه عمل نماید، مثالی می‌آوریم: یک سیستم پرداخت حقوق و دستمزد را در نظر بگیرید. توسعه چنین سیستمی شامل یک مجموعه ورودی عبارت از: نام، نام خانوادگی، شماره پرسنلی، میزان حقوق، میزان اضافه کاری، ساعات کار و غیره و خروجی‌ها شامل پرداخت‌ها مانند پرداخت مالیات و غیره می‌باشد. این فرآیند باید شامل محاسبه برای پرداخت مالیات باشد که از روش ضرب ساعات کار در نرخ آن بدست می‌آید. از اینرو، نیازهای چنین سیستمی به وضوح می‌تواند مشخص گردد. بنابراین استفاده از مدل مرحله‌ای در اینجا توسعه می‌گردد.

اما در مقابل یک سیستم خبره برای پرداخت مالیات در نظر بگیرید. نیازهای آن چه می‌باشد؟ برای ساخت چنین سیستمی ابتدا باید نیازهای مسأله، محدوده و اهداف مسدله کاملاً تعریف شود. این مسدله ممکن است یک سیستم بزرگ را در برگیرد. اما برای استفاده از روش نمونه اولیه توسعه دهنده تنها باید یک هدف و یک قسمت کوچک از دامنه (15% از آن) را به عنوان یک نقطه شروع تعریف کند. توسعه دهنده تنها روی یک هدف متمرکز می‌شود که شاید آن هدف مالیات بر درآمد پرداخت شده باشد. برای مثال، مالیات بر درآمد افراد مجرد. توسعه دهنده باید این دانش را به مجموعه‌ای کوچک از قوانین که در سیستم خبره استفاده می‌شود، تبدیل نماید.

نمونه اولیه ممکن است نسخه نهایی به نظر برسد و یا این طور بنظر برسد که این نسخه نیاز به پالایش و تغییرات اضافی دارد. این تغییرات و پالایش ممکن است شامل اضافه شدن قسمت های مختلفی در نمونه اولیه فعلی باشد. این نوع از نمونه اولیه «نمونه اولیه افزایشی (نموی¹)» نامیده می شود.

مزایای نمونه سازی

- ◀ نمونه سازی یک ایده روشن به تولید کنندگان پروژه می دهد که آیا عملیات با استفاده از تکنولوژی سیستم خبره کامل می شود یا خیر؟
- ◀ نمونه سازی یک روش سودمند برای تشریح سیستم در مراحل اولیه محسوب می شود.
- ◀ نمونه سازی می تواند مدیریت را با امکان نمایش سریع جذب نماید.
- ◀ نمونه سازی در حکم وسیله ای است برای متخصصین جهت انتقاد از سیستم یا ایجاد اصلاحات و استثنائات در قوانین.
- ◀ نمونه سازی هم از کاربر و هم متخصص پشتیبانی می کند.
- ◀ نمونه سازی اطلاعاتی در مورد تأثیر ابزارهای انتخاب شده و نیز روش نمایش دانش انتخاب شده، می دهد و امکانی برای ایجاد تغییرات لازم در مراحل اولیه کار می باشد.
- ◀ نمونه سازی مسائلی را که در طول مرحله امکان سنجی تعریف نشده است نشان می دهد.
- ◀ نمونه سازی سبب ایجاد تسهیلات مناسب برای سیستم های خبره در برخی دامنه های خاص می شود.

مطالعه امکانات سنجی

امکان سنجی یک سیستم خبره پیوسته چندین معیار را ارزیابی می کند و مشخص می نماید. بنابراین یک روش برای ارزیابی در اینجا لازم و ضروری می باشد. یک استراتژی ممکن که منسوب به بک من (1991) می باشد، که نسبت و میزان اهمیت آن را مشخص می نماید یک نشان (علامت) نیز به هر نتیجه نسبت داده می شود.

سه معیار مهم و عمومی برای مطالعه امکان سنجی در نظر گرفته می شود.

- 1- هزینه
- 2- خصوصیات
- 3- دسترسی به متخصصین (خبرنگاران)

هزینه

یک محدوده از هزینه باید در ابتدا توسعه سیستم خبره در نظر گرفته شود منابع باید برای تجزیه و تحلیل، طراحی، کدنویسی، پیاده سازی و نگه داری تعریف شوند. بعلاوه هزینه سخت افزار و نرم افزار نیز باید مشخص شود. هزینه های منابع دانش نیز باید مشخص شوند.

مخصوصیات

همانگونه که در فصل های قبلی دیده شد، سیستم تهای خبره در مسائلی که احتیاج به یک راه حل ذهنی (هیوریستیک) برای حل مسأله داشتند، موفق عمل کرده اند به هر حال، سیستم های خبره برای همه انواع مسائل مناسب نیستند. تجزیه نشان داده است که همه مسائل تابع حل سیستم خبره نیستند. تست تلفن برای تعیین آنهایی که تابع حل سیستم خبره هستند یک مقیاس مناسب می باشد. اگر فرد خبره ای بتواند از طریق مکالمه تلفنی با کاربر مسئله ای را حل کند، احتمالاً یک سیستم خبره می تواند برای حل این مسئله توسعه پیدا کند. اساس عقلی این تست از این حقیقت که خبره باید به اطلاعات اضافی که در منابع دیگر ناشی می شود دسترسی نداشته باشد و کاربر

قادر به تشریح شفاهی مسئله باشد پیروی می کند. برعکس، اگر کاربر قادر به تشریح شفاهی مسئله نباشد و یا فرد خبره در گفتگوی تلفنی یک راه حل مناسب ارائه ندهد، احتمال توسعه سیستم خبره با شکست روبه رو خواهد شد. برای موفقیت در تست تلفن، زمان مصرف شده برای حل مسئله، توسعه خبره، نباید متجاوز از 30 دقیقه باشد.

دسترسى به متخصصین (فبرگان)

خبرگان باید در محدوده مسئله تعریف شده باشند و قادر باشند زمان کافی را برای انجام پروژه اختصاص بدهند. بعلاوه خبره باید مؤثر در کار و مشتاق به کار باشد، در غیر این صورت شانس موفقیت کمتر است. وقتی که پروژه تعریف شد و اجازه داده شد، فرآیند مهندس دانش می تواند شروع شود.

هدف سیستم

فاز مهندسی سیستم شامل تعدادی مرحله (زیر فاز) می باشد. واضح است که هدف اصلی برای توسعه سیستم باید از ابتدا مشخص و معلوم باشد. برای مثال هدف ممکن است نیازمند به پشتیبانی یک فرد خبره، یا افزایش کیفیت و یا سازگاری انجام برخی از کارها داشته باشد. اینها، کارهای متفاوتی هستند که برای توسعه و ایجاد، نیازهای متفاوتی را خواهانند.

دامنه تجزیه و تحلیل فبره

جنبه دیگر فاز مهندسی دانش تعریف ویژگی های اصلی دامنه می باشد. لازم نیست مهندس دانش توانایی لازم را در زمینه دانش مسئله داشته باشد، بلکه توانایی مهندس دانش بستگی به میزان درک او از مفاهیم دارد. این مرحله ادراک¹ نامیده می شود.

1. Conceptualisation

نگهداری سیستم های فبره

تعداد کمی از سیستم های خبره برای مدت طولانی به صورت ایستا باقی می ماند برای مثال یک سیستم خبره مالیات به تغییرات سالیانه برای انعکاس میزان تغییرات بودجه نیاز دارد.

اگر سیستم های خبره بخواهند همواره بطور مؤثر باقی بمانند باید طوری باشند که تغییرات ایجاد شده در سازماندهی شرکت ها و یا تغییرات در اقتصاد، سیاست و یا تغییرات فرهنگی در آنها نیز اعمال می شود. تاکنون بسیاری از سیستم های خبره به اندازه کافی نگهداری نشده اند. فانن باوم (1993) به چیزی معتقد بود که "فنای دانش نامیده می شود" به این معنا که عدم نگهداری سیستم های خبره یکی از فاکتورهایی است که مانع پذیرش سیستم های خبره می شود.

عامل اصلی برای ایجاد تغییرات در یک سیستم خبره از برخی منابع ناشی می شود. برای مثال یک متخصص ممکن است سیستم را تغییر دهد تا تغییرات دانش در محدوده دامنه انعکاس پیدا کند همچنین کاربر ممکن است بعضی تغییرات را برای نیازهای شخصی خود اعمال کند.

تغییرات چنین می تواند لازم باشد. مانند تغییرات در تولید یک کمپانی. تغییرات دیگر ممکن است احتیاج باشد مثل تغییرات در سیستم عامل برای نسخه جدید. مهندس دانش ممکن است به ایجاد تغییرات نیاز پیدا کند و بخواهد سیستم را به واسطه های خارجی مثل پایگاه داده ها و یا پایگاه قوانین پیوند دهد.

طراحی برای نگهداری

دو فاکتور اصلی که در قابلیت نگهداری سیستم خبره موثر هستند عبارتند از:

1- قابلیت ادراک: اگر درک کد برنامه آسان باشد، آنگاه بهتر قابل نگهداری است یعنی مهندسی نگهداری سیستم که اعضای اصلی تیم توسعه سیستم نیستند راحت ترمی توانند از سیستم نگهداری کنند.

2- قابلیت ایجاد تغییر: به این معنا که ساختار سیستم خبره به گونه ای باشد که بتوان راحت تر سیستم را توسعه داد و تغییرات را در آن ایجاد نمود.

این دو قابلیت باهم ارتباط دارند، اما قابلیت ادراک مهم تر است. بعید به نظرمی رسد که نگهداری برنامه جای نگرانی داشته باشد. تولید کنندگان مجرب سیستم های خبره باید روش های تولید برنامه های قراردادی را به منظور برنامه های قابل نگهداری قبول نمایند. نکاتی از قبیل: در نظر گرفتن خط فاصله در کد برنامه، در نظر گرفتن یکسانی خطوط کد برنامه، استفاده از فرامین، استفاده درست از کاراکترهای کوچک و بزرگ و... علاوه بر این کد برنامه باید به روش پیمانه ای طراحی شده باشد که این کار خود نگهداری برنامه را آسان ترمی کند.

خلاصه مطالب فصل نهم

در حال حاضر تفاوت اساسی بین سیستم های خبره و سیستم های قراردادی در فاز مهندسی دانش می باشد. مرحله طراحی، پیاده سازی و تست همگی از این فاز تبعیت می کنند. تجزیه و تحلیل سیستم های قراردادی یک فعالیت خوش تعریف می باشد. جزئیات و مدت هر کاری تواند مستند شده، نمایش داده شده، مدیریت و پشتیبانی می گردد. نیازهای مختلفی در سیستم های خبره ای که پایگاه دانش پیچیده ای دارند احساس می شود. بنابراین اصلاح پایگاه دانش ممکن است لازم باشد و این نشان می دهد که مهندس دانش باید سیستم را مجدداً بررسی نموده، با اصلاح آن موافقت کند.

سیستم تهای خبره در مسائلی که احتیاج به یک راه حل ذهنی (هیوریستیک) برای حل مسأله داشتند ، موفق عمل کرده اند.

جنبه دیگر فاز مهندسی دانش تعریف ویژگی های اصلی دامنه می باشد. لازم نیست مهندس دانش توانایی لازم را در زمینه دانش مسئله داشته باشد، بلکه توانایی مهندس دانش بستگی به میزان درک او از مفاهیم دارد. این مرحله ادراک¹ نامیده می شود.

اگر سیستم های خبره بخواهند همواره بطور مؤثر باقی بمانند باید طوری باشند که تغییرات ایجاد شده در سازماندهی شرکت ها و یا تغییرات در اقتصاد، سیاست و یا تغییرات فرهنگی در آنها نیز اعمال می شود.

موضوعات

آنچه در این فصل خواهید آموخت :

- ◀ چه وقت از سیستم های خبره استفاده کنیم.
- ◀ کاربردهای موفق سیستم های خبره
- ◀ شناخت برخی زمینه های کاربرد سیستم های خبره
- ◀ چرا مدیران سازمان ها مایل به استفاده از سیستم های خبره نیستند.
- ◀ آینده سیستم های خبره

مقدمه

بازار کنونی سیستم خبره و کاربردهایی که در آینده مطرح خواهند بود از جمله موضوعات مورد بحث در این فصل می باشند.

کاربرد سیستم های خبره

سیستم های خبره در زمینه های مختلفی از جمله صنعت ، تجارت و کاربرد های مالی مفید می باشند. در واقع زمینه های مختلف کاربرد سیستم خبره امروزه چنان گسترده است که تقریباً در انجام هر تصمیمی که توسط انسان گرفته میشود موفق هستند. بیشترین کاربردها در زیر بر حسب وظایف طبقه بندی شده اند :

(هایس - رود، واترمن و لدنت 1993) :

- ◀ سیستم های تشخیص : این سیستم ها در مشاغل مختلف وظایف مشکلی را بر عهده می گیرند، کاربرد این نوع سیستم ها برای مثال در پزشکی ، مهندسی و نرم افزار تشخیص الگو می باشد.

◀ سیستم های طراحی و زمانبندی: سیستم هایی هستند که عملیات را طراحی می کنند برای مثال برنامه ریزی اتوماتیک، حرکت رباط، استراتژی نظامی و حتی ساعت حرکت قطار.

◀ سیستم های مفسر: سیستم هایی هستند که شرح مشاهدات را برعهده دارند. مانند سیستم های مراقبت (نظارت) سیستم های تشخیص گفتار.

◀ سیستم های پیش بینی: سیستم هایی هستند که نتایج موقعیت ها و وضعیت ها را پیش بینی می کنند مانند پیش بینی ترافیک یا پیش بینی وضع هوا.

هدف یک سیستم خبره فقط تصرف دامنه تخصص نیست. بلکه شبیه سازی عملکرد حل مسئله در یک محدوده خاص است. جدول 1-10 کاربردهای سیستم خبره را شرح می دهد.

خودآزمایی

شرکت های مالی و اعتباری از سیستم های خبره برای سرویس دهی به مشتریانی که برای دریافت وام و یا پول مراجعه می کنند استفاده می نمایند. آیا شما می توانید سه لیست یا بیشتر از کاربردهای مفید سیستم خبره در این زمینه ارائه کنید؟

زمینه های کاربرد جدید در سیستم های خبره

سیستم هایی مانند MYCIN، سیستم های خبره پزشکی هستند که برای تشخیص امراض عفونی بکار می روند، و سیستم PROSPECTOR، سیستم های خبره ای است که برای کشف معادن بکار می رود. بخش دیگری از سیستم های خبره، کاربرد عمومی در زمینه های تشخیص و طبقه بندی دارند. امروزه زمینه های کاربردی جدیدی ایجاد شده، که به خوبی خود را با سیستم های خبره وفق میدهند. اینها عبارتند از: نشر دانش، سیستم های کمک آموزشی، ابزارهای اینترنت، وب و

جدول 10-1 : انواع سیستم‌های فبره.

تفصص سیستم فبره	تشریح	کاربرد سیستم
تفسیر	ترجمه و توضیح داده‌های حساس را انجام می‌دهد.	مشاوره در VAT
تشخیص	عملیات طراحی را انجام می‌دهد	DART- توسط دولت آمریکا در جنگ خلیج برای نقشه مسیر انتقال سربازان و مهمات استفاده شد
طراحی	شکل دادن به اشیا طبق مدل	XCON- برای طراحی و شکل دادن به سفارشات مشتریان در کامپیوترهای VAX
پیش بینی	به چیزهایی نظیر نتایج وضعیت‌ها و یا حوادث اشاره می‌کند	PROSPECTOR- یک سیستم کشف معدن که می‌تواند معجل معادن را در نواحی مختلف تخمین بزند
نمایش	این سیستم نمایشی اکثراً برای نظارت و مواظبت بر علیه متجاوزان بکار می‌رود.	FRAUDWATCH- این سیستم دربانک بارکلیز برای تشخیص کارت‌های اعتباری تقلبی استفاده می‌شوند.
آموزش	شناسایی دانش آموزان خاطی و آموزش آنها	GUIDON- یک سیستم برای آموزش مسائل و مشکلات
کنترل	تشخیص، پیش بینی و نمایش رفتار سیستم را برعهده دارد	VM- یک سیستم که بیماران را در بخش پرستاری نشان می‌دهد و طرز مداوای بیماران را کنترل می‌کند. این سیستم حالت بیمار را شرح می‌دهد و بنابر اطلاعات هر خطری که بیمار را تهدید می‌کند تشخیص داده و راه

		درمان مفید را پیشنهاد می دهد
تعمیر	تولید روش های ترمیم معایب و خرابی های سیستم	TQMSTONE- سیستمی است که معایب و خرابی های سیستم را تشخیص داده، سپس روش های ترمیم را ارائه می کند
اشکال زدایی	تولید روش هایی برای رفع خرابی های سیستم	ONOCIN- یک سیستم که به پیدا کردن معایب و رفع آن کمک می کند.

نشر دانش

نشر و گسترش دانش یکی از کاربرد های سیستم خبره می باشد. این نشر دانش از لحاظ مفهوم یک کتاب بسته بندی شده است. کتاب یک شیء بی اراده است و برعهده خواننده می باشد. که چگونه از آن استفاده کند و سود ببرد. نشر دانش، علم و دانش را در اختیار کاربران قرار می دهد و با توجه به اینکه کاربر چه درخواست و تقاضایی داشته باشد از آن استفاده می کند.

کاربرد های کمک آموزشی

سیستم های کمک آموزشی شبیه حل المسائل می باشند که زمینه رشد در آینده را داشته و می توانند نجات بخش سیستم های خبره باشند (1995- براون) بیشتر سیستم های کمک آموزشی از سیستم های خبره مبتنی بر CBR استفاده می نمایند. هنگامی که مردم با مشکلی در زمینه کار با وسیله و یا خواندن راهنمای آن مواجه می شوند، از طریق تلفن با مرکز تولید آن ارتباط برقرار کرده و مشکلشان را حل می نمایند. که در اینجا می توان از سیستم های خبره مبتنی بر CBR استفاده نمود. به عنوان مثال، شرکت کامپیوتری کامپک دارای یک سیستم کمکی چاپگر بر خط

می باشد که با همه چاپگرها ارتباط بر خط دارد این برنامه Quick source نامیده می شود که می تواند 5000 مورد از اشکالات چاپگرها را پشتیبانی نماید و تخمین زده می شود که به میزان 20٪ نسبت به حالتی که مشتریان برای شتیبانی تماس تلفنی برقرار می نمایند صرفه جویی در هزینه و زمان انجام می شود.

واسط های هوشمند

واسط های هوشمند به صورت اتوماتیک کارهای مهمی را انجام می دهند که تنها توسط انسان قابل انجام است بسیاری از کاربران کامپیوتر تجربیات سختی را در کسب و دسترسی به اطلاعات مورد نیاز خود در سیستم های کامپیوتری داشته اند. گاهی اوقات کاربران وقت بسیار زیادی را صرف می نمایند تا اطلاعات مورد نیازشان را بدست آورند. یک نمونه از این کارها کسب اطلاعات از شبکه جهانی وب است. یک راه برای تسهیل این امر این است که برخی از عملیات واسط ها را اتوماتیک نماییم. سیستم هایی که اطلاعات را فیلتر می کنند یا اطلاعات را کشف و شرح می دهند، بطور اتوماتیک وظایف بیشتری را برعهده کامپیوتر قرار می دهند و کار کاربر را کمتر می کنند. برای موفقیت این واسط ها باید بیشترین کارایی و کمترین خطا را داشته باشند.

سیستم های خبره و شبکه جهانی وب

واسط های هوشمند کاربردهای مختلف وسیعی می توانند داشته باشند، هدف اصلی، ارائه و تولید سیستم هایی است که بتواند با زبانی طبیعی بحث و گفتگو کند، بیند، بشنود و به طور اتوماتیک استدلال نمایند. به عنوان مثال شبکه جهانی وب یک کتابخانه وسیع از اطلاعات می باشد و کاربران زمان زیادی را برای پیدا کردن اطلاعات از این منبع بزرگ جهانی صرف می کنند. سیستم های خبره ای برای کمک به کاربران برای دسترسی سریع به بانک های اطلاعاتی خاص

در شبکه وب توسعه یافتند. به عنوان مثال این سیستم های خبره ممکن است از کاربران تعدادی سؤال مربوط به استخراج اطلاعات و تشخیص مشکل پرسند سپس سیستم بطور مستقیم به کاربران، اطلاعاتی مربوط به راه حل صحیح مسئله ارائه دهد.

پیکربندی

سیستم پیکربندی XCON یکی از بهترین سیستم های شناخته شده امروزی است که از آن استفاده می شود. شرکت DEC از این سیستم در ترکیبات کامپیوتر VAX برای پاسخ به سفترشات مشتریان استفاده نمود. سیستم حدود سال 1980 کامل شد و یک کاربرد بسیار موفقیت آمیز داشت، صرفه جویی زیاد برای شرکت و سود قابل توجه برای مشتریان به دلیل پردازش سریع سفارش تا حد امکان، از جمله مزایای این سیستم می باشد.

پردازنده های پیشروی هوشمند

یک پردازشگر پیشروی هوشمند (IFE) نرم افزاری است که واسط بین یک کاربر و یک برنامه نرم افزاری است. یک IFE از تکنیک های KBS یا AI جهت تأثیر بیشتر استفاده می کند. مثال های کلاسیک بسیاری در یک نرم افزار بانک اطلاعاتی یافت می شود. یک IFE واسط آسان و مفیدی برای استفاده از یک بانک اطلاعاتی ایجاد می کند. یعنی با انعطاف پذیری بیشتر با کاربر ارتباط برقرار می کند. IFE بر مبنای درک خواسته ها و نیازهای کاربر عمل می کند و سپس این مشخصات را برای جمع آوری دستورات در اجرای بسته نرم افزاری بکار می برد. ارتباط با مشتری اغلب عامل مؤثر خواهد بود IFE ها همچنین وقتی که ارتباط با کاربر قطع می شود از یک تکنیک ویژه استفاده می کنند، به این صورت که یک مشخصه از کاربر ایجاد می نمایند. مانند super

base