





بنیاد نخبگان ستاد کل

شرکت صنایع الکترونیک ایران

شرکت ایز ایران

گزارش میانی فعالیت دوره ضرورت نخبگان بخش دفاع

عنوان

پیاده سازی و مجتمع سازی ماژول مدیریت قوانین کسب و کار در زیر ساخت تولید

نرم افزار رایتن

استاد راهنما

جواد پوردیلمی

مجری

سعید شکوهی

آبان ۱۳۹۴

فهرست مطالب

شماره

عنوان مطالب

صفحه

چکیده ۱

۱ فصل اول : کلیات مفاهیم موتورهای قوانین کسب و کار ۸

۸-۱	مقدمه	۸
۹-۱	قوانین کسب و کار	۹
۱۰-۱	مقدمه ای بر موتور قوانین کسب و کار	۱۰
۱۰-۴	تعریف موتور قوانین کسب و کار	۱۰
۱۰-۴-۱	پیشینه موتور قوانین کسب و کار	۱۰
۱۶-۱	علل استفاده از موتور قوانین کسب و کار	۱۶
۱۶-۵-۱	مزیت های استفاده از موتور قوانین	۱۶
۱۹-۵-۱	شرایطی که باید از موتورهای قوانین استفاده کرد	۱۹
۲۱-۶-۱	شرایطی که نباید از موتور قوانین استفاده کرد	۲۱
۲۲-۷-۱	ویژگی های سیستم های مدیریت قوانین کسب و کار	۲۲
۲۳-۷-۱	کامپوننت ها و ویژگی های فنی سیستم مدیریت قوانین کسب و کار (BRMS)	۲۳
۲۵-۷-۱	قوانین	۲۵
۲۵-۷-۱	قالب های قوانین	۲۵
۲۵-۷-۱	چک کردن نحوی قوانین	۲۵
۲۶-۷-۱	رویه ها و الگوریتم ها	۲۶
۲۶-۷-۱	جریان قوانین	۲۶
۲۶-۷-۱	جداول و درخت تصمیم گیری	۲۶
۲۷-۷-۱	استنتاج	۲۷
۲۸-۷-۱	عدم قطعیت و شرح دهی	۲۸

۲ فصل دوم : بررسی سیستم های مدیریت قوانین کسب و کار موجود ۳۰

۳۰-۱-۲	مقدمه	۳۰
۳۱-۲-۲	معرفی سیستم های مدیریت قوانین موجود	۳۱
۳۱-۲-۲	Drools	۳۱
۳۲-۲-۲	Jess	۳۲
۳۳-۲-۲	Hammurapi Rules	۳۳
۳۴-۲-۲	InfoSapient	۳۴
۳۵-۲-۲	JLisa	۳۵
۳۵-۲-۲	Mandarax	۳۵
۳۶-۲-۲	OpenLexicon	۳۶
۳۶-۲-۲	OpenRules	۳۶

۳۷.....	<i>Jamacho (morendo)</i> (۹-۲-۲)
۳۷.....	<i>JRuleEngine</i> (۱۰-۲-۲)
۳۹.....	(۳-۲) مقایسه موتور قوانین کسب و کار.....

فهرست شکل ها و نمودارها

۱۳	شکل ۱- نمایش سطح بالایی از موتور قوانین ...
۱۴	شکل ۲- روش استنتاج زنجیره ای رو به جلو
۱۵	شکل ۳- روش استنتاج زنجیره رو به عقب
۲۴	شکل ۴- معماری سطح بالا سیستم مدیریت قوانین کسب و کار
۳۲	شکل ۵- معماری سیستم مدیریت قوانین Drools
۳۴	شکل ۶- معماری کامپوننت های Hammurapi Rules
۳۷	شکل ۷- معماری سیستم مدیریت قوانین OpenRules

چکیده

از آنجا که اغلب نرم افزار های مورد نیاز وزارت دفاع ، نرم افزار های مبتنی بر قانون می باشند و حجم قوانین منطقی آنها زیاد است . از سوی دیگر، جزئیات این قوانین ثابت نیست و با توجه به شرایط موجود تغییر می کند . اعمال قوانین در نرم افزار با توجه به حجم زیاد قوانین باعث ایجاد کدهای پیچیده ای میشود که اغلب نگهداری این کدها در بلند مدت بسیار دشوار خواهد بود . و مهمتر از آن در حالتی که قوانین در زمان اجرای نرم افزار تغییر میکنند نیاز است که توسط برنامه نویس در کد تغییرات اعمال شود و بعد از کامپایل مجدد در برنامه قابل مشاهده خواهد بود . در حالی که نیازمندی این است که کاربر نهایی آموزش دیده بتواند در حالت اجرا قوانین را تغییر دهد و بدون متوقف شدن نرم افزار اثر تغییرات را در برنامه ببیند .

فصل اول

کلیات مفاهیم موتورهای قوانین کسب و کار

(۱) فصل اول : کلیات مفاهیم موتورهای قوانین کسب و کار

(۱-۱) مقدمه

کسب و کار بر اساس قوانین اجرا می شوند . آنها فرآیندهای کسب و کار را تعریف می کنند، و توصیف می کنند که اگر چیزی درست باشد یا غلط باشد چه اتفاقی خواهد افتاد . مانند اینکه "همه مشتریان با رتبه طلایی ۱۰ درصد تخفیف بگیرند و خدمات خاصی دریافت کنند " . قوانین کسب و کار بخش‌هایی از سیستم های پشتیبانی کننده از تصمیم گیری^۱ هستند که هر فرآیند کسب و کار را پی ریزی می کنند .

می توان قوانین را در منطق کسب و کار برنامه کاربردی پیاده سازی کرد. بدین صورت که اگر بخواهید هر قانون را بصورت شرط های تودرتو بنویسید . آنگاه بعد از مدتی به کدهای اسپاگتی^۲ تبدیل خواهند شد . از سوی دیگر در طول عمر^۳ نرم افزار قوانین کسب و کار تغییر خواهند کرد . تغییر و خطایابی^۴ هر شرط و قانون، پیچیدگی و زمان توسعه و تست نرم افزار را افزایش خواهد داد و در عین حال صاحب فرآیند کسب و کار منتظر اعمال آخرین تغییرات قوانین کسب و خواهد ماند .

درست این است که شما کد های قوانین را خودتان ننویسید . راحت تر است که از یک موتور قوانین کسب و کار برای رسیدگی ، مدیریت و سازماندهی قوانین استفاده کنید . موتورهای قوانین اعلانی^۵ هستند و این اجازه را به شما می دهند که گروه های قوانین را که رویدادها و سرویس ها را راه انداز می کنند . بدون سرزدن به برنامه نویسی رویه ای، به سرعت در کنار هم قرار دهید. همچنین اگر موتور قوانین کسب و کار امکانات و ابزارهای مناسبی برای ایجاد و تغییر قوانین فراهم کند . می توان مسئولیت ایجاد و

¹ Decision Making

² Spaghetti Code

³ Application Life-cycle

⁴ Debugging

⁵ Declarative

تغییر قوانین کسب و کار به صاحبین کسب کار واگذار کرد. این بدین معناست که تیم توسعه، بعد از هر تغییر قوانین کسب و کار نیاز به تغییر کد های برنامه کاربردی ندارند.

۱-۲) قوانین کسب و کار^۶

قانون کسب و کار، یک قاعده برای یک کسب و کار، شرکت و یا یک سازمان است. قانون کسب و کار، قانونی است که برخی از جنبه های کسب و کار را تعریف و یا محدود کرده و همیشه به حل درست یا نادرست برخی از مسائل در این حوزه ها می انجامد. قوانین کسب و کار برای اثبات ساختار کسب و کار و یا برای کنترل یا تحت تاثیر قرار دادن رفتار کسب و کار در نظر گرفته می شود. قوانین کسب و کار عملیات ها، تعاریف و محدودیت هایی را که در یک سازمان اعمال می شود را توضیح می دهد. قوانین کسب و کار می تواند در مورد افراد، فرآیندها^۷، رفتار سازمانی و سیستم های محاسباتی در سازمان اعمال شود و برای کمک به سازمان در دستیابی به اهداف خود در جای خود مورد استفاده قرار گیرد. به عنوان مثال یک قانون کسب و کار ممکن است اینگونه تصریح کند که نیازی به بررسی اعتبار در مورد مشتریان خاص نیست. نمونه های دیگر از قوانین کسب و کار عبارتند از: نیاز به یک عامل اجاره برای پذیرفتن یک مستاجر اجاره ای در صورت پایین بودن رتبه بندی اعتباری آن و یا نیاز به عوامل شرکت برای استفاده از تعدادی از تامین کنندگان ترجیحی و برنامه های عرضه. در حالی که یک قانون کسب و کار ممکن است رسمی یا حتی نانوشته باشد نوشتن قواعد روشن و اطمینان از اینکه این قوانین با سایر قوانین در تضاد نیست خود یک فعالیت ارزشمند محسوب می شود. قوانین کسب و کار جزئیات اقدامات را به یک سازمان معرفی می کند در حالی که استراتژی از چگونگی تمرکز کسب و کار در سطح کلان برای بهینه سازی نتایج صحبت می کند. قوانین کسب و کار در مورد نحوه تبدیل یک استراتژی به عمل رهنمودهای دقیق ارائه می کنند. قوانین کسب و کار برای یک سازمان در مورد آگاهی های سازمانی صحبت می کند. یک سازمان ممکن است

⁶ Business Rules

⁷ Process

بصورت فعالانه برای شرح شیوه های کسب و کار خود، اقدام به تولید یک پایگاه اطلاعاتی از قوانین کسب و کار نماید.

۱-۳) مقدمه ای بر موتور قوانین کسب و کار^۸

موتورهای قوانین با سنجش و ارزیابی حقایق^۹ و استفاده از نتایج آنها حقایق جدیدی را ایجاد می-کنند. بر اساس این تعریف بزرگترین مشکل پیش رو برای استقرار موتور قوانین کسب و کار، پیدا کردن ابزارو تکنولوژی مناسب نیست بلکه تعریف صحیح قوانین است. این مهم است که همه قوانینی که هم اکنون در حال استفاده هستند منعکس شوند و انتقال قوانین کار ساده ای نیست. تجربه نشان داده است که قوانین به همان اندازه که صریح و روشن هستند به همان اندازه ضمنی و پنهان نیز می باشند.

۱-۴) تعریف موتور قوانین کسب و کار

۱-۴-۱) پیشینه موتور قوانین کسب و کار

هوش مصنوعی^{۱۰} حوزه گسترده ای از تحقیقات است که بر روی "فکر کردن کامپیوتر مانند انسان" تمرکز دارد. و شامل بخش های مانند شبکه های عصبی^{۱۱}، الگوریتم ژنتیک^{۱۲}، درخت تصمیم گیری و سیستم های خبره^{۱۳} می باشد. نمایش دانش^{۱۴} بخشی از حوزه هوش مصنوعی است که با اینکه چگونه دانش بازنمایی و دستکاری می شود مرتبط است. سیستم های خبره نیز از سیستم های مبتنی بر دانش برای تسهیل کد کردن دانش به پایگاه دانشی که جهت استنتاج قابل استفاده باشد، بهره می برد. سیستم های خبره نیز بعنوان سیستم های مبتنی بر پایگاه دانش شناخته می شوند و سیستم های خبره مبتنی بر پایگاه دانش

⁸ Business Rule Engine (BRE)

⁹ Facts

¹⁰ Artificial Intelligence (A.I)

¹¹ Neural Network

¹² Genetic Algorithms

¹³ Expert Systems

¹⁴ Knowledge Representation

نیز به عنوان هوش مصنوعی کاربردی در نظر گرفته می‌شوند. فرآیند توسعه با یک سیستم خبره مهندسی دانش گفته می‌شود.

سیستم های مدیریت قوانین کسب و کار^{۱۵} امکانات افزوده ای را بر روی موتور قوانین همه منظوره با فراهم کردن امکان ایجاد، مدیریت، توسعه ترکیب و آنالیز قوانین برای کاربران کسب و کار می سازند. که به رسمی کردن نقش قوانین موتور قوانین در فرآیند تجاری کمک می کنند.

موتور قوانین به هر سیستمی گفته می شود که به هر شکلی از قوانین استفاده می کند تا بر روی داده ها اعمال می شود و خروجی تولید می کند. که شامل سیستم های ساده مانند اعتبارسنجی و موتورهای عبارت های پویا می شود.

یک سیستم قوانین تولید زمانی کامل می شود که بر روی بازنمایی دانش جهت ارائه گزاره ای و منطق مرتبه اول در یک روش مختصر، اعلانی و نامبهم تمرکز داشته باشد. هسته متفکر یک سیستم قوانین تولید^{۱۶}، یک موتور استنتاج حقایق و داده ها را با قوانین تولید مطابقت می دهد و عملیات متناظر را انجام می دهد. یک قانون تولید از ساختار دوبخشی تشکیل شده است که از منطق مرتبه اول برای استنتاج بر روی بازنمایی دانش استفاده می کند.

when

<conditions>

then

<actions>;

¹⁵ Business Rule Management Systems

¹⁶ Production Rules System

فرآیند انطباق^{۱۷} بر روی حقایق جدید یا موجود با قوانین تولید را الگوی انطباق^{۱۸} می گویند . که توسط موتور قوانین انجام می شود . تعدادی الگوریتم برای الگوی انطباق وجود دارد و توسط موتورهای استنتاج استفاده میشوند:

- Linear
- Rete
- Treat
- Leaps

به عنوان مثال موتور قوانین Drools از الگوریتم Rete استفاده می کند . الگوریتم leaps قبلاً استفاده میشده اما بدلیل غیر قابل نگهداری بودن آن استفاده نمی شود . پیاده سازی الگوریتم Rete توسط Drools به نام ReteOO شناخته می شود . که نشان می دهد که این الگوریتم برای سیستم های مبتنی بر اشیا،^{۱۹} توسعه داده شده است . دیگر موتورهای استنتاج مبتنی بر الگوریتم Rete ، مانند RetePlus و Rete III که این الگوریتم با توجه به زمینه کاربرد خود توسعه داده اند . عمومی ترین توسعه بر روی سیستم های آموزش استفاده شده که Rete/UL نام دارد .

قوانین در حافظه تولید نگهداری^{۲۰} می شوند و حقایق^{۲۱} که موتور استنتاج آنها را با قوانین تطابق میدهد در حافظه کاری^{۲۲} نگهداشته می شوند . حقایق به حافظه کاری وارد می شوند و در آنجا ممکن است تغییر کنند . یک سیستم با تعداد زیادی از قوانین و حقایق . این امکان وجود دارد که چندین قانون با یک داده موجود در حافظه کاری مطابقت داشته باشد . در این صورت گفته می شود که این قوانین با هم تداخل دارند . دستورکار^{۲۳} ترتیب اجرا این تداخل قوانین را با استفاده از استراتژی حل تداخلات^{۲۴} مدیریت می کند.

¹⁷ matching

¹⁸ Matching Pattern

¹⁹ Object Oriented

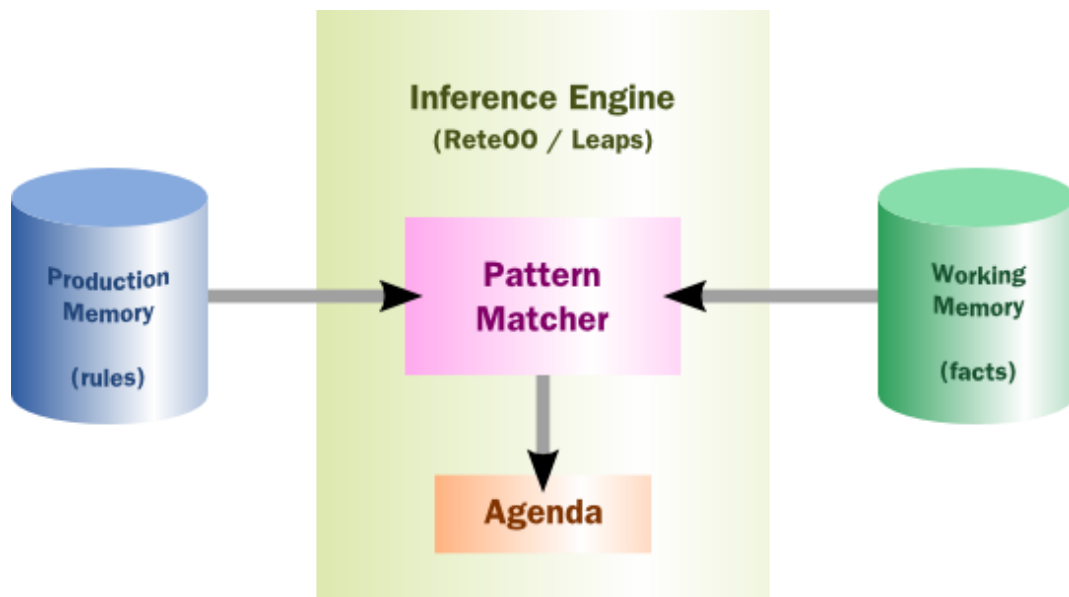
²⁰ Production Memory

²¹ Facts

²² Working Memory

²³ Agenda

²⁴ Conflict Resolution strategy



شکل ۱- نمایش سطح بالایی از موتور قوانین

دو روش اجرایی برای سیستم قوانین وجود دارد: زنجیره ای رو به جلو^{۲۵} و زنجیره ای رو به عقب^{۲۶}. سیستم هایی که ترکیبی از این دو روش استفاده میکنند سیستم های زنجیره ای ترکیبی^{۲۷} گفته می شوند. درک این دو روش عملیات مشخص میکند که چرا یک سیستم قوانین تولید متفاوت است و بهترین فرم بهره وری و استنتاج از قوانین را موجب می شود.

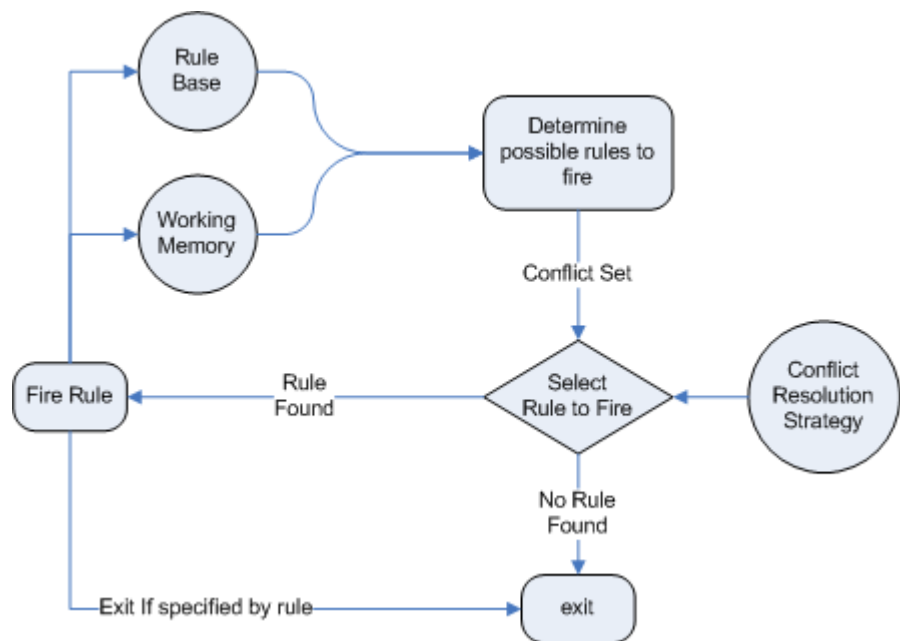
روش زنجیره ای رو به جلو داده-محور^{۲۸} است و در نتیجه به اصطلاح واکنشی است. با وارد شدن حقایق یا داده ها به حافظه کاری، یک یا چند قانون که مطابقت دارند و قسمت شرطی آنها مقدار درست برمی گرداند بر اساس دستورکاری مشخص شده اجرا می شوند. به اختصار، در این روش با یک داده شروع میشود که منتشر شده و با یک نتیجه گیری به اتمام می رسد.

²⁵ Forward Chaining

²⁶ Backward Chaining

²⁷ Hybrid Chaining Systems

²⁸ Data-Driven

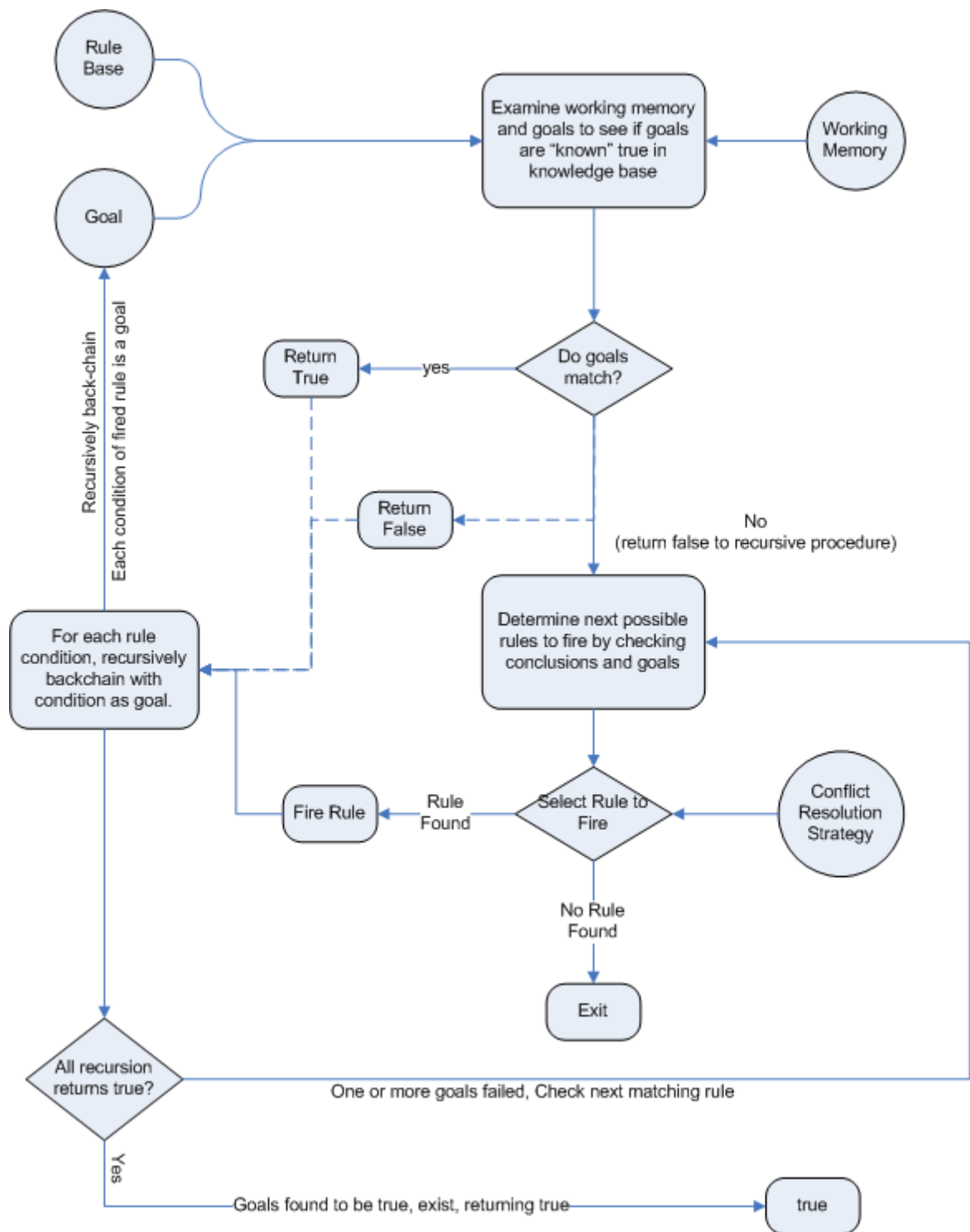


شکل ۲- روش استنتاج زنجیره ای رو به جلو

روش زنجیره‌ای رو به عقب روش به اصطلاح هدف-محور^{۲۹} است. به این معنا که موتور استنتاج کارش را با یک نتیجه شروع می‌کند که هدف رسیدن به آن است. اگر قانونی با مستقیما با این نتیجه ارتباط نداشته باشد. قوانینی جستجو می‌شود که بطور غیر مستقیم به هدف برسند. که اصطلاحا اهداف فرعی^{۳۰} گفته می‌شود. که کمک خواهد کرد تا بعضی از بخش های ناشناخته هدف اولیه ارضا شود. این فرآیند ادامه پیدامی‌کند تا زمانی‌که به هدف اولیه رسیده و یا هیچ قانون فرعی باقی نمانده باشد. زبان Prolog یک نمونه پیاده سازی از الگوریتم زنجیره رو به عقب است.

²⁹ Goal-driven

³⁰ Sub-Goal



شکل ۳- روش استنتاج زنجیره رو به عقب

۱-۵) علل استفاده از موتور قوانین کسب و کار

سوالاتی که غالباً در استفاده از موتور قوانین ایجاد می شوند :

چه زمانی باید از موتور قوانین استفاده کرد ؟

استفاده از موتور قوانین چه مزیت هایی نسبت به کدهای دستی "if ... then" دارد ؟

چرا باید از موتور قوانین به جای چارچوب های اسکریپتی مانند BeanShell استفاده کرد ؟

در زیر سعی می شود که به سوالات بالا پاسخ داده شود.

۱-۵-۱) مزیت های استفاده از موتور قوانین

برنامه نویسی اعلانی^{۳۱}

موتورهای قوانین این امکان را فراهم می کنند که گفته شود "چه کاری باید انجام شود" نه اینکه "چگونه انجام شود".

مهمترین مزیت این نکته این است که استفاده از قوانین، بیان کردن راه حل برای مسائل پیچیده و در نتیجه ارزیابی آن هارا تسهیل می کند.

سیستم های قوانین کسب و کار قادر به حل مسائل بسیار پیچیده می باشند . به همراه توضیحاتی برای اینکه چگونه یک راه حل حاصل شده است و هر تصمیم چگونه گرفته شده .

جدا سازی منطق کسب و کار از داده

³¹ Declarative Programming

داده های برنامه در اشیاء دامنه وجود دارند و منطق برنامه در قوانین . این شکستن ساختاری داده و منطق برنامه با توجه به اینکه از چه منظری نگاه کرده شود ممکن است مزیت و یا عیب باشد . نتیجه این است که اگر منطق برنامه بطور کامل در قوانین پیاده سازی شود قابلیت نگهداری و اعمال تغییرات آینده بر روی منطق برنامه بسیار ساده تر خواهد بود. و این مزیت زمانی بیشتر خودش را نشان می دهد که منطق کسب و کار بین- دامنه‌ای^{۳۲} و یا چند دامنه‌ای^{۳۳} باشد . بجای اینکه منطق کسب و کار بین اشیاء یا کنترلر های چندین دامنه پخش شود می تواند درون فایل های قوانین جداگانه سازمان دهی شود .

سرعت و مقیاس پذیری^{۳۴}

الگوریتم های انطباق که در موتورهای قوانین جهت استنتاج استفاده می شوند راه های بسیار کارایی را جهت پیدا کردن انطباق در داده ها و قوانین ارائه می کنند. این الگوریتم های در حالتیکه مجموعه داده ها بصورت جزیی تغییراتی می کنند و موتورهای قوانین انطباق های گذشته را به یاد دارد و با کارایی بیشتری قوانین مورد نظر پیدا می کنند .

مترکز سازی دانش

با استفاده از قوانین، یک منبعی از دانش ایجاد می شود که قابل اجرا است . این بدین معناست که یک نقطه متمرکز از سیاست های کسب و کار وجود خواهد داشت . و در حالت ایده آل قوانین به قدری خوانا هستند که بتوانند به عنوان مستندات قابل استفاده باشند .

ابزارهای یکپارچگی

ابزارهای متفاوتی جهت توسعه و تغییر قوانین چه در محیط توسعه و چه در محیط اجرا و مبتنی بر وب موجود است . که بازخورد سریعی جهت اعتبار سنجی قوانین فراهم می کند.

³² Cross-domain

³³ Multi-domain

³⁴ Scalability

سهولت شرح دهی و توصیفات اجرا

سیستم های قوانین، مکانیزم هایی را فراهم می کنند که مراحل اجرا و گزینش قوانین بطور کاملاً کارا قابل پیگیری و ثبت باشند .

قوانین قابل فهم

با استفاده از ایجاد مدل های اشیاء و زبان خاص دامنه^{۳۵} که با استفاده از آنها می توان قوانینی نوشت که به زبان طبیعی بسیار نزدیک باشند. با استفاده از این امکانات متخصصین دامنه با کمترین اطلاعات تکنیکی می توانند قوانین را ایجاد کنند و یا در صورت تغییر سیاست ها تغییر دهند .

³⁵ Domain Specific Language

۱-۵-۲) شرایطی که باید از موتورهای قوانین استفاده کرد .

کوتاه‌ترین جواب به این سوال این است که :

" زمانیکه هیچ راه حل برنامه نویسی سنتی رضایت بخشی برای حل مشکل نباشد"

با این جواب کوتاه توضیحات بیشتری مورد نیاز خواهد بود . علت اینکه هیچ راه حلی سنتی وجود

نداشته باشد یکی از موارد زیر است :

- وقتی تعداد قوانین کسب و کار بیش از اندازه باشد.

در شرایطی که ممکن است مسئله پیچیدگی نداشته باشد اما حجم قوانین آنقدر زیاد باشد که نتوان راه حل سنتی مناسبی برای آن ارائه داد .

- مسئله ورای هیچ یک راه حل روشن الگوریتمیک باشد.

در شرایطی که پیچیدگی زیاد است . و هیچ راه حل سنتی روشنی برای آن نباشد و یا اینکه مسئله بطور کامل درک نشده است .

- شرایطی که منطق برنامه تغییر خواهد کرد .

ممکن است خود منطق کسب و کار پیچیدگی نداشته باشد و ساده باشد اما منطق کسب و کار قرار است در طول عمر نرم‌افزار تغییر کند . در بسیاری از سازمانها ، انتشار نسخه های جدید نرم افزار ها به تعداد کم و در فواصل زمانی زیاد انجام می‌شود و قابلیت تغییر آسان قوانین کسب و کار بسیار مورد نیاز خواهد بود.

- متخصصان کسب و کار آماده به کار هستند ولی فنی نیستند.

متخصصان دامنه اغلب مالک ارزش دانش مربوط به قوانین کسب و کار فرآیندها هستند . آنها بطور عمومی غیرفنی هستند، اما می‌توانند بسیار منطقی باشند . قوانین به آنها اجازه می‌دهند که منطق کسب و کار را به روش خود اظهار کنند . خیلی از افراد در جایگاه غیر فنی، معمولا آموزش منطق رسمی ندیده اند ، و باید با کد کردن کسب و کار دانش در قوانین، این خلاء پر شود.

اگر قوانین تکنولوژی جدیدی برای تیم پروژه باشند، سر بار راه اندازی و آشنایی با این تکنولوژی باید در نظر گرفته شود .

بطور خاص در برنامه های کاربردی شی گرا مدرن، از موتور جستجو به عنوان بخشی از منطق کسب و کار در نظر گرفته می شود. مخصوصا بخشی که بسیار پیچیده است . این در حقیقت وارونگی مفهوم شی گرایی کپسوله کردن همه منطق درون اشیاء است.

اگر در پروژه بخش هایی باشند که با عبارتهای شرطی تودرتو و پیچیده پیاده سازی شده باشد آن بخش دقیقا بخش هایی است که باید توسط قوانین پیاده سازی شوند . و اگر بخش هایی از کد وجود دارد که برنامه نویسان پی در پی در حال تغییر آنها هستند که ممکن است به دلیل اشتباهات یا تغییرات پیشنهاد شده توسط طراحان باشد آن قسمت دقیقا جایی است که قوانین و موتور قوانین میتواند نقش موثری داشته باشد و می توانید به آن فکر کنید.

قوانین می‌توانند درون برنامه کاربردی جاسازی شوند^{۳۶} و یا اینکه بعنوان سرویس استفاده شود . که البته برای بهره بردن از ویژگی حافظه دار^{۳۷} موتور قوانین باید از ماژول موتور قوانین به عنوان بخشی از برنامه کاربردی استفاده کرد که در فصل های بعد ذکر خواهد شد.

برای سازمانی که قرار است از موتور قوانین استفاده شود . مهم است که تصمیم بگیرد که فرآیند به روز رسانی و یا اضافه کردن قوانین چه فرآیندی استفاده خواهد شد . گزینه های زیادی برای این کار وجود دارد،

³⁶ Embedded

³⁷ Stateful

اما سازمان های متفاوت نیازمندیهای متفاوتی دارند . هرچند که تغییر کردن قوانین و سیاست های سازمان برای مالکین یا توسعه دهندگان پروژه قابل کنترل نیست .

۱-۶) شرایطی که نباید از موتور قوانین استفاده کرد

باید توجه داشت که نباید فراموش کرد که یک موتورقوانین فقط بخشی از یک پروژه پیچیده و یا راه حل است. موتورهای قوانین قرار نیست که گردش کار یا فرآیند اجرایی را به دست بگیرند و همچنین موتورهای گردش کار و یا ابزارهای مدیریت فرآیند برای انجام قوانین کسب و کار طراحی نشده اند و باید از ابزار مناسب برای هر کار استفاده کرد.

همانطور که موتورهای قوانین پویا هستند (پویا به این مفهوم که قوانین می توانند به عنوان داده ذخیره و مدیریت شوند)، اغلب به آنها به عنوان راه حلی برای مسئله توسعه نرم افزار دیده می شوند . اگر به این علت است که از موتور قوانین استفاده می شود، باید آگاه بود که موتورقوانین زمانی بهترین استفاده را دارند که از خاصیت اعلانی بودن قوانین استفاده کنید . به عنوان یک جایگزین، می توان از طراحی داده- محور یا موتورهای فرآیند اسکریپت بطوریکه اسکریپت ها در پایگاه داده مدیریت شده اند و قابلیت به روز رسانی را دارند .

۷-۱) ویژگی های سیستم های مدیریت قوانین کسب و کار

با توجه به تعریف کوتاهی که توسط Tony Morgan بیان شده است. "قانون کسب و کار یک عبارت فشرده درباره جنبه ای از کسب و کار است که می تواند بگونه ای ساده و غیر مبهم بیان شود که توسط همه طرفین کسب و کار از جمله صاحبین ، تحلیلگر و طراحان و توسعه دهندگان کسب و کار قابل دسترسی باشد و این قوانین در یک نظام تعبیه شده اند تا با هم در ارتباط باشند".

عبارت $x = y - z$ یک قانون نیست . این یک عبارت انتساب است و یا به عبارت دیگر یک رویه^{۳۸} محاسبه x است. یک قانون در سیستم مدیریت قوانین کسب و کار غیر - رویه ای^{۳۹} است . آنها مشخص می کنند که چه چیزی درست است نه اینکه چگونه محاسبه شوند . همچنین برقراری ارتباط بین قوانین را استنتاج^{۴۰} می گویند . یک سیستم مدیریت قوانین علاوه بر پشتیبانی از حقایق و قوانین باید از استنتاج خودکار پشتیبانی کند .

در این بخش مروری بر ویژگی های تکنیکی سیستم های مدیریت قوانین کسب و کار خواهیم داشت. این ویژگی ها شامل موارد زیر می شود :

- معماری
- یکپارچه سازی با برنامه کاربردی تجاری
- بازنمایی دانش
- استراتژی های استنتاج
- تبدیل کردن قوانین به مجموعه قوانین مستقل و قابل اتصال
- وضعیت ویژگی هایی که در سیستم های مدیریت قوانین کنونی دیده می شوند .

³⁸ Procedural

³⁹ Non-procedural

⁴⁰ Interfere

- ایجاد برنامه های کاربردی هوشمند که بتواند با استفاده از گفتگو منطقی و قابل فهم با کاربر ارتباط برقرار کند .
- دادن امکان ایجاد و تغییر و مدیریت قوانین کسب و کار به کاربران و طراحان سیستم .
- مدیریت نسخه^{۴۱} و مخازن^{۴۲} قوانین.

۱-۷-۱) کامپوننت ها و ویژگی های فنی سیستم مدیریت قوانین کسب و کار (BRMS^{۴۳})

سیستم های مدیریت قوانین زیرمجموعه سیستم های خبره ای هستند که از سال ۱۹۸۰ تولید شده اند . سیستم های قانون محور یا خبره که گاهی اوقات سیستم های دانش – محور نیز گفته می شوند . سیستم های کامپیوتری هستند که می توانند پیشنهاد دهند و یا تصمیم گیری کنند. دونه از این سیستم ها وجود دارد : سیستم هایی که تصمیم گیری میکنند که بطور کلی کنترلر ها و برنامه های کاربردی مانند سیستم- های مالی ، سیستم های تجاری و سیستم هایی که به عنوان سیستم های پشتیبانی از تصمیم گیری عمل میکنند، آنها پیشنهاد می دهند اما تصمیم نمیگیرند . این تعریف نشان دهنده بخش از کارهایی است که سیستم های خبره انجام میدهند . مهم تر اینکه، سیستم های قانون- محور با توجه اینکه چه معماری دارند تعریف می شوند . مهمترین ویژگی معماری اینگونه سیستم ها این است که دانش مربوط به مسئله جدا از کدی که دانش را بر مسئله در دست اعمال میکند، نگهداری میشود.

مخزن دانش در یک سیستم مدیریت قوانین بعنوان پایگاه قوانین^{۴۴} یا پایگاه دانش^{۴۵} رجوع می شود و مکانیزمی که دانش را بر داده های ارائه شده اعمال می کند موتور قوانین یا موتور استنتاج^{۴۶} گفته می شود.

⁴¹ Versioning

⁴² Repository

⁴³ Business Rule Management System

⁴⁴ Rule Base

⁴⁵ Knowledge Base

⁴⁶ Inference Engine



شکل ۴- معماری سطح بالا سیستم مدیریت قوانین کسب و کار

اکنون بطور گسترده پذیرفته شده است که سیستم مدیریت قوانین از چهار بخش اصلی تقسیم شده است . بخش اول، محیط اساسی از سمبل و تغییر مقادیر که همه سیستم های کامپیوتری به اشتراک می گذارند که میتواند به عنوان زبان برنامه نویسی در نظر گرفته شود . مانند ویرایشگرها و کامپایلرها که بخش خاکستری شکل بالا این بخش را نشان می دهد .

بخش دوم، ساختار پایگاه دانش است که شامل روش های بازنمایی و دسترسی به دانش نیز می شود و نهایتا باید تکنیک هایی برای اعمال دانش برای مسئله در دست وجود داشته باشد .

بخش سوم، موتور استنتاج است که قوانین را به هم زنجیر میکند تا به نتیجه معتبر برسد .

بخش چهارم، مخزنی است که در آن قوانین نگهداری می شوند و قابل ویرایش و مدیریت نسخه است .

پایگاه دانش و موتور استنتاج از یکدیگر جدا در نظر گرفته شده اند تا نگهداری را تسهیل بخشند .

علاوه بر این در بیشتر موارد قوانین و سیاست ها به مرور زمان تغییر میکنند در حالیکه موتور استنتاج قرار نیست بازنویسی شوند .

پایگاه دانش معمولا شامل انواع مختلف دانش می شود : دانش مربوط به اشیاء، روندها و ارتباطات .

دانش در رابطه با اشیاء معمولا در فرم مدل اشیاء ، شمای XML ، مدل داده و شبکه معنایی ذخیره می شود . دانش روند ممکن است بصورت متدهای جاوا یا ماکروهای اکسل بازنمایی شوند . بعضی از روندهای

کسب و کار نیز توسط قوانین بازنمایی می شوند.

۱-۷-۲) قوانین

دانش مربوط به ارتباطات معمولاً بصورت قوانین به فرم "IF A THEN B" نگهداری می شوند . برخلاف عبارت های if/then که در زبان های رسمی مانند جاوا پیدا می شوند . زبان های قوانین عموماً اعلانی^{۴۷} با غیر-رویه ای^{۴۸} هستند به این معنا که ترتیبی که قوانین نوشته می شوند در نتیجه نهایی تأثیری ندارد . این قوانین بر روی دانش موجودیت ها و اشیاء کار میکنند . همانجور که گفته شد . راه دیگر بازنمایی دانش رویه ها هستند . مانند رویه هایی که در زبان های مرسوم وجود دارند . راه های دیگری نیز برای بازنمایی دانش وجود دارد اما قوانین ، رویه ها و اشیا مهمترین آنها هستند .

۱-۷-۳) قالب های قوانین^{۴۹}

قالب های قوانین الگوهای طراحی برای قوانین هستند . در بسیاری از شرایط یک قانون ممکن است برای داده های مختلف قابل اعمال باشند . در اینچنین مواردی ، قالب های قوانین قابلیت ایجاد قوانین با یک جایگاه خالی که قرار است بعداً پر شود را میدهد . یک قالب قانون کسب و کار یک قانون کسب و کار تعریف شده نسبی را که شامل جانما^{۵۰}ها برای داده هایی است که بعداً پر می شود . در اینصورت با پرکردن جاهای خالی با مقادیر جدید قوانین جدید تولید خواهد شد .

۱-۷-۴) چک کردن نحوی قوانین^{۵۱}

یک موتور قوانین خوب امکاناتی را برای بررسی بی وقفه نحو قوانین ، همزمان با نوشتن قوانین فراهم میکند . با زبان های قوانین ساخت یافته ، بسیار مفید است که ابزار مناسبی جهت چک کردن صحت نحوی قوانین ، نمایش کلمات کلیدی و متغیرها با رنگ های متفاوت وجود داشته باشد .

⁴⁷ Declarative

⁴⁸ Non-procedural

⁴⁹ Rule Template

⁵⁰ Place holder

⁵¹ Rule Syntax Checking

(۵-۷-۱) رویه ها و الگوریتم ها

بعضی دانش ها از نوع رویه ای هستند . برای مثال قانون محاسبه مالیات . بازنمایی دانش زمانیکه قرار است در رویه ها نگهداری شوند، می تواند بسیار سنگین باشد. یک سیستم مدیریت قوانین خوب به مجموعه قوانین امکان استفاده از رویه های را میدهد. بطوریکه در اجرای قوانین قابلیت اجرا و برگرداندن قوانین را داشته باشد.

(۶-۷-۱) جریان قوانین^{۵۲}

مکانیزم جریان قوانین باید در سیستم مدیریت قوانین وجود داشته باشد که به طراحان این اجازه را میدهد که ماژول ها وقوانین با ترتیب یا اولویت مشخصی اجرا شوند . این ویژگی برای یک سیستم مدیریت قوانین خوب الزامی است .

(۷-۷-۱) جداول و درخت تصمیم گیری^{۵۳}

در بعضی از محصولات، بازنمایی جایگزینی برای بازنمایی قوانین بصورت if/then وجود دارد . و این دو نوع بازنمایی یعنی درخت تصمیم گیری و جدول تصمیم گیری در نظر گرفته شده است . درختان تصمیم گیری قوانین را بصورت ساختار درختی نمایش می دهد . این نوع بازنمایی برای عیب یابی و ارتباط بین کاربران و برنامه نویسان و طراحان کسب و کار مفید است. جداول تصمیم گیری دانش مشابهی را بصورت ساختار جدولی نمایش می دهند .

مشکل اصلی جداول تصمیم گیری این است که معمولا در حالتی که قوانین زیاد باشند بصورت غیر مدیریت شده ای رشد میکند . در این نوع بازنمایی هر ردیف جدول یک قانون را معرفی می کنند که خوانایی کمی دارد.

⁵² Rule flows

⁵³ Decision Tables and decision Tree

مهمترین مزیت جداول تصمیم‌گیری برای حالتی است که این قوانین از قبل در سازمان بصورت جدولی وجود داشته باشد. مانند استفاده از جدول قیمت یا جدول نرخ سود بعنوان بخشی از قوانین کسب و کار که در این حالت بهتر است از این داده‌های جدولی بعنوان داده‌های قوانین استفاده شود.

۸-۷-۱) استنتاج

یک موتور استنتاج یک یا چند ابزار را برای اعمال دانش بر روی داده پیشنهاد میکند. مهمترین استراتژی‌ها زنجیره رو جلو و زنجیره رو به عقب نام دارند.

روش زنجیره‌ای رو به جلو داده-محور^{۵۴} است و در نتیجه به اصطلاح واکنشی است. با وارد شدن حقایق یا داده‌ها به حافظه کاری، یک یا چند قانون که مطابقت دارند و قسمت شرطی آنها مقدار درست برمی‌گرداند بر اساس دستورکاری مشخص شده اجرا می‌شوند. به اختصار، در این روش با یک داده شروع میشود که منتشر شده و با یک نتیجه‌گیری به اتمام می‌رسد.

روش زنجیره‌ای رو به عقب روش به اصطلاح هدف-محور^{۵۵} است. به این معنا که موتور استنتاج کارش را با یک نتیجه شروع می‌کند که هدف رسیدن به آن است. اگر قانونی با مستقیما با این نتیجه ارتباط نداشته باشد. قوانینی جستجو می‌شود که بطور غیر مستقیم به هدف برسند. که اصطلاحا اهداف فرعی^{۵۶} گفته می‌شود. که کمک خواهد کرد تا بعضی از بخش‌های ناشناخته هدف اولیه ارضا شود. این فرآیند ادامه پیدامی‌کند تا زمانی که به هدف اولیه رسیده و یا هیچ قانون فرعی باقی نمانده باشد. زبان Prolog یک نمونه پیاده‌سازی از الگوریتم زنجیره رو به عقب است.

⁵⁴ Data-Driven

⁵⁵ Goal-driven

⁵⁶ Sub-Goal

دو ویژگی مهمی که سیستم های قانون محور را از دیگر سیستم های کامپیوتری متمایز میکند، این است که آنها اغلب میتوانند :

- شرح چگونگی نتیجه گیری را فراهم کنند .
 - ترکیب استدلال کیفی یا قضاوت و مدیریت عدم قطعیت.
- اگر این دو ویژگی با هم وجود داشته باشند، سیستم های قانون محور ممکن است بیش یک نتیجه همراه امتیاز دهی هر نتیجه پیشنهاد دهند . که البته در صورت نیاز این دو ویژگی اعمال آنها در سیستم باعث هزینه و پیچیدگی بیشتر سیستم می شود .

تسهیلات تعبیه شده شرح دهی ابزار مناسبی جهت رفع باگ و بررسی مسیر تصمیم گیری ارائه می کنند اما به ندرت برای کاربر نهایی قابل توجه است . امکانات مفید برای شرح دهی در استنتاج برای کاربران باید دست ساز باشند.

چندین تکنیک برای مدیریت عدم قطعیت که بسیار رایج است وجود دارد:

- استدلال صریح با استفاده از برچسب کلامی برای شرایط نامشخص
- امکانات What-if
- سیستم های تعمیر و نگهداری حقیقت
- عامل اطمینان
- احتمال Bayesian
- مجموعه های فازی

استدلال در مورد عدم قطعیت به پیچیدگی یک سیستم می افزاید و کسب دانش در ارتباط با آن است، اما اجازه مقابله با مشکلات پیچیده تر را به ما میدهد.

⁵⁷ Uncertainty

⁵⁸ Explanation

فصل دوم

بررسی سیستم های مدیریت قوانین کسب و کار موجود

۲) بررسی سیستم های مدیریت قوانین کسب و کار موجود

۱-۲) مقدمه

تولیدات زیادی وجود دارند که به کاربران اجازه می دهند تا سیستم های مبتنی بر قانون را توسعه دهند. البته ممکن است همه آنها به عنوان سیستم مدیریت قوانین کسب و کار دسته بندی نشوند. در این بخش فاکتورهایی در نظر گرفته می شود که تعداد زیادی از آنها را از لیست کاندید حذف شوند. شروطی که اعمال شده اند که لیست باند سیستم های کاندید کوتاه شوند به اختصار در زیر آمده است.

- ✓ باید اجازه دهند تا تحلیلگران کسب و کار قوانین را ایجاد کنند و تغییر دهند.
- ✓ استفاده از یک مخزن دانش با ویژگی های کامل.
- ✓ مبتنی بر الگوریتم rete باشد.
- ✓ پشتیبانی از روش زنجیره عقب.
- ✓ باید امکان استفاده از موتور قوانین به عنوان یک جزء و یا سرویس در برنامه های کاربردی بزرگتر وجود داشته باشد.
- ✓ باید اجازه دهد تا برنامه های کاربردی در معماری سرویس گرا مستقر می شود.
- ✓ تمرکز بر مسائل مدیریت قوانین کسب و کار.
- ✓ مدارک و شواهد از برنامه های کاربردی تجاری موفق استفاده از آن را فراهم کند.
- ✓ منبع باز بودن سیستم مدیریت قوانین.

ملاحظات ذکر شده باعث حذف زیادی از سیستم های کاندید استفاده می شود. در ادامه این فصل

لیست کامل سیستم های کاندید معرفی می شوند و مورد مطالعه و بررسی قرار می گیرند.

۲-۲) معرفی سیستم های مدیریت قوانین موجود

۲-۲-۱) Drools

Drools یک سیستم مدیریت قوانین کسب و کار توسعه یافته است که بر اساس الگوریتم Charles Forgy's Rete پیاده سازی شده است و برای JVM^{۵۹} مناسب است. Drools امکان برنامه نویسی را تامین میکند و برای تطبیق مسئله دامنه کسب و کار با زبان های خاص، ابزارهای ویرایش گرافیکی، ابزارهای مبتنی بر وب و ابزارهای بهره وری طراحان فراهم کرده است که بسیار انعطاف پذیر است. Drools یک مجوز کسب و کار متن باز دارد که اجازه دانلود، استفاده، تعبیه و توزیع را بطور رایگان را ارائه میکند. به کاربران اجازه میدهد که بیان کنند چه انجام شود و به چگونگی انجام آن کاری ندارد. بصورتی که هم برای کاربران و هم برای طراحان قابل فهم باشد. زبان های خاص دامنه^{۶۰} به طراحان اجازه میدهد سمتیک زبان طبیعی را به عنوان قاعده بنویسند. به این ترتیب فاصله بین توسعه دهندگان و طراحان غیر فنی را کاهش میدهد. در آخرین نسخه مجموعه ابزار Drools محیطی گرافیکی به نام Kie-Workbench^{۶۱} توسعه داده شده که محیطی برای استفاده از کلیه امکانات این ابزار فراهم میکند.

گروه توسعه Drools افزونه هایی^{۶۲} را جهت تسهیل توسعه دهندگان در محیط توسعه Eclipse توسعه داده اند که نوشته قوانین را در محیط توسعه و اعتبار سنجی و تست آنها آسان تر کرده است.

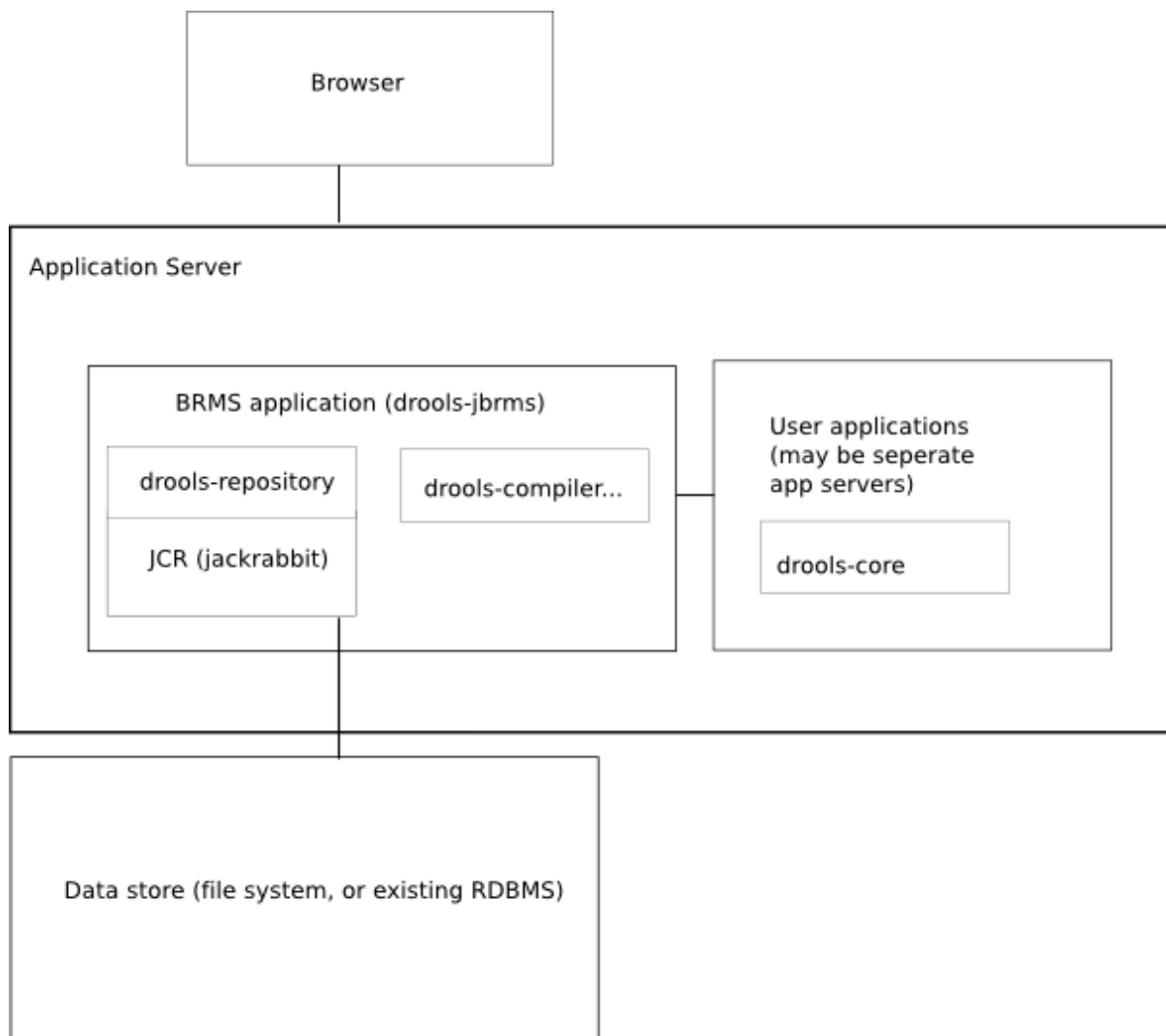
معماری سیستم Drools و چگونگی ارتباط با برنامه کاربردی در شکل زیر نمایش داده شده است :

⁵⁹ Java Virtual Machine

⁶⁰ Domain Specific Language

⁶¹ KIE : stands for " Knowledge Is Everything"

⁶² Plugins



شکل ۵- معماری سیستم مدیریت قوانین Drools

Jess (۲-۲-۲)

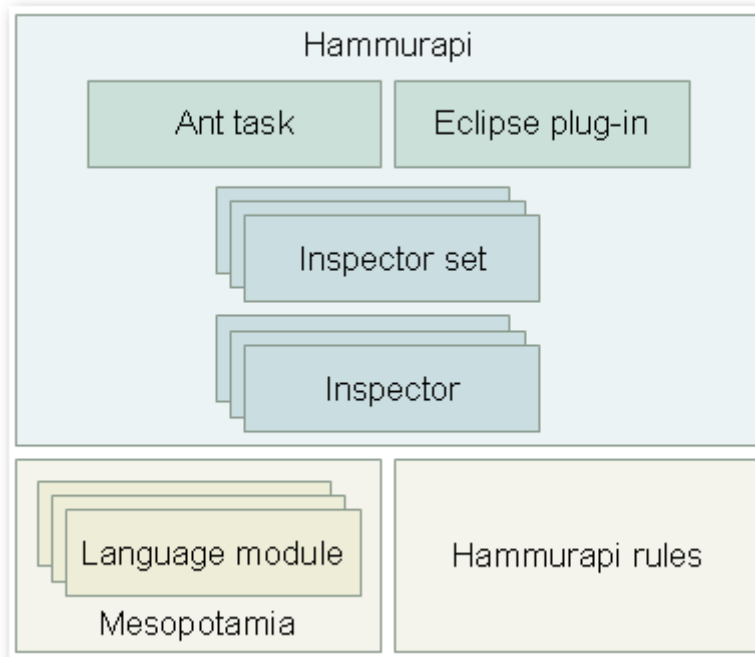
Jess یک موتور قوانین و یک محیط اسکریپت است که توسط Ernest Friedman در آزمایشگاه های Sandia National در Livermore به زبان جاوا نوشته شده است . با استفاده از Jess می توان یک نرم افزار جاوا توسعه داد که با استفاده از اطلاعاتی که در فرم های قوانین وجود دارد، توانایی استدلال داشته باشد. Jess کوچک، سبک و یکی از سریعترین موتور قواعد موجود است . زبان اسکریپتی و قدرتمند آن، دسترسی به همه API های جاوا را ممکن می سازد. شامل یک محیط توسعه یافته با تمام ویژگی های سکوی Eclips است . از یک نسخه پیشرفته الگوریتم Rete برای قوانین فرآیند استفاده میکند. Rete یک مکانیزم کارا برای

حل پیچیده تطبیق است. همچنین از متدهای زنجیره رو به عقب و زنجیره رو به جلو پشتیبانی می کند . در رابطه با اشیاء جاوا، میتواند مستقیما بطور دستی عمل کرده و استدلال کند . یک محیط اسکریپت جاوا دارد که می توان در آن اشیاء جاوایی ایجاد کرد . متدهای جاوا را فراخوانی کرد و واسط های جاوا را بدون کامپایل کرهای جاوا پیاده سازی کرد. برای استفاده های تجاری تایید شده استو برای استفاده های آموزشی بدون هزینه در دسترس است .

Hammurapi Rules (۳-۲-۲)

Hammurapi Rules یک موتور قواعد برای توسعه دهندگان جاوا است . قوانین در آن با زبان جاوا طراحی شده اند. از متد زنجیره رو به جلو وروبه عقب برای استنتاج پشتیبانی می کند . طراحی که با زبان جاوا آشنایی دارند لازم نیست برای نوشتن قوانین زبان خاص دیگری یاد بگیرند . . همه اطلاعات برای توسعه جاوا با یکریگر تلفیق می شوند و می توانند برای توسعه قوانین به کار روند. به ابزارها و فرآیند های خاصی نیازی نیست . قوانین منطقی نیز مانند دیگر کلاس های جاوا به JVM و منابع سازمان دسترسی دارند . قوانین بوسیله کامپایلرهای جاوا بر حسب bytecode کامپایل می شوند و JVM بهینه سازی قوانین را در زمان اجرا تامین میکند . اجرای موازی قوانین نیز امکان پذیر است . بطور کلی ویژگی های زیر را دارد :

- پشتیبانی از زنجیره رو به جلو از طریق واسط های JSR-94
- پشتیبانی از زنجیره رو به عقب در مواقعی که از زنجیره رو به جلو کارایی بیشتری دارد .
- استفاده از کد جاوا ساده برای نوشتن قوانین .
- قوانین با استفاده از ساختار XML پیکربندی و اسمبل می شوند .
- از واسط چند نخه پشتیبانی می کند .



شکل ۶ - معماری کامپوننت های Hammurapi Rules

InfoSapient (۴-۲-۲)

InfoSapient برنامه ای برای بیان آسان اجرا و نگهداری از قوانین کسب و کار در یک شرکت است و کاملاً بر اساس استاندارد جاوا نوشته شده است . یک موتور قوانین است که به اعضای انجمن کسب و کار اجازه میدهد که قوانین کسب و کاری که روزانه استفاده می کنند، ثبت کرده و نگهداری کنند. بسیاری از متغیر ها بطور همزمان برای سنجیدن ریسک ها و فرصت ها برای رسیدن به بهترین حالت در نظر گرفته می شوند . InfoSapient موقعیت های مبهمی را که در فرآیندهای کسب و کار ایجاد می شوند به بهترین حالت راه اندازی می کند . بدون نیاز به مهارت و تجربه ی نیروی انسانی تصمیمات پیچیده و به موقع گرفته می شود. با استفاده از apache Tomcat و IBM WebSphere با موفقیت آزموده شده است .

۵-۲-۲ JLisa

JLisa یک بستر قوی برای قوانین کسب و کار قابل دسترس به جاوا بوده و با JSR-94 سازگار است . از Clips قدرتمندتر است زیرا مزایای بسط یافته از ویژگی های عمومی Lisp را دارد . این ویژگی ها برای نرم افزار توسعه یافته multi-paradigm ضروری هستند . JLisa بر اسا ABL1 و Lisa است و قابلیت های آن عبارتند از :

- اجرای قوانین Clips-like روی اشیاء جاوا
- ساخت سیستم های Java agent که قابلیت استدلالی دارد .
- ساخت سیستم های جاوا با مشارکت اشیاء جاوا
- ساخت GPS در جاوا
- اجرای ارزیابی قانون کسب و کار
- ایجاد و اجرای درخت تصمیم برای اشیاء جاوا
- اجرای جستجوها و پرسش و پاسخ های منطقی روی اشیاء جاوا
- ایجاد smart Wizard و غیره...

۶-۲-۲ Mandarax

Mandarax یک موتور قوانین است که کاملاً بر اساس جاوا پیاده سازی شده است . انواع قوانین بر اساس پایگاه داده ها، reflection ، EJB و استانداردهای XML پشتیبانی می شوند . با استفاده از زنجیره رو به عقب یک موتور استنتاج J2EE را تامین میکند .

OpenLexicon (۷-۲-۲)

OpenLexicon یک ابزار مدیریت فرآیند و قوانین کسب و کار است که برنامه های کاربردی را برای تراکنش برنامه های مبتنی بر فرآیند توسعه می دهد . برای مدیریت و آزمون قوانین کسب و کار، شامل یک انبار فراداده، یک موتور قوانین کسب و کار و یک واسط کاربری جامع مبتنی بر وب است .

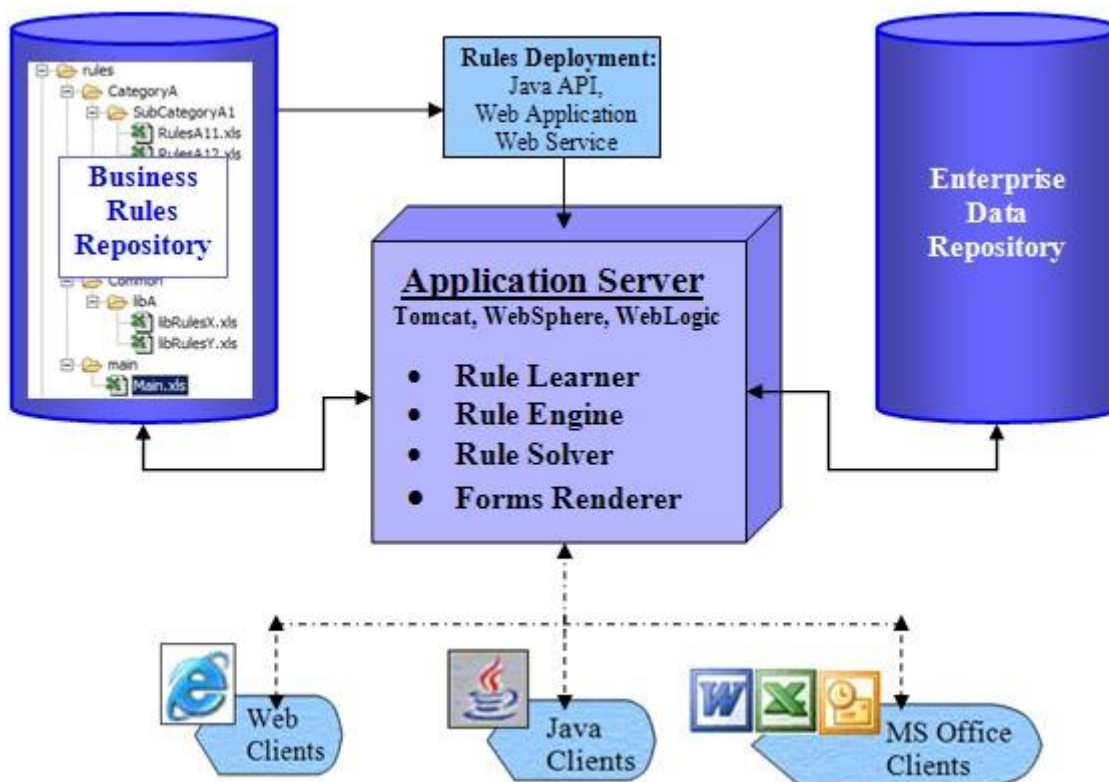
برای هماهنگ سازی تعاملات پیچیده در قوانین کسب و کار و اشیاء کسب و کار دارای ابزارهای مدیریت فرآیند ها می باشد. شهرت Openlexicon به علت تامین راه کارهایی با کارایی بالاست که در مجموعه ای از برنامه های کاربردی در سطح سازمانی استفاده می شود. دارای دو بخش اصلی است که عبارتند از انبار فراداده و موتور قوانین کسب و کار .

بخش مهم openLixicon به عنوان نرم افزار متن باز منتشر شده است . علاوه بر آن یک ویزارد دارد که یک ازار برای ساخت قوانین و موردهای کاربری در کسب و کار است . ویزارد برای طراحان و تحلیلگرانی طراحی شده است که از مهارت تکنیکی بالایی برخوردار نیستند .

OpenRules (۸-۲-۲)

openRules یک متدولوژی ، ابزارهای متن باز و سرویس هایی را ارائه کرده است که تحلیلگران کسب و کار را قادر می سازد که با طراحان نرم افزار برای ایجاد و پشتیبانی از برنامه های کاربردی مبتنی بر قاعده در کسب و کارهای پیچیده و ارائه منطقی همکاری کنند . OpenRules یک سیستم مدیریت قوانین کسب و کار کامل است . بخش های ساده و قدرتمند متن باز را برای برنامه های کاربردی توسعه یافته با استفاده از قوانین تامین می کند . با استفاده از ابزارهایی مثل اکسل ، Google spreadsheets و Eclips IDE می توان تصمیمات پیچیده ای را برای پشتیبانی از سیستم های در محدوده یک سازمان ایجاد کرد .

openRules تنها برای استفاده از آموزشی و بررسی لیسانس محدود دارد.



شکل ۷ - معماری سیستم مدیریت قوانین OpenRules

۹-۲-۲ Jamacho (morendo)

Jamocha یک موتور قوانین متن باز است که هدف آن تامین یک موتور قواعد با کیفیت بالا و یک محیط system shell خبره است. به علاوه ابزارهای توسعه یافته ی مورد نیاز و بهترین متدولوژی عملی را برای برنامه های کاربردی مبتنی بر قوانین تامین میکند .

۱۰-۲-۲ JRuleEngine

JRuleEngine یک موتور قواعد جاوا است و بر اساس خصوصیات Request94 نسخه ی ۱/۱ است. قوانین می تواند به وسیله ی یک فایل xml یا JRuleEngine APIs بارگذاری شود و در یک پایگاه داده ذخیره شوند. این توزیع شامل یک کتابخانه است که می تواند در هر نوع برنامه ی کاربردی جاوا ادغام شود، در نتیجه می تواند در هر نوع برنامه ی کاربردی (بر اساس وب یا غیره) استفاده شوند.

JRuleEngine در وظیفه مندی و کارایی ضعیفتر از دیگر موتورهای قاعده عمل می کند. در مورد دسترسی به راهنمای کاربری این موتور می توان وضعیت خیلی ضعیف را در نظر گرفت. نقطه قوت این ابزار متن باز بودن و استفاده از ابزارهای مبتنی بر جاوا می باشد.

۲-۳) مقایسه موتور قوانین کسب و کار

موتور های قوانین نام برده در گزارشی با عنوان "گزارش مطالعه و اولویت بندی ابزارهای BPMS"

که توسط مرکز تحقیقات صنایع انفورماتیک انجام شده است استخراج شده است .

مجموع امتیازات	لیسانس نرم افزاری برای ابزار مورد استفاده	دورنمای توسعه ابزار مورد استفاده	موجود بودن تیم پشتیبانی قوی برای ابزار مورد نظر	میزان استفاده سایرین از ابزار مورد نظر	به روز بودن ابزار مورد استفاده	پشتیبانی از زبان های هماهنگ سازی سرویس ها	پشتیبانی از سرویس های مختلف در گذرگاه سرویس	پشتیبانی از استانداردهای مرتبط	استفاد از ابزارهای مبتنی بر جاوا	دسترسی به راهنمای کاربری ابزار مورد استفاده	کاربر پسند بودن ابزار مورد استفاده	من بن بودن ابزار های مورد استفاده	سازگاری ابزارهای مورد استفاده با یکدیگر	کارایی ابزارهای مورد استفاده	وظیفه مندی ابزارهای مورد استفاده	
۵۶	۴	۴	۴	۴	۴	۰	۰	۵	۵	۴	۴	۵	۲	۵	۵	Drools
۴۹	۲	۲	۲	۲	۴	۰	۰	۵	۵	۲	۲	۵	۲	۵	۵	Jamocha
۴۹	۲	۲	۲	۲	۴	۰	۰	۵	۵	۲	۲	۵	۲	۵	۵	Jess
۴۲	۲	۲	۲	۲	۴	۰	۰	۲	۵	۲	۲	۵	۲	۲	۲	Hammurapi
۳۰	۲	۱	۱	۱	۱	۰	۰	۱	۵	۲	۲	۵	۲	۳	۲	InfoSapient
۴۰	۲	۲	۲	۲	۲	۰	۰	۲	۵	۲	۲	۵	۲	۳	۲	JLisa
۳۴	۲	۲	۲	۲	۲	۰	۰	۱	۵	۲	۲	۵	۲	۳	۲	Mandarax
۳۱	۲	۲	۲	۲	۲	۰	۰	۱	۵	۲	۲	۵	۱	۲	۲	OpenLexicon
۳۵	۲	۲	۲	۲	۵	۰	۰	۱	۵	۲	۲	۵	۱	۲	۲	Open Rules

۲۳	۳	۱	۱	۱	۱	۰	۰	۱	۵	۱	۱	۵	۱	۱	۱	Jrule Engine
----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--------------

بر اساس مقایسات انجام شده ، استفاده از سیستم مدیریت قوانین Drools پیشنهاد می شود که ابزاری متن باز و رایگان به زبان جاوا است که قابلیت مجتمع سازی بصورت تعبیه شده و مبتنی بر سرویس را دارد . همچنین ابزارهای تحت وب مناسبی برای ایجاد و دستکاری قوانین فراهم میکند .

در ادامه چگونگی استفاده و مجتمع سازی این سیستم با برنامه کاربردی تجاری مورد نظر شرح داده خواهد شد .