



ارائه شده توسط:

سایت ترجمه فا

مرجع جدیدترین مقالات ترجمه شده

از نشریات معتبر

سیستم چندعاملی برای کنترل دنباله راه اندازی توربین گاز صنعتی

چکیده

این مقاله به معرفی تکنیک جدیدی برای کنترل شرایط دنباله های راه اندازی توربین گاز می پردازد. میزان وسیعی از داده ها و فرایندهای پیچیده مربوط به آشکار سازی خطای آنلاین، نیاز به راه حل خودکار را نشان می دهند. سیستم چند عاملی که مسئله را همچون تعامل نهاد نرم افزاری مستقل می بیند، برای استفاده موثر از داده های قابل دسترس ارائه شده است. راه حل کلی از ترکیب راه حل های جزئی ارائه شده توسط مولفه های سیستم چندعاملی استخراج می شود. در نتیجه، تفسیر داده ها از طریق تبدیل داده ها به اطلاعات مناسب و ترکیب اطلاعات عوامل فردی بدست آمده و به تشخیص خطای اتوماتیک برای مهندسين منجر می شود. این سیستم چندعاملی می تواند چندین تکنیک سیستم هوشمند را بکار گرفته و با استفاده از **ZEUS Agent Building Toolkit** اجرا شود.

واژه های کلیدی: کنترل شرایط، عوامل هوشمند

1. مقدمه

براساس تعریف، کنترل شرایط زمانی اجرا می شود که ارزیابی وضعیت کارخانه و تعیین اینکه آیا از طریق استدلال و مشاهده، عملکرد بدی دارد یا نه، ضروری است. تشخیص اولیه و آشکار سازی خطا، فعالیت مهمی برای به حداکثر رسانی اقتصاد کارخانه، هزینه های عملیاتی و سطوح ایمنی آن می باشد. مهندسين به معرفی پشتیبانی از تصمیم بهتر برای روشهای کنترل شرایط پیچیده از طریق اعمال سیستم های هوشمند مرکزی و استفاده از انواع تکنیک های هوش مصنوعی **AI** پرداخته اند. اکنون معلوم است که بر مسائل ناشی از پیچیدگی عملکردی کنترل شرایط، می توان با معماری غلبه کرد که شامل چندین ماژول توزیعی هوشمند - که بطور دینامیک تعامل دارند- با نام عوامل هوشمند می باشد. هر عامل یک سیستم خودکار است که دارای انتخابی از ورودیها و تفسیر کامل است و تشخیص، از طریق فرایند تعامل با عوامل دیگر صورت می گیرد.

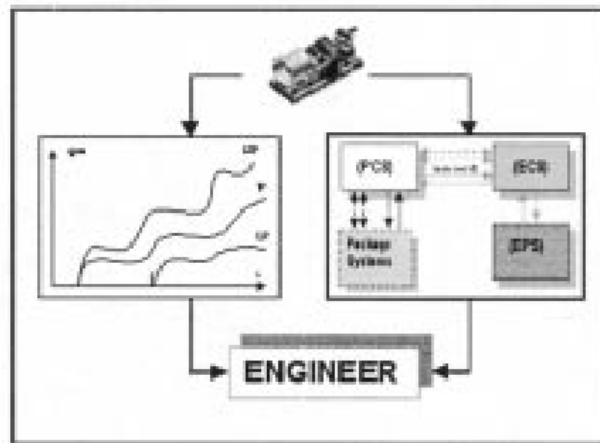
این مقاله به معرفی سیستم چندعاملی کنترل شرایط فوری (**COMMAS**) توسعه یافته جهت تسهیل وظیفه کنترل عملکرد توربین گاز می پردازد. عوامل، ارزیابیها را از سنسورهای توربین و سیستم مدیریت مهندسی

موجود EMS جمع آوری کرده و به تحلیل ارزیابی های غیرعادی پرداخته و خطاها را شناسایی می کنند. هدف این اثر، بهبود دقت سیستم های کنترل موجود از طریق ایجاد مولفه های نرم افزاری کوچکتر است که حاوی اطلاعات جزئی مربوط به وضعیت کارخانه کنترل شده می باشد. آنها به روش دینامیک و برای حمایت از ترکیب داده ها، اثبات بین سنسوری و توابع پشتیبان تصمیم تعامل می کنند. این گفته، در تضاد با اپلیکیشن های مرکزی است که کل کارخانه را در یک سیستم نرم افزاری مدلسازی می کنند.

در این مقاله، بخش II به بحث درباره دامنه مسئله اپلیکیشن کنترل شرایط توربین گازی می پردازد. بخش III به ارائه چارچوب مفهومی سیستم چندعاملی پیشنهادی پرداخته و بخش IV به توصیف تحلیل و طراحی، از جمله جریان اطلاعات بین عوامل، جهت تکرار وظایف تشخیصی اجرایی توسط مهندس می پردازد. بخش V نتایج و برخی جات اثرات آینده را مطرح می کند.

II. دامنه مسئله

کنترل شرایط توربین های گاز شامل ارزیابی دائمی و ارزیابی وضعیت توربین ها در طول مراحل مختلف آن است (راه اندازی، عملیات، نگهداری و تعمیر). در معماری پیشنهادی، نهادهای نرم افزاری مستقل، توانایی شناسایی مرحله عملیاتی توربین گاز و استدلال مربوطه، بروش مناسب و مبتنی بر پایگاه دانش مخصوص مرحله KB را خواهند داشت. بنابراین، مدل چند عاملی تغییر نخواهد کرد بلکه هر عامل از پایگاه دانش مختلف برای مراحل مختلف عملیاتی استفاده خواهد کرد. در این اثر، سیستم نرم افزاری، ابتدا مورد هدف پیاده سازی در طول مرحله راه اندازی توربین قرار می گیرد که پایگاه دانش در آن، از متخصصین کسب شده است و ارشيو می شود. عوامل نرم افزاری هوشمند بر تمام اقدامات صورت گرفته از استارت اولیه موتور سرد تا نقطه ای تمرکز خواهد داشت که سرعت سنکرون در آن بدست می آید. سیستم مدیریت موتور EMS، اطلاعات متنی و گرافیکی را برای مهندسین، همانند تصویر 1 ارائه می دهد. تقسیم مسئولیت بین EMS، سیستم کنترل بسته (PCS) و سیستم حفاظت از موتور (EPS) صورت می گیرد.

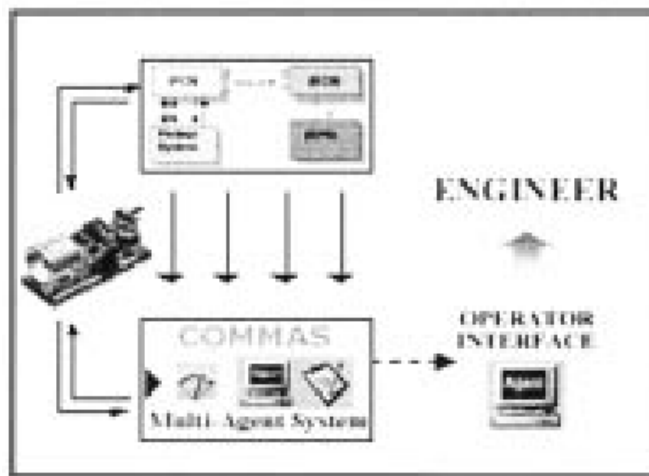


تصویر 1. وضعیت فعلی دامنه مسئله

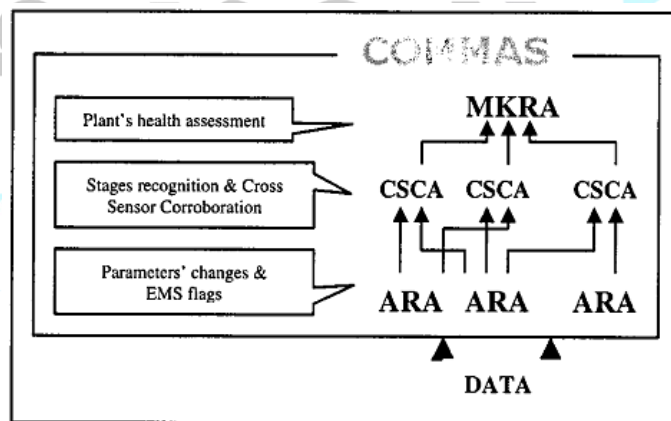
از EMS چندین پرچم تولید می شوند که آلامر خطا یا نشانگر وضعیت هستند و از EMS, PCS and EPS عبور می کنند. اینها نشانگر مراحل مختلف راه اندازی توربین گاز می باشند. در حال حاضر، به خاطر فقدان ابزارهای تشخیص، متخصصین سیستم کنترل، به اجرای تشخیص آنلاین از طریق پرچم های نمایشی مرتبط با نمودارهای چاپ شده مختلف آنلاین می پردازند که سطوح پارامترهای مختلف توربین گاز در هر نقطه زمانی را نشان می دهند (برای مثال، HP, IP and LP spool speeds که سرعت فشار بالا، متوسط و پایین را در توربین نشان می دهند). پیچیدگی وظایف مهندسی، نیاز به سیستم هوش غیرمتمرکز را ارتقاء می بخشد که از منابع مختلف داده و از طریق بکارگیری تکنیک های مختلف استدلال استفاده خواهد کرد. رویکرد هوشمند متمرکز، فاقد قابلیت انعطاف و توسعه پذیری است. اهداف و فعالیت های استدلال، را باید تثبیت کرد. حرکت به معماری مبتنی بر عامل، به اجرای وظایف همزمان و پیچیده اجازه داده، اداره بهتر داده های نادرست صورت گرفته و هر عامل را می توان بطور مستقل بروزرسانی کرد. بنابراین، این نیز بر نیاز به استفاده از عوامل هوشمند، نهادهای نرم افزاری توزیعی مستقل تاکید دارد که توانایی استفاده از تکنیک های هوش مصنوعی مختلف و ارتباط و تعامل با مسئله مهندسی پیچیده، همچون کنترل شرایط را دارند.

در مقایسه با سناریوی قدیمی تصویر 1، راه حل اتوماتیک و غیرمتمرکز پیشنهادی در تصویر 2 نشان داده شده است. نرم افزار جدید، مبتنی بر عوامل هوشمند است که سیستم های توربین را از جانب متخصصین نظارت کرده و کنترل می کنند. عوامل باید در سیستم فیزیکی بزرگ، دینامیک و پیچیده ای عمل کنند که قرار است به راه حل های جهانی دست یافت (برای مثال، تشخیص خطا)، بنابراین، تکنیک های مختلف حل مسئله را باید برای

تحلیل و پردازش منابع مختلف داده بکار برد. مولفه های نرم افزار، منابع داده خود را بطور محلی پردازش کرده و برای ارتباط نتایج خود جهت تولید راه حل منسجم، تعامل می کنند.



تصویر 2. راه حل پیشنهادی دامنه مشکل



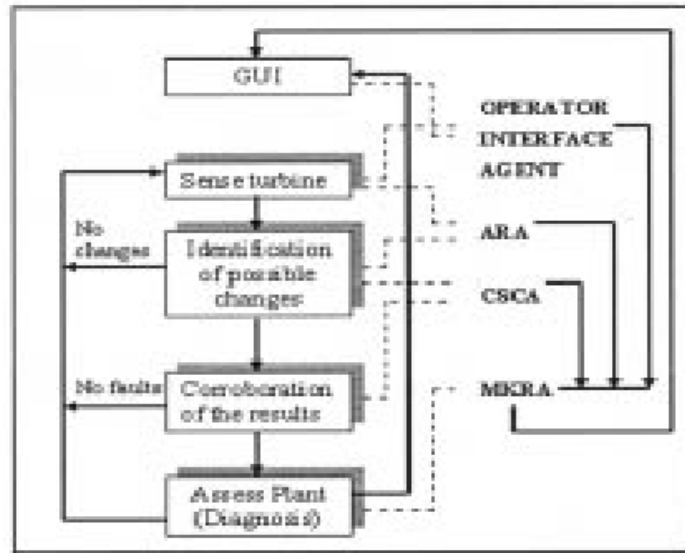
تصویر 3. سیستم چندعاملی کنترل شرایط

مکانیسم رابط تعبیه شده در هر عامل، از جلسات استخراج دانش ساختاری با متخصصین سیستم کنترل توربین گاز، مشتق شده است. اطلاعات متمرکز و دقیق درباره کارخانه و فرایندهای آن، به شکل مدل‌های دانش و نوشتاری ارائه شد که برای مشخص کردن سیستم هوشمند استفاده شده اند. ارتباط بین سیستم چندعاملی و متخصصین به وسیله عامل Operator Interface شکل می گیرد که نتایج فرایند نرم افزاری و متخصص را بررسی کرده و می تواند نگرش جهانی به وضعیت کارخانه داشته و به جریان اطلاعات در نرم افزار، در طول

اجرای آن دسترسی داشته باشد. معماری مبتنی بر عامل، به هر نمونه معقول، اجازه استفاده شدن توسط هر عامل فردی را می دهد. مدل‌های توربین برتی گنجایش در عامل بررسی شده اند که از استدلال مبتنی بر مدل استفاده می کند. بعلاوه، استدلال موردی و شبکه های عصب مصنوعی نیز بررسی شده اند.

III. چارچوب مفهومی

گسترده‌گی دانش، انواع وظایف کنترل کننده شرایط و تعاملات پیوسته بین ماژول های نرم افزاری، استفاده از سیستم چندعاملی را نشان می دهند که می تواند به انواع مختلف اطلاعات از منابع مختلف دسترسی داشته، فیلتر کرده و ترکیب کنند. ساختار سلسله مراتبی سیستم چندعاملی، نتیجه عملکرد فرایند کنترل شرایط است. در مرحله اول، تحلیل پرچم های ارائه شده از EMS و داده های خام مربوط به HP, IP and LP spool speeds را باید اجرا کرد. نتایج این تحلیل تایید شده اند، همانگونه که پارامترها باید روابط معینی در مراحل معین راه اندازی توربین داشته باشند. تشخیص نهایی، نتیجه کل روندی است که در آن، انواع مختلف مسائل را می توان شناسایی کرد. رویکرد سلسله مراتبی COMMAS که در تصویر 3 آمده است، شامل سه لایه مختلف می باشد: عوامل استدلال ویژگی (ARA)، عوامل تایید سنسور متقابل (CSCA) و عوامل استدلال فرادانش (MKRA). ARA، با استفاده از تکنیک های حل مسئله محلی، تفسیر داده های سنسور کارخانه را برای پارامترهای اصلی توربین (برای مثال)، براساس دانش تعبیه شده آنها (حاصل فرایند استخراج با متخصصین) درباره محدودیت و آستانه های آنها در طول مراحل معین راه اندازی توربین گاز اجرا خواهد کرد. لایه بعدی عوامل CSCA، نتایج تفاسیر قبلی را از طریق درخواست شواهد تاییدی مربوط به مسائل و به واسطه ارتباط تایید خواهد کرد و آنها نیز نتایج خود را به MKRA ارسال خواهند کرد. نتیجه نهایی مربوط به وضعیت عمومی کارخانه، جزئیات مرحله راه اندازی توربین، خطاهای موجود در طول فرایند کنترل شرایط و همچنین تبیین علت وقوع آنها، به وسیله عامل Operator Interface به کاربران داده می شوند. مهندسیین مجبور به استخراج این، بدون کمک گرفتن از هر نوع نرم افزاری هستند. تحلیل و طراحی سیستم را می توان فرایند اول مشخص کردن مدل‌های سیستمی دانست که قرار است ساخته شود و در بخش بعد ارائه شده اند.



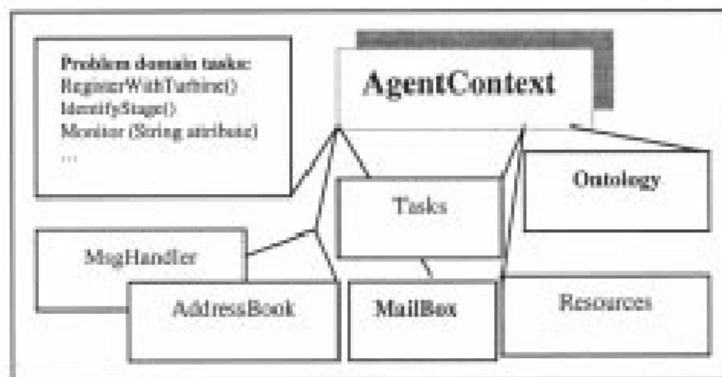
تصویر 4. نمودار فعالیت COMMAS عمومی

IV. تحلیل و طراحی سیستم

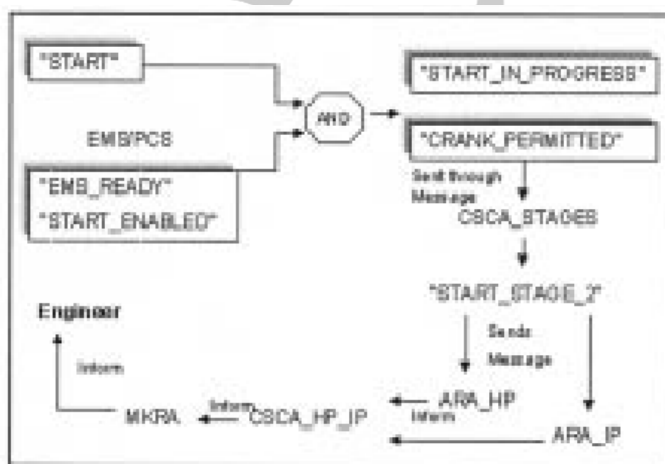
هدف این مرحله تحلیل، توسعه و درک سیستم نرم افزاری و ساختار آن است. این درک، در دامنه مسئله در سازمان سیستم حاصل می شود. سیستم چند عاملی را همچون مجموعه نقش هایی می بینیم که روابط معین با یکدیگر را نشان داده و در تعاملات سیستماتیک روی میدهند. بنابراین، برای تعریف سازمان، باید نقشها و نحوه ارتباط این نقشها به یکدیگر و تعامل آنها را تعریف کنیم. عوامل دارای باورها (حقایقی که وضعیت کارخانه را نشان داده و بعد از هر سنجش، ارتقاء می یابند)، تمایلات (اهدافی که باید به آنها دست یابند) و نیت (وظایفی که باید اجرا کنند تا به اهداف خود برسند) خود هستند. این مدل باور، تمایل و نیت DBI عوامل، رویکرد نظری رسمی است که از منطق مدل برای حل رویدادهای وقوعی در طول عملیات کارخانه استفاده می کند و برای تعدادی از اپلیکیشن های دیگر اعمال شده اند. مسئولیت هر کدام از عوامل، عملکردی بودن و نقش مربوط به آنها را تعیین می کند. برای مثال، یک نقش می تواند، توانایی برای خواندن مورد خاصی از اطلاعات (، ارائه نگرش گرافیکی از وضعیت کارخانه به کاربر باشد. دو نوع کلی عوامل توسعه یافته اند:

- عوامل رابط (عامل Operator Interface جهت کنترل تعامل بین کاربر و سیستم)
- عوامل اپلیکیشن (یعنی ARA, CSCA, and MKRA) برای کنترل اجرای وظایف مختلف اپلیکیشن

بین عوامل مختلف، وابستگی و رابطه وجود دارد؛ بنابراین، باید تعاملات را طوری اجرا کرد که از استدلال متخصص تقلید کنند. این ارتباطات بین نقش ها در تصویر 4 ارائه شده اند که نشان می دهد، عملکرد مدل تعاملی بین عوامل را تعیین می کند. این الگوی تعامل، بطور رسمی تعریف شده و از مراحل اجرای متخصص استخراج گردید.



تصویر 5. مدل عامل

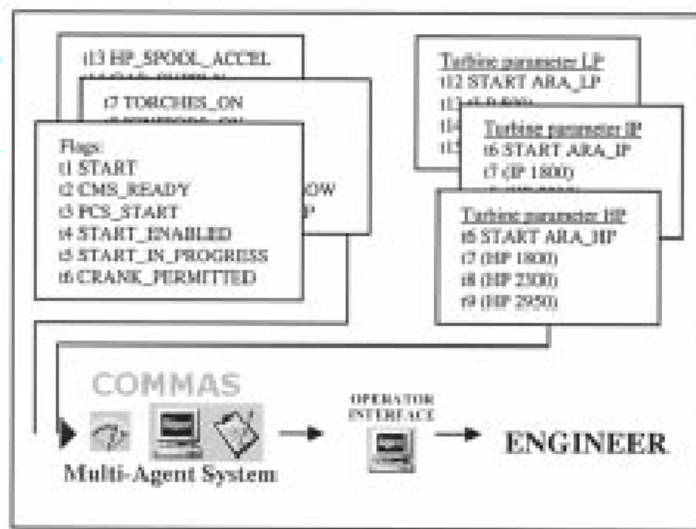


تصویر 6. تعامل عوامل مبتنی بر قانون

مسئولیت و تعاملات عامل پیکربندی شده بین آنها، اساس طراحی اپلیکیشن را تشکیل می دهد که شامل چیکده سطح پایین کد برنامه نویسی برای اجرا است. نمونه عاملی، بسط برنامه نویسی شیء گرا است و در نتیجه، عوامل باید تکنیک های لازم جهت تحقق اهداف اپلیکیشن را احاطه کرده و نتایج خود را با عوامل دیگر مرتبط سازند. فرایند طراحی شامل ایجاد عوامل فردی بوده و مدل کلی عوامل در تصویر 5 ارائه شده است. هر

عامل تواناییها، منابع (حقایق)، وظایف (جداول دامنه مسئله و وظایف کلی عامل)، ایمیل (برای ارسال و دریافت اطلاعات از عوامل دیگر) و هستی شناسی (تمام عوامل موجود در جامعه باید به معنی مشابه مفاهیمی اشاره کنند که نشان می دهند) خود را دارد.

مثالی از دانش متخصص، که در عوامل تعبیه شده است، و براساس آن، سیستم نرم افزار چند عاملی، توربین را حس می کند و سعی در شناسایی هر تغییری می کند که در تصویر 6 ارائه شده است. مهندس COMMAS (پرچم START ایجاد شده است) را شروع کرده است و EMS پرچم های "EMS_READY" and "START_ENABLED." را تولید می کند. اگر بعد از دوره زمانی معین، پرچم های "START_IN_PROGRESS" and "CRANK_PERMITTED" از EMS داده شدند، در اینصورت از CSCA_STAGES مشخص می شود که (که مسئول شناسای مرحله توربین است) توربین مرحله 2 را شروع کرده است. سپس پیامی به ARA_IP and ARA_HP ارسال می شود چون آنها مجبور هستند به جامعه وارد شوند تا سرعت های ARA_IP and ARA_HP را که در این مرحله دیده می شوند، ظاهر شوند.



تصویر 7. اطلاعات ارائه شده به COMMAS

آنها درباره هر تغییری، CSCA_HP_IP را مطلع می سازند و سپس نتایج را به MKRA ارسال می کنند. دومی، مهندس را از نتایج آینده نرم افزار مطلع خواهد ساخت.

روش کلیدی برای تحلیل و طراحی سیستم، حرکت از مجموعه شرایط به طراحی دقیق سیستم چند عاملی برای اجرای مستقیم است که در بخش بعد توصیف خواهد شد.

V. پیاده سازی

این بخش نشان می دهد که چگونه چارچوب مفهومی، براساس تحلیل و طراحی چندعاملی برای کنترل شرایط دنباله راه اندازی توربین گاز اعمال شده است. دنبال شروع را می توان به شش بخش تقسیم کرد که نرم افزار باید شناسایی کند. در حالت موازی، باید خصوصیات سنسور را کنترل کند. عوامل نرم افزار هوشمند، براساس عملکرد خاص خود در نرم افزار توسعه یافته اند. برای مثال، سه ویژگی اصلی HP, IP and LP spool speeds از عوامل ARA_HP, ARA_IP and ARA_LP کنترل شده اند در حالیکه شناسایی مرحله از CSCA_STAGES خاص بدست می آید که پرچم را از EMS توربین دریافت می کند.

جدول 1. نمایش دانش BDI داخل عوامل

CSCA_HP_IP	MKRA
BELIEFS	BELIEFS
(HP_spool_speed (t7 ok))	(stage1 (started true) (finished true))
(IP_spool_speed (t7 ok)) ...	((HP_IP(status ok)), t7)
	(stage2 (started true) (finished ?)) ...
DESIRES	DESIRES
(HP_IP (status ?), t7)	(Turbine_state (t ?))
(HP_IP (status ?), t8) ...	(Fault_occurred (t ?)) ...
INTENTIONS	INTENTIONS
L(HP_spool_speed ok), t7) ^	L(stage1(started true) (finished true) ^ L((HP_IP (status ok)), t-1)
L(IP_spool_speed ok), t7) ^	^ ~M((faultStage1), t-1) →
L((HP_IP ok), t7) →	□ ((stage1 finished), t) ^
□ ((HP_IP (status ok)), t7) →	□ ((proceed next_stage), t)

از آنجاییکه دامنه مسئله، بصورت مجموعه تعاملات در نظر گرفته می شود، توجه با هدف تعاملات و تبادل پیام بین عوامل به آن داده می شود. عوامل داخل COMMAS، با استفاده از ساختار Knowledge Query and Manipulation Language (KQML) [10] ارتباط برقرار می کنند که از نوع Send_message ((name ARA_HP) (context ((name ARA_HP) (sender MKRA)(receiver ARA_HP) (type inform) (register))) است و MKRA پیام (از نوع اطلاعات) را به ARA-HP ارسال می کند تا دستور دهد که دومی در جامعه ثبت شده است و شروع به کنترل HP spool speed کند. اطلاعات COMMAS، پرچم ها

را از EMS و داده های مربوط به HP, IP, and LP spool speeds را در نقاط زمانی خاص (بصورت t_1, t_2, \dots, t_n نشان داده می شود) دریافت می کند که در تصویر 7 نشان داده شده است.

براساس این حقایق، استدلال داخلی عوامل، شامل تغییرات پردازشی، حوادث و روابط آنها در فاصله زمانی است. فرمول منطقی برای نمایش این روش در چارچوب bdi استفاده شده است. اوپراتورهای منطق موجهاست که استفاده شده اند، بصورت زیر برگردان می شوند:

$L(\varphi, t) =$ باور بر این است که ϕ در زمان t روی داده است.

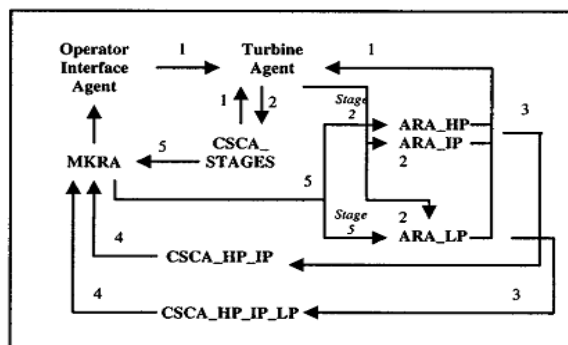
$M(\varphi, t) =$ این امکان وجود دارد که ϕ در زمان t روی دهد.

$\square(\varphi) =$ لازم است که ϕ روی دهد.

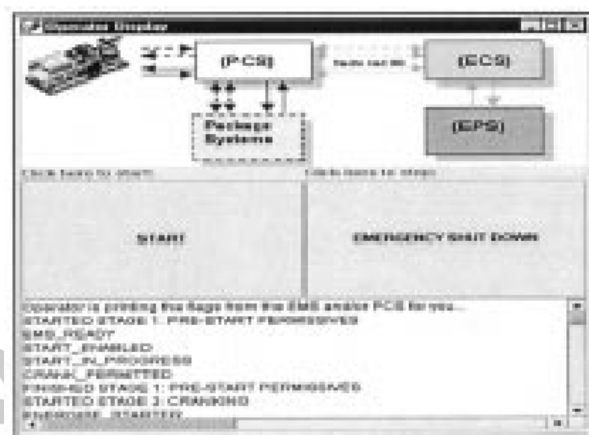
که ϕ هر نوع حقیق ارائه شده توسط عوامل می باشد. همانطور که جدول I نشان می دهد، CSCA and MKRA براساس داده های دریافتی، باورهای معینی دارند. ARA_HP and ARA_IP پیام های مناسب را به CSCA_HP_IP ارسال می کنند که باورهای آنها را نشان می دهد. دومی باید مشخص کند که آیا رابطه بین HP و IP spool speeds مناسب است تا اینکه استدلال خود را اجرا کرده و نتیجه را به MKRA ارسال کند. MKRA همراه با نتایج دریافتی از دیگر عوامل، برای ارزیابی وضعیت توربین و تشخیص اینکه آیا خطا روی داده است یا نه، بطور منطقی استدلال می کند.

برای مثال، در جدول I، باور بر این است که مرحله 1 شروع شده و پایان یافته است و رابطه بین HP و IP spool speeds نرمال است و هیچ امکانی برای وجود خطا نیست. بنابراین MKRA به مهندس اطلاع می دهد که مرحله اول با موفقیت پایان یافته است و به مرحله بعد می رود. باورها، تمایلات و نیت اهدیت خواهند شد. مثالی از تعامل عوامل در COMMAS در تصویر 8 ارائه شده است. که در آن

1. عوامل نیازمند ثبت با عامل توربین می باشند که پایگاه داده توربین را تصدیق می کند
2. بین عامل توربین و CSCA_STAGES و بین عامل توربین و عوامل ARA تعامل وجود دارد.
3. بین ARA_HP, ARA_IP and ARA_LP with the CSCA_HP_IP and CSCA_HP_IP_LP بعد از پردازش اطلاعات در پاسخ به درخواست عوامل CSCA، تعاملاتی وجود دارد.



تصویر 8. مثالی از نمودار تعاملات

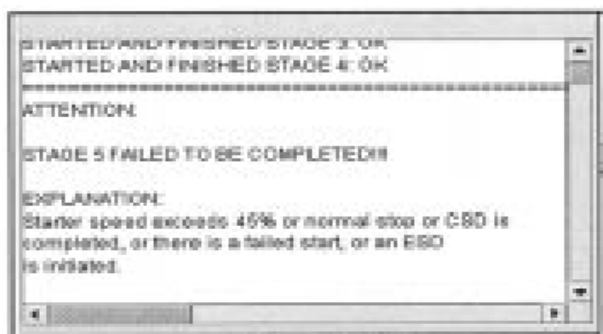


تصویر 9. رابط گرافیکی کاربر عامل راه انداز

4 عوامل CSCA به درخواست MKRA پاسخ می دهند تا به کاربر اجازه بررسی تغییر خصوصیات در نمودار را در طول زمان بدهند

MKRA نه تنها در باره وضعیت توربین به کاربر اطلاع می دهد، بلکه سیستم نرم افزاری را کنترل می کند. در یک محیط چند عاملی، تناقض ها به چند دلیل مختلف شکل می گیرند. عوامل می توانند باورها یا اهداف متناقضی داشته یا منابع محدودی را به اشتراک بگذارند. عوامل نرم افزاری این اثر، مجموعه های متناقض را گروه بندی کرده و با عوامل دیگر جامعه جهت یافت تناقض حقیقی مذاکره می کنند و بسته به منابع قابل دسترس آنها، از کاربر توصیه می خواهند. در میان وظایف مربوط به توسعه سیستم نرم افزاری آینده، مکانیسمی جهت رسیدگی به حل تناقض و دانش ناقص از طریق بکارگیری تکنیک های یادگیری هوشمند مختلف وجود دارد.

COMMAS با استفاده از ZEUS Agent Building Toolkit اجرا می شود که محیطی برای توسعه سریع اپلیکیشن های مشارکتی عاملی است و در تمام سکوهای سخت افزاری اصلی اجرا می شوند. تصویر 9، رابط کاربر گرافیکی توسعه یافته برای عامل Operator Interface را نشان می دهد که در آن، کاربر می تواند پرچم های تولیدی EMS را تصور کند، یعنی زمانیکه آنها برای تحلیل آینده به MKRA ارسال می شوند. در مورد خاموش کرد فوری، کاربر می تواند به سیستم پیشنهاد خاموش شدن را دهد.



تصویر 10. تبیین MKRA

نتایج نهایی، از Operator Interface ارائه خواهند شد که مثالی از آن در تصویر 10 ارائه شده است. راه اندازی این مورد، به خاطر مشکل مرحله 5 کامل نشد.

با افزایش تعداد تحقیقات و افزایش هوش بیشتر، سیستم قادر به تمایز بین خطاهای مرتبط خواهد بود. خلاصه، سیستم وظیفه زیر را اجرا می کند:

- پردازش پرچم های تولیدی سیستم نرم افزاری موجود جهت تعیین مرحله راه اندازی توربین
- کنترل HP, IP, LP spool speeds و همچنین دیگر پارامترهای مهم (یعنی، دما، فشار) توربین و در نهایت
- ترکیب نتایج این فرایندهای توزیعی برای کنترل اتوماتیک شرایط آنلاین

VI. بحث

در این اثر، طراحی و پیاده سازی سیستم جدیدی توصیف شده است که معرفی فناوری عامل، راه اندازی سیستم های قدیمی کنترل شرایط را به چالش خواهد کشید. طراحی مدولار جدید، معماری مرکزی را ارتقاء داده و به ارزیابی اتوماتیک خطی سلامت کارخانه، از طریق پردازش اطلاعات غیرمرکزی پرداخته و از تصمیمات مهندسی حمایت خواهد کرد که تشخیص خطا را بطور دستی اجرا می کنند.

شرکت Rolls-Royce اخیراً به دنبال فناوری عاملی و با هدف راه اندازی اپلیکیشن های بستر آزمون جهت تایید عملکرد سیستم های توربین واقعی است.

اگرچه Zeus ابزار با کیفیت تجاری برای توسعه اپلیکیشن های مقاوم صنعتی نیست، اما نتایج نویدبخشی برای اپلیکیشن ارائه کرده است. این سیستم با وارد کردن بیش از یک تکنیک هوشمند محاسباتی ارتقاء یافته و بعداً به ارزیابی نتایج روش موثرتر خواهد پرداخت. این را می توان با افزودن عملکرد جدید به سیستم چند عاملی ناشی از مدولاریته طراحی آن، به آسانی اجرا کرد.

REFERENCES

- [1] J. H. Williams, A. Davies, and P. R. Drake, *Condition-Based Maintenance and Machine Diagnostics*: Chapman & Hall, 1992.
- [2] J. R. McDonald et al., *Intelligent Knowledge Based Systems in Electrical Power Engineering*. London: Chapman & Hall, 1997.
- [3] D. Gemmell, J. R. McDonald, R. W. Stewart, R. N. T. Brooke, and B. J. Weir, "A consultative expert system for fault diagnosis on turbine generator plant," *Institution of Mechanical Engineers' Proceedings Part A, Journal of Power and Energy*, Dec. 1994.
- [4] S. D. J. McArthur, S. C. Bell, J. R. McDonald, R. Mather, and S. M. Burt, "Knowledge and model based decision support for power system protection engineers," in *Proceedings of the International Conference on Intelligent Systems Applications to Power Systems*, 1996, pp. 215-219.
- [5] C. Booth, J. R. McDonald, and W. Hagman, "The use of artificial neural networks for the prediction and classification of vibration behavior in plant transformers," in *American Power Conference*, Chicago, 1995.
- [6] C. P. Azevedo, B. Feiju, and M. Costa, "Control centres evolve with agent technology," *IEEE Computer Applications in Power*, vol. 13, no. 3, pp. 48-53, 2000.
- [7] M. J. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practise," *Knowledge Engineering Review*, vol. 10, no. 2, pp. 115-152, 1995.
- [8] A. S. Rao and Georgeff, "BDI agents: From theory to practice," Australian Artificial Intelligence Institute, Melbourne, Australia, 56, 1995.
- [9] E. Walther, H. Eriksson, and M. A. Musen, "Plug and play: Construction of task specific expert system shells, using sharable context ontologies," Knowledge based systems laboratory, Stanford University, KSI-92-40, 1992.
- [10] T. Finin, Y. Labrou, and J. Mayfield, "KQML as an agent communication language," Bradshaw Jeffrey, Software Agents, 1997.
- [11] ZEUS Agent Building Toolkit (1999). [Online]. Available: <http://www.labs.bt.com/projects/agents/>
- [12] The source for Java technology [Online]. Available: <http://java.sun.com/>



برای خرید فرمت ورد این ترجمه، بدون واتر مارک، اینجا کلیک نمایید.



این مقاله، از سری مقالات ترجمه شده رایگان سایت ترجمه فا میباشد که با فرمت PDF در اختیار شما عزیزان قرار گرفته است. در صورت تمایل میتوانید با کلیک بر روی دکمه های زیر از سایر مقالات نیز استفاده نمایید:

لیست مقالات ترجمه شده ✓

لیست مقالات ترجمه شده رایگان ✓

لیست جدیدترین مقالات انگلیسی ISI ✓

سایت ترجمه فا ؛ مرجع جدیدترین مقالات ترجمه شده از نشریات معتبر خارجی