ارائه چهار مدل کاربردی جدید در قالب یک متدولوژی ترکیبی به منظور ارتقای کاربرد دامنه

عرفان قندهاری[۱] و فاطمه سعادت‌جو[۲،*]

اطلاعات مقاله

دریافت مقاله: ۱۳۹۵/۰۱/۱۱

پذیرش مقاله: ۱۳۹۶/۰۷/۰۵

واژگان کلیدی:

مهندسی نرم‌افزار عامل‌گرا،
سیستم مبتنی بر عامل،
متدولوژی،
ADELFE،
PASSI،
AP.

چکیده

در این مقاله با توجه به نقاط ضعف متدولوژی‌های عامل‌گرا در فازهای مختلف و تأثیر به‌سزایی که این کاستی‌ها، می‌تواند در کیفیت و کارایی پروژه‌های نرم‌افزاری داشته باشد به ارائه راهکاری ترکیبی برای معرفی متدولوژی (AP) از متدولوژی‌هایADELFE و PASSI پرداخته‌ایم. بدین منظور دو متدولوژی مورد بررسی در قالب یک چارچوب ارزیابی جامع بر اساس معیارهای مفاهیم و ادراک‌ها، زبان مدل‌سازی، فرآیند و عمل‌گرایی مورد ارزیابی قرار گرفته‌اند. با توجه به محدودیت‌های منابع و زمان، تصمیم به انتخاب متدولوژی‌های نمونه برای ارزیابی و یکسان‌سازی اتخاذ گردید. این انتخاب بر اساس میزان کاربرد متدولوژی‌های مورد بررسی و همچنین امکان یکسان‌سازی آن‌ها از دید افراد متخصص و خبره صورت گرفته است. بر اساس نتایج ارزیابی، متدولوژی ADELFE مراحل تحلیل سیستم‌های عامل‌گرا را به‌خوبی پشتیبانی می‌کند و مرحله طراحی در آن به دلیل عدم مدل کردن قابلیت عامل‌ها، کامل نمی‌باشد، از طرف دیگر با توجه به این که، متدولوژی PASSI مراحل طراحی و تعامل-های بین عامل‌ها را به‌خوبی پشتیبانی می‌کند، با استفاده از مزایای دو متدولوژی، یک متدولوژی ترکیبی ارائه می‌گردد که در آن از مرحله تحلیل متدولوژی ADELFE و مرحله طراحی متدولوژی PASSI استفاده می‌شود. پس از ایجاد متدولوژی ترکیبی چهار مدل کاربردی جدید به منظور ارتقای کاربرد دامنه پیشنهاد گردیده است. این مدل‌ها شامل کنشگرگسترش‌یافته، هدف مجزا، سرویس و مدل سیستم گسترش یافته می‌باشند. به منظور تشریح فازهای مختلف این متدولوژی، از آن در یک مطالعه‌ی موردی استفاده شده است. نتایج این پژوهش می‌تواند زمینه را برای معرفی نسل بعدی متدولوژی‌های عامل‌گرا فراهم آورد.

۱- مقدمه

امروزه دامنه وسیعی از مدل‌های مهندسی نرم‌افزار (مانند برنامه‌نویسی ساختاری، مدل‌های شی‌گرا، روش‌های مبتنی بر مؤلفه و موارد مشابه) با هدف ایجاد تسهیلاتی در پروسه مهندسی برای تولید نرم‌افزار یا کاهش پیچیدگی کاربردهایی که می‌توانند ساخته شوند، مطرح می‌باشد [۱]. در حالی که هیچ‌کدام از مدل‌های جاری مهندسی نرم‌افزار نمی‌توانند به‌طور کامل تکنولوژی نرم‌افزار را با نیازهای تجاری هماهنگ کنند، مهندسی نرم‌افزار عامل‌گرا به‌عنوان مدل قدرتمند جدید در سیستم‌ها و محاسبات پدیدار گشته است [۲]. هدف اصلی آن ایجاد متدولوژی، ابزار و امکاناتی برای تهیه‌ی ساده و نگهداری از نرم‌افزارهای مبتنی بر عامل است [۳]. مفاهیم متدولوژی‌ها و ابزارها، از مدل‌های عامل، یکی از مناسب‌ترین کاندیدها برای پایه‌گذاری نسل بعدی

* پست الکترونیک نویسنده مسئول: saadatjou@sau.ac.ir

۱. کارشناسی ارشد، دانشکده مهندسی گروه کامپیوتر، دانشگاه علم و هنر یزد

۲. استادیار، دانشکده مهندسی گروه کامپیوتر، دانشگاه علم و هنر یزد

سیستم‌های نرم‌افزاری می‌باشد [۴]. در حقیقت مهندسی نرم‌افزار عامل‌گرا مرحله بعدی در سیر تکاملی مهندسی نرم‌افزار است، که قدرت و مزایای تکنیک‌های دیگر مهندسی نرم‌افزار مانند شی‌گرایی را افزایش می‌دهد. مهندسی نرم‌افزار شی‌گرا نمی‌تواند خاصیت‌های مشابه عامل‌ها از قبیل هوشمندی و انعطاف‌پذیری، ارتباطات و قدرتمند بودن را در محیط‌های پیچیده، پویا و محیط‌های حساس و توزیع‌شده و اینترنت داشته باشد [۵]. در حقیقت توسعه کاربردهای قدرتمند صنعتی، قابلیت دسترسی به متدولوژی‌های نرم‌افزاری را ایجاب می‌کند. این متدولوژی‌ها باعث توسعه نرم‌افزار به‌طور سیستماتیک شده و منجر به افزایش کیفیت محصول نرم‌افزاری می‌شوند. توسعه سیستم‌های عامل‌گرا به‌وسیله روش‌های غیررسمی و ابتکاری و نه اصول رسمی و تکنیک‌های مهندسی تعریف می‌شود [۶].

با ظهور دیدگاه عامل‌گرا در تولید سیستم‌های نرم‌افزاری، زبان‌های برنامه‌نویسی، ابزارهای تولید نرم‌افزار و متدولوژی-های مهندسی نرم‌افزار مناسب با آن نیز مطرح گردید [۷]. عامل‌ها دارای شباهت بسیاری با اشیا می‌باشند، اما دیدگاه و ویژگی‌هایی که یک عامل را از یک شی متمایز می‌نماید، موجب می‌شود متدولوژی‌های شی‌گرا برای سیستم‌های مبتنی بر عامل مناسب نباشند. به همین علت این سیستم-ها، نیاز به متدولوژی‌های مهندسی نرم‌افزار بر اساس ویژگی‌های منحصر به فرد عامل‌ها دارند [۸]. گسترش متدولوژی‌های موجود این مزیت را دارد که مفاهیم، نمادها و روش‌های به‌کار رفته در متدولوژی‌های موجود، به‌خصوص روش‌های شی‌گرا، برای مهندسین نرم‌افزار آشنا می‌باشند. اما، باید توجه داشت که اساس تجزیه سیستم‌ها در روش-های شی‌گرا با تجزیه در روش‌های مبتنی بر عامل متفاوت است و روش‌های شی‌گرا امکانات لازم برای مدل‌سازی بسیاری از جنبه‌های سیستم‌های مبتنی بر عامل مانند وضعیت‌های ذهنی، رفتار اجتماعی، سازمان عامل‌ها، همکاری، هماهنگی و بعضی از خصوصیات عامل‌ها مانند واکنشی و پیش فعال بودن را فراهم نمی‌کنند. به همین منظور نیاز به روش‌ها و تکنیک‌های جدید مخصوص تجزیه و تحلیل سیستم‌های مبتنی بر عامل مطرح می‌گردد، با توجه به موارد بیان شده در این مقاله قصد داریم، با ترکیب نقاط قوت دو متدولوژی پرکاربرد ADELFE [۹] و PASSI [۱۰] و همچنین ارائه چهار مدل کاربردی شامل مدل‌های کنشگرگسترش‌یافته، هدف مجزا، سرویس و مدل سیستم گسترش یافته به یک متدولوژی ترکیبی توسعه یافته دست یابیم. بدین منظور ابتدا بر اساس یک چارچوب ارزیابی جامع در حیطه عامل‌گرایی متدولوژی‌های ADELFE و PASSI را با روش دلفی‌فازی [۱۱] مورد ارزیابی قرار می‌دهیم، سپس با ترکیب نقاط قوت دو متدولوژی مورد بررسی، یک متدولوژی یکسان‌سازی شده را ایجاد و پس از آن راهکارهایی را برای توسعه آن مطرح می‌کنیم.

در ادامه به بررسی کارهای انجام شده خواهیم پرداخت. سپس در بخش سوم به منظور تفکیک رویکردهای عامل‌گرا و شی‌گرا به بررسی تفاوت‌های عامل‌ها و اشیا می‌پردازیم. در بخش چهارم به معرفی چارچوب انتخاب شده و نتایج ارزیابی دو متدولوژی اشاره خواهیم نمود. در بخش پنجم به معرفی سیستم انتخابی در مقاله می‌پردازیم. در بخش ششم به تشریح فازهای تحلیل و طراحی در متدولوژی AP خواهیم پرداخت و در نهایت در بخش هفتم به نتیجه‌گیری و راهکارهای آینده اشاره خواهیم کرد.

۲- مروری بر کارهای انجام شده

در این بخش تحقیقات انجام شده در زمینه توسعه متدولوژی‌های پرکاربرد GAIA، MASE، TROPOS و همچنین متدولوژی‌های مورد بررسی در مقاله و سایر متدولوژی‌های عامل‌گرا را، مورد تحلیل قرار می‌دهیم.

۲-۱- متدولوژی GAIA و توسعه‌های آن

یکی از نخستین متدولوژی‌های ارائه شده در بحث مهندسی نرم‌افزار عامل‌گرا، متدولوژی GAIA نام دارد که در آن مراحل تحلیل و طراحی پشتیبانی شده است. تحلیل در متدولوژی GAIA شامل مراحل مدل‌سازی نقش‌ها و مدل-سازی تعامل است. در مرحله طراحی مدل عامل‌ها، مدل سرویس و مدل شناخت سیستم تولید می‌گردد [۱۲]. در این متدولوژی عامل‌ها در دو سطح مختلف، سطح اجتماع عامل‌ها و ساختار سازمانی سیستم و سطح ساختار عامل مورد بررسی قرار می‌گیرند. با توجه به این‌که متدولوژی GAIA یکی از نخستین متدولوژی‌های ارائه شده برای سیستم‌های مبتنی بر عامل بوده است، در نسخه اصلی این متدولوژی کمبودهایی وجود داشته و تحقیقات مختلفی برای بهبود آن، صورت گرفته است. یکی از بهبودهایی که برای متدولوژی GAIA ارائه شده است، توانایی مدل کردن

سیستم‌های قابل پیاده‌سازی در اینترنت را به این متدولوژی اضافه نموده است [۱۳]. در این تحقیق به ویژگی‌هایی از قبیل باز بودن و تضاد اهداف عامل‌های شرکت کننده در سیستم توجه شده است.

با استفاده از بهبود دیگری که بر روی متدولوژی GAIA صورت گرفته است، متدولوژی جدیدی با عنوان ROADMAP ارائه شده است. در این متدولوژی جدید توانایی مدل کردن سیستم‌های باز به متدولوژی GAIA اضافه شده است [۱۴].

بهبود دیگری که بر روی متدولوژی GAIA صورت گرفته است [۱۵]، با استفاده از ترکیب آن با روش مدل‌سازی AUML[1] می‌باشد. در این بهبود، مرحله طراحی در متدولوژی GAIA در نظر گرفته شده است و در چهار مرحله پروتکل، تعامل، عامل و سازمان مدل‌های توسعه‌یافته UML برای عامل‌ها بهبود یافته است.

در [۱۶] دو بهبود دیگر برای متدولوژی پیشنهاد شده است. نخست آن که مرحله طراحی عامل به متدولوژی اضافه شده است. بر اساس این بهبود، مرحله طراحی عامل که در متدولوژی اولیه به‌صورت یک جعبه سیاه در نظر گرفته شده است، بر اساس معماری‌های شناخته شده برای عامل، گسترش یافته است. همچنین برای بهبود فرآیند تولید سیستم مبتنی بر عامل با استفاده از متدولوژی GAIA یک رویکرد مبتنی بر تکرار در متدولوژی در نظر گرفته شده است. بر اساس این پیشنهاد، برای جلوگیری از تولید سیستمی که با نیازهای کاربر منطبق نیست و برای غلبه بر پیچیدگی سیستم، بهتر است سیستم در چندین تکرار تولید شود.

۲-۲-متدولوژی MASE و توسعه‌های آن

متدولوژی MASE یک متدولوژی برای تحلیل و طراحی سیستم‌های مبتنی بر عامل می‌باشد و از نقاط قوت آن این است که در آن ابزاری با نام Agent Tool ارائه شده است که کلیه مراحل تحلیل و طراحی سیستم در متدولوژی MASE را پشتیبانی می‌نماید [۱۷]. مرحله تحلیل در متدولوژی MASE، شامل سه مرحله اصلی تعیین اهداف سیستم، اعمال موارد کاربرد و بازبینی نقش‌ها است. مرحله طراحی در متدولوژی MASE شامل چهار مرحله ایجاد کلاس‌های عامل، ساخت گفتگوها، ترکیب کلاس‌های عامل و طراحی سیستم می‌باشد.

متدولوژی MASE نیز از زمان ارائه نسخه اولیه تاکنون، به‌صورت‌های مختلف بهبود و توسعه یافته است. در یکی از بهبودهای متدولوژی MASE، مرحله مدل‌سازی هستان شناسی به مرحله تحلیل متدولوژی اضافه شده است [۱۸]. در این توسعه برای متدولوژی، ابتدا مشخص می‌گردد که هدف و محدوده هستان‌شناسی مورد نیاز عامل چیست، سپس داده‌های موجود در دامنه سیستم که بدین منظور مورد استفاده قرار می‌گیرند، جمع‌آوری می‌گردد.

همچنین امکان مدل‌سازی روابط سازمانی میان عامل‌ها نیز در توسعه دیگری به متدولوژی MASE افزوده شده است [۱۹]. در این توسعه، پس از مرحله مدل‌سازی هستان-شناسی عامل در مرحله تحلیل، تحلیل و مدل‌سازی ساختار سازمانی انجام می‌شود. در این مرحله، قوانین سازمانی که محدودیت‌های رفتار عامل و ارتباط و تعامل میان عامل‌های یک سازمان است، مشخص می‌شود. بدین ترتیب قوانین سازمانی در قالب تعدادی محدودیت بر روی رفتار نقش (در مرحله تحلیل) ارائه می‌شود. برای این منظور نمادهایی برای توصیف این محدودیت‌ها در نظر گرفته شده است.

افزودن قابلیت حرکت به عامل‌های سیستم از دیگر بهبودهای متدولوژی MASE می‌باشد [۲۰]. بدین منظور، در مرحله تحلیل، در مدل‌سازی فعالیت‌ها که در قالب نمودار حالت انجام می‌شود، دستور حرکت به متدولوژی اضافه شده است. در این مرحله، نتیجه حرکت و علت شکست آن (در صورت شکست) مشخص می‌شود تا امکان بازیابی عامل از این شکست، فراهم شود. همچنین در مرحله طراحی، این امکان به عامل‌ها اضافه شده است که با دریافت دستور حرکت، وضعیت داخلی همه‌ی اجزای خود را ذخیره نمایند تا پس از انتقال به مقصد امکان شروع مجدد فعالیت-ها از وضعیتی که در آن قرار داشته‌اند، فراهم شود.

بهبود دیگری که در متدولوژی MASE صورت گرفته است، استفاده از روش Method Fragment برای ارائه یک فرآیند تولید نرم‌افزار است [۲۱]. در این بهبود، فرآیند قابل سازمان‌دهی تعریف شده است که با استفاده از آن می‌توان متدولوژی را برای کاربردهای مختلف اختصاصی نمود.

[1] Agent Unified Modeling Language

۲-۳-متدولوژی TROPOS و توسعه‌های آن

متدولوژی TROPOS یکی از متدولوژی‌های نسبتاً جدید در مهندسی نرم‌افزار عامل‌گرا است [۲۲]. تفاوت اصلی این متدولوژی، با سایر متدولوژی‌ها، تمرکز آن بر روی تحلیل نیازمندی‌ها است به‌طوری که در مرحله اول ذینفعان سیستم و تمایلات آن‌ها مورد بررسی و تحلیل قرار می‌گیرند. در طراحی متدولوژی TROPOS، از تئوری‌های ساختارهای اجتماعی مانند تئوری سازمانی، طراحی فرآیند و پیوستگی استراتژیک الهام گرفته شده است. با وجود این-که متدولوژی TROPOS به‌عنوان یک متدولوژی نسبتاً جدید در سیستم‌های مبتنی بر عامل ارائه شده است، توسعه‌های متعددی برای این متدولوژی پیشنهاد شده است. در [۲۳]، یک مدل رسمی تحلیل هدف به متدولوژی TROPOS اضافه شده است که باعث تقویت این متدولوژی در مرحله تحلیل نیازمندی‌ها گردد. این تحقیق، با استفاده از ابزارها و مدل‌های استدلال رو به جلو و رو به عقب، و با به‌کارگیری روش تحلیل هدف رسمی، در مرحله تحلیل نیازمندی‌ها، امکان یافتن اهداف متضاد تعریف شده در مرحله تحلیل نیازمندی‌ها و تعیین اهدافی که در صورت برآورده شدن، هدف سیستم به‌صورت کلی برآورده می‌شود را فراهم نموده است. همچنین با استفاده از این روش، این امکان وجود دارد که با استفاده از استدلال رو به جلو مشخص گردد، برای این‌که هدف سیستم برآورده شود، کدام مجموعه اهداف سیستم باید برآورده شوند. با به-کارگیری این روش، امکان وجود گزینه‌های مختلف در مرحله تحلیل سیستم فراهم می‌شود و تحلیل‌گر سیستم امکان انتخاب میان اهداف مختلف را خواهد داشت. بدین منظور گراف هدف و نمادهای ارتباط میان اهداف برای توصیف هدف‌ها و ارتباط میان آن‌ها به‌صورت رسمی، ارائه شده است.

همچنین در [۲۴] تحلیل اهداف سیستم‌ها و گزینه‌های مختلف برای اختصاص هدف به نقش در سیستم مبتنی بر عامل در مرحله تحلیل اهداف سیستم به متدولوژی TROPOS در نظر گرفته شده است. بدین منظور یک روش تخصیص هدف به نقش ارائه شده است که امکان تولید گزینه‌های مختلف برای نقش‌های سیستم و انتخاب میان آن‌ها را فراهم می‌کند.

در تحقیق دیگری که در ارتباط با متدولوژی TROPOS انجام شده است، به نیازهای تعامل میان عامل در سیستم‌های مبتنی بر عامل و متدولوژی TROPOS پرداخته شده است [۲۵]. بنابر نظر نویسندگان مقاله نیازمندی‌های تعامل عامل در این متدولوژی در مراحل اولیه مورد توجه قرار نمی‌گیرد و پروتکل‌های ارتباطی تا مرحله طراحی سیستم مشخص نمی‌شوند. از طرف دیگر، در نظریهCommitment Protocol، پروتکل‌های تعامل و زمینه اجرای آن‌ها در سیستم کاربردی مشخص نمی‌شود. برای حل این مسئله در این مقاله، از ترکیب این دو موضوع استفاده شده است به‌طوری که Commitment Protocol ها به متدولوژی TROPOS اضافه شده است. بدین منظور راهنمایی‌هایی برای تعیین تعامل میان بازیگران مختلف سیستم بر اساس مفاهیم ارائه شده در متدولوژی (مانند وابستگی میان بازیگران) ارائه شده است. بر اساس این راهنمایی‌ها به‌صورت سیستماتیک می‌توان مشخص کرد که چه پروتکل‌هایی میان بازیگران مختلف سیستم باید تعریف گردد.

۲-۴-سایر متدولوژی‌ها

علاوه بر متدولوژی‌های بیان شده، که در مهندسی نرم‌افزار عامل‌گرا بسیار مورد توجه و استفاده هستند، متدولوژی‌های دیگری در این حوزه معرفی شده‌اند، اما تحقیقات زیادی برای توسعه و بهبود آن‌ها گزارش نشده است. در این بخش به معرفی اجمالی این متدولوژی‌ها می‌پردازیم.

- متدولوژی MASSIVE که در آن شش دیدگاه محیط، وظیفه، نقش، تعامل، جامعه و معماری مدل‌سازی می-شود [۲۶].
- متدولوژی MEESAGE که در آن مفاهیم نقش، سازمان و منابع در قالب موجودیت‌های سیستم، وظیفه و پروتکل‌های تعامل در قالب فعالیت‌های موجود در سیستم و هدف به‌عنوان حالت ذهنی عامل تحلیل می-شوند [۲۷]، و در پی آن متدولوژی INGENIAS که در آن پنج مدل عامل، تعامل، وظیفه و هدف، سازمان و محیط تولید می‌شود [۲۸].
- متدولوژی MAS-Common KADS که توسعه متدولوژی‌های مبتنی بر دانش است و در آن فعالیت-های مدل‌سازی عامل، مدل کردن وظایف، مدل کردن هماهنگی، کامل کردن دانش، مدل کردن ساختار سازمانی در مرحله تحلیل صورت می‌گیرد. طراحی در این متدولوژی شامل طراحی شبکه عامل‌ها، طراحی

عامل و طراحی ساختار زیربنای عامل است [۲۹].

- متدولوژی ADELFE که در آن مراحل نیازمندی‌های اولیه، نیازمندی‌های نهایی، طراحی کلی و تست در نظر گرفته شده است [۹].
- متدولوژی PASSI که بر پایه نقش، تعامل و ارتباط بنا شده است [۱۰] .
- متدولوژی ODCA [۳۰] که در آن پنج دیدگاه Enterprise، اطلاعات، محاسبات، تکنولوژی و مهندسی [۳۱] مدل می‌شوند.

۳- مقایسه رویکرد شی‌گرا و عامل‌گرا

مهندسی نرم‌افزار عامل‌گرا تکاملی از نرم‌افزار شی‌گرا است. به‌عبارت دیگر می‌توان گفت عامل‌ها از اشیا مشتق شده‌اند [۳۲].

لیند در سال ۲۰۰۱ [۳۳] به مقایسه سیستم‌های شی‌گرا و عامل‌گرا در پنج سطح سخت‌افزار، تئوری، زمان اجرا[1]، زبان‌های برنامه‌نویسی[2] و زبان‌های طراحی پرداخته است که در ادامه برخی از تفاوت‌ها بیان شده است: الف) اشیا دارای سازمان متمرکزی هستند، در حالی که عامل‌ها محاسبه‌ی توزیعی را مجاز می‌دانند.

ب) اشیا در یک سیستم، بیشتر با یکدیگر یکپارچگی دارند، در حالی که یکپارچگی عامل‌ها ضعیف است.

پ) عامل‌ها نمی‌توانند مانند اشیا آزادانه ایجاد شده و از بین بروند.

ت) رفتار و ساختار شی ثابت است و تغییر نمی‌کند، در حالی که عامل‌ها از تجربه‌هایشان یاد می‌گیرند و رفتارشان را تغییر می‌دهند.

ث) تعاملات اشیا اکثراً توسط شی دیگر درخواست می‌شود، در حالی که تعاملات عامل‌ها شامل واکنش به وقایع محیط‌شان یا درخواست از سایر عامل‌هاست.

ج) تعاملات اشیا معمولاً همگام است در حالی‌که تعاملات عامل‌ها معمولاً ناهمگام است.

چ) عامل کپسوله‌بندی قوی‌تری از شی دارد.

البته از آن‌جایی که عامل‌ها از اشیا مشتق شده‌اند، می‌توان گفت شباهت‌هایی نیز بین عامل‌ها و اشیا وجود دارد. با وجود این تفاوت‌ها و شباهت‌ها می‌توان پارامترهایی از هر دو رویکرد را به هم نگاشت کرد.

جدول (۱) یک نگاشت نوعی بین رویکردهای شی‌گرا و عامل‌گرا را نشان می‌دهد.

جدول۱- نگاشت بین رویکردهای عامل‌گرا و شی‌گرا [۳۳]

رویکرد شی‌گرا	رویکرد عامل‌گرا
کلاس انتزاعی	نقش کلی[3]
کلاس	نقش مخصوص یک قلمرو[4]
متغیرهای کلاس	دانش، باور
متد	قابلیت
وراثت[5]	انقیاد نقش[6]
نمونه‌سازی	نقش خاص+ دانش فردی
ترکیب	عامل‌های هولونیک
فرخوانی متد	تبادل پیغام
همکاری	تعامل

بر اساس جدول ۱ می‌توان گفت که رویکرد عامل‌گرا برای تمامی قابلیت‌های رویکرد شی‌گرا راهکار ارائه نموده است البته راهکارهایی که، برای تحلیل و طراحی سیستم‌های مبتنی بر عامل مناسب است.

۴- معیارها و روش ارزیابی

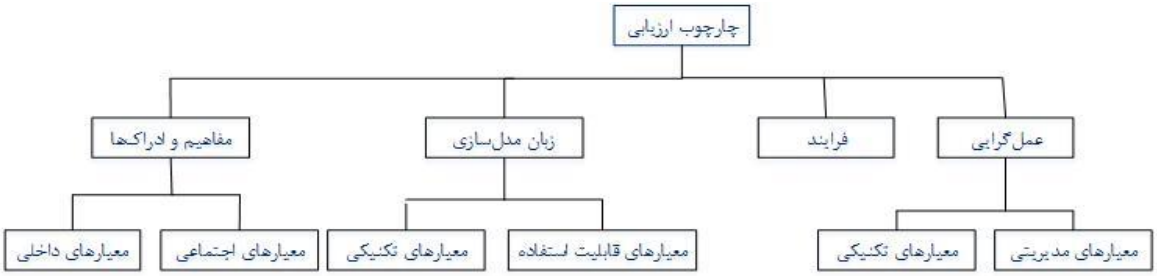
اولین و مهمترین مرحله از هر ارزیابی مشخص نمودن اهداف آن می‌باشد [۳۴]. در این مقاله دو هدف عمده از ارزیابی دنبال می‌شود: هدف اول مشخص نمودن نقاط قوت و ضعف و همچنین تشابه‌ها و تفاوت‌های متدولوژی‌های مورد بررسی به منظور بهبود و توسعه‌ی سیستم‌های نرم‌افزاری عامل‌گرا می‌باشد و هدف دوم پیشنهاد یکسان‌سازی مقدماتی از متدولوژی‌های عامل‌گرا به‌وسیله ترکیب نقاط قوت و همچنین فراهم آوردن پیشنهاداتی برای بهبود محدودیت‌های جاری در متدولوژی‌ها را شامل می‌شود.

پس از این مرحله، چارچوب ارزیابی که شامل ویژگی‌ها، خصوصیات و معیارها می‌باشد، ایجاد می‌گردد. ویژگی‌ها و معیارهای استفاده شده در این مقاله بر اساس کارهای موجود برای مقایسه متدولوژی‌های OOSE[7] و نیز کارهای انجام شده بر روی متدولوژی‌های عامل‌گرا می‌باشد [۳۷-۳۴]. ارزیابی انجام شده، روی ویژگی‌های تکنیکی متدولوژی‌ها متمرکز شده است. از این رو کشف و مشخص

[1] Run-time
[2] Programming Language
[3] Generic role
[4] Domain-Specific role
[5] Inheritance
[6] Role binding
[7] Object Oriented Software Engineering

نمودن مدل‌ها و پروسه‌های مشترک در دو متدولوژی مورد مقایسه و نیز جنبه‌های متمایز آن‌ها مورد بررسی قرار گرفته است. این موضوع می‌تواند نقش مهمی برای توسعه نسل بعدی متدولوژی‌های عامل‌گرا ایفا نماید.

در ادامه یک چارچوب ارزیابی متدولوژی، با مقایسه بر اساس خصوصیات ارائه می‌گردد. این چارچوب شامل مجموعه‌ای از معیارها و ضوابط می‌باشد که خصوصیات انحصاری AOSE را نیز در بر دارد. شکل (۱) چارچوب کلی ارزیابی متدولوژی‌های عامل‌گرا را نشان می‌دهد.



شکل ۱- چارچوب کلی ارزیابی متدولوژی‌های عامل‌گرا

بر اساس شکل (۱)، چهار جنبه مختلف ارزیابی شامل مفاهیم، زبان مدل نمودن، فرآیندها و عمل‌گرایی بیان شده است. هر کدام از شاخص‌های بیان شده، به معیارهایی تقسیم می‌شوند که به طور کامل در [۳۶] تشریح شده است. به منظور ارزیابی متدولوژی‌های مورد بررسی بر مبنای چهار شاخص، پرسشنامه‌هایی در اختیار خبرگان قرار گرفت، نتایج حاصل از پاسخ خبرگان بر اساس روش دلفی‌فازی مورد تحلیل و آنالیز قرار گرفت. در ادامه در مورد نتایج پایایی پرسشنامه و نتایج ارزیابی متدولوژی‌ها مورد بررسی، بحث شده است.

۴-۱- روایی، پایایی و مقیاس اندازه‌گیری پرسشنامه

برای تعیین روایی، ابتدا آزمون مقدماتی در مورد پرسشنامه به‌عمل آمد. بدین منظور پرسشنامه اولیه در اختیار پانزده نفر از استادان متخصص در زمینه مهندسی نرم‌افزار، قرار گرفت. پس از جمع‌آوری پرسشنامه‌ها، سؤالات مبهم و غیرمرتبط شناسایی، و آن دسته از سؤالاتی که قابل اصلاح بودند، ویرایش مجدد، و در پرسشنامه نهایی منظور شد و آن دسته از سؤالاتی که کاملا بی‌ارتباط شناخته شد، از پرسشنامه حذف گردید. برای تعیین پایایی پرسشنامه نیز از ضریب آلفای کرونباخ بهره برده شد. در این تحقیق مقدار محاسبه شده برای آلفای کرونباخ برابر ۸۴۲/ . بوده است، که بر اساس اصول روش تحقیق مقدار مطلوبی تلقی می‌شود. همچنین مقیاس اندازه‌گیری مورد استفاده در این تحقیق، طیف پنج امتیازی لیکرت بوده است.

۴-۲-یافته‌ها و نتایج ارزیابی

در بخش قبل، روش و چارچوب استفاده شده به منظور ارزیابی تشریح شد. در اصول چارچوب ارزیابی استفاده شده، شاخص‌هایی در چهار قسمت تعریف گردید. در این بخش با استفاده از این معیارها نتایج ارزیابی ارائه می‌گردد. این نتایج بر اساس روش دلفی‌فازی آنالیز و تحلیل شده است. در مورد بعضی از معیارها با توجه به ماهیتی که دارند، تنها وجود یا عدم وجود آن‌ها در متدولوژی بررسی شده است.

- شاخص اول؛ مفاهیم و ادراک‌ها

- نوع ویژگی۱: ویژگی‌های داخلی

۱- خودمختاری: خودمختاری به‌عنوان یکی از خصوصیات کلیدی، در عامل‌ها مورد نظر می‌باشد. این ویژگی، عامل‌ها را از دیگر موجودیت‌ها مانند اشیا متمایز می‌نماید. بر اساس ارزیابی‌های انجام شده، هر دو متدولوژی عامل‌گرا دارای این ویژگی می‌باشند. سطوح پشتیبانی برای پارامتر خودمختاری در هر دو آن‌ها مناسب است (از متوسط تا خوب). هر دو متدولوژی انواع پشتیبانی‌ها برای ویژگی‌های خودکنترلی عامل را ارائه می‌دهند، و اعمال و تسهیلاتی را در عامل‌ها قرار می‌دهند. به‌علاوه نمودار هم‌اجرایی در متدولوژی PASSI، امکان مدل نمودن عامل‌ها را، صرف نظر از محیط و موجودیت‌های دیگر، فراهم می‌آورد.

۲- گرایش‌های ذهنی: متدولوژی ADELFE استفاده از گرایش‌های ذهنی را در مدل داخلی عامل‌ها به‌خوبی پشتیبانی می‌کند و دانش عامل‌ها را از محیط نمایش می‌دهد، همچنین اهداف عامل‌ها در این متدولوژی مدل‌سازی می‌شود. در مقابل متدولوژی PASSI پشتیبانی

ضعیف‌تری از گرایش‌های ذهنی عامل‌ها به‌عمل می‌آورد.
۳- هدف‌گرا و واکنش‌دار بودن: ارزیابی نتایج اندازه‌گیری این دو ویژگی مشکل می‌باشد. این ویژگی به وسیله بعضی از متدولوژی‌ها به‌خوبی پشتیبانی می‌شوند (متوسط برای متدولوژی PASSI و خوب برای متدولوژی ADELFE). مشابه پارامتر قبل، این دو متدولوژی ، اهداف را به‌دست آورده و پس از آن فعالیت‌هایی را برای دستیابی به اهداف اجرا می‌کنند.
۴- هم‌اجرایی: پشتیبانی از هم‌اجرایی، میان متدولوژی‌ها، خیلی متنوع می‌باشد. از کم تا خوب در نوسان است. از دیدگاه افراد متخصص و خبره متدولوژی PASSI بهترین متدولوژی در پوشش این مدل می‌باشد. در این متدولوژی، فعالیت‌های یک نقش منفرد، می‌تواند به‌طور هم‌اجرا تعریف شود.
۵- در محیط واقع شدن: پشتیبانی از معیار در محیط واقع شدن، در میان متدولوژی‌های مورد بررسی از متوسط تا زیاد تغییر می‌کند و از دیدگاه افراد متخصص و خبره متدولوژی ADELFE از این خاصیت بیشتر از متدولوژی PASSI پشتیبانی می‌کند.

- نوع ویژگی۲: ویژگی‌های اجتماعی

۱- روش‌های همکاری و کارتیمی: برای پارامتر روش‌های همکاری و کارتیمی دو زیر پارامتر برنامه‌ریزی چندعامله و کارتیمی مورد ارزیابی قرار گرفت. در متدولوژی‌های ADELFE و PASSI ایجاد کنندگان آن‌ها[۹][۱۰]، اظهار می‌دارند که این متدولوژی‌ها، از همکاری‌های عامل‌گرای عمومی حمایت می‌کنند و هر نوع همکاری را می‌توان از آن‌ها به‌دست آورد. اما از دیدگاه افراد متخصص و خبره هیچ یک از دو زیر پارامتر به‌طور صریح توسط متدولوژی‌های مورد بررسی پوشش داده نمی‌شود.
۲- پروتکل: متدولوژی ADELFE با پروتکل تحلیل‌گر خود یعنی نمودار کلاس ارتباطی، به‌روشنی از متدولوژی PASSI جلوتر است. متدولوژی PASSI، مدل خاصی برای نمایش پروتکل‌ها فراهم نمی‌آورد، ولی تعامل‌های میان عامل‌ها را در سطوح بالا ارائه می‌دهد.
۳- زبان ارتباطی: این ویژگی از دیدگاه افراد متخصص و خبره در هر دو متدولوژی وجود دارد. از آن‌جا که تعامل میان عامل‌ها، دارای سطوحی از دانش است. هر دو متدولوژی دارای هدف Speech Act، به‌عنوان زبان ارتباطی می‌باشند.

- شاخص دوم؛ زبان مدل‌سازی

- نوع ویژگی۱: ویژگی‌های قابلیت استفاده

۱- قابلیت فهم و روشن بودن: این دو معیار چگونگی روشن بودن نمادها و همچنین چگونگی تعریف مناسب ترکیب نحوی مدل‌ها و سمبل‌ها را مشخص می‌کند. نمادهای فراهم شده توسط هر دو متدولوژی به‌خوبی قابل فهم است.
۲- رسایی: تعداد مدل‌های ایستا و پویا و نیز دیدگاه‌های مختلفی که سیستم مقصد را نمایش می‌دهد، محک خوبی برای این معیارها می‌باشند. متدولوژی ADELFE جنبه‌های دینامیک سیستم را مدل کرده و با پروتکل‌ها به‌خوبی سر و کار دارند. متدولوژی PASSI برای پروتکل‌ها با مدل‌سازی دینامیک سیستم به استثناء بعضی از پشتیبانی‌ها در سطوح طراحی تفضیلی، حمایت قدرتمندی ارائه نمی‌دهد. گرچه سمبل‌ها در متدولوژی ADELFE به‌طور مناسبی با معنی به نظر می‌آیند. اما این متدولوژی دیدگاه‌های مختلفی از سیستم مقصد فراهم نمی‌آورد.
۳- سادگی استفاده: بر اساس نظر افراد متخصص و خبره هر دو متدولوژی دارای نمادها و سمبل‌هایی می‌باشند که استفاده و یادگیری آن‌ها ساده می‌باشد.

- نوع ویژگی۲: ویژگی‌های تکنیکی

۱- سازگاری: بر حسب چک نمودن سازگاری، متدولوژی‌های مورد بررسی دارای سطوح متفاوتی می‌باشد. متدولوژی ADELFE به‌خوبی از آن پشتیبانی می‌کند. در حالی که متدولوژی‌های PASSI از آن پشتیبانی چندانی نمی‌کند. از دیدگاه افراد متخصص و خبره در دسترس بودن ابزار پشتیبانی، علت این پشتیبانی ضعیف می‌باشد.
۲- قابلیت پیگیری: مشابه معیار سازگاری، متدولوژی ADELFE پیشرو حمایت از این ویژگی می‌باشد. این متدولوژی ارتباط روشنی میان مدل‌های خود ایجاد می‌نماید. برای نمونه اهداف، نقش‌ها، عامل‌ها و اعمال با هم پیوند می‌خورند. این چنین اتصالاتی برای توسعه‌دهنده اجازه به‌دست آوردن مدل‌های طراحی (به طور مثال معماری داخلی عامل‌ها) را می‌دهد.
۳- پالایش: پشتیبانی دو متدولوژی از این معیار مناسب نمی‌باشد. از دیدگاه افراد متخصص و خبره این موضوع انعکاس دهنده این حقیقت است که زبان مدل‌سازی دو متدولوژی در فرآیند توسعه‌ی آن‌ها مجتمع نشده است. در واقع توسعه‌دهندگان در حرکت میان فازها، برای اضافه

نمودن جزئیات در مدل ساخته شده، آزاد می‌باشند.
۴- قابلیت استفاده مجدد: هیچ‌کدام از دو متدولوژی مورد بررسی، صریحا تکنیک‌هایی برای حمایت از طراحی و استفاده از مؤلفه‌های قابل استفاده، به‌عمل نمی‌آورند. همچنین استفاده مجدد مؤلفه‌های موجود در هر متدولوژی، مشخص نمی‌باشد.

- شاخص سوم؛ فرآیند

۱- اصول توسعه: از نقطه نظر سیکل حیات توسعه نرم‌افزار، هر دو متدولوژی مورد بررسی دارای طراحی معماری می‌باشند. مرحله تست و خطایابی تنها در متدولوژی ADELFE مورد پشتیبانی است. ADELFE تنها متدولوژی‌ای می‌باشد، که توسعه عامل‌ها را توصیف می‌کند و این موضوع قسمتی از فاز طراحی سیستم می‌باشد. بر حسب دیدگاه توسعه، متدولوژی ADELFE از روش بالا به پایین و پایین به بالا پشتیبانی می‌کند. در حالی که متدولوژی‌های PASSI مناسب روش بالا به پایین می‌باشد.
۲- مراحل فرآیند: این مراحل در فازهای تحلیل نیازمندی‌ها و طراحی در متدولوژی PASSI به خوبی توصیف می‌شود. این موضوع در حالی است که طراحی تفضیلی در متدولوژی ADELFE به خوبی مستند نشده است. دلیل این موضوع از دیدگاه افراد متخصص و خبره محدودیت منابع در متدولوژی ADELFE است.
۳- مفهوم پشتیبانی توسعه: چندین مفهوم اصلی توسعه مانند نمونه‌سازی و استفاده مجدد مؤلفه‌ها وجود دارد. از دیدگاه افراد متخصص و خبره هیچ کدام از دو متدولوژی به روشنی موضوعات مرتبط با نمونه‌سازی در فرآیند یا ایجاد مؤلفه‌های قابل استفاده مجدد را در بر ندارد.
۴- رهنمودهای تضمین کیفیت و تخمین: به علت عدم تکامل متدولوژی‌های عامل‌گرا، از دیدگاه افراد متخصص و خبره اظهار نظر دقیقی نمی‌توان راجع به این پارامتر ارائه نمود.

- شاخص چهارم؛ عمل‌گرایی

- نوع ویژگی۱: ویژگی‌های مدیریتی

۱- هزینه: هزینه به‌دست آوردن متدولوژی‌ها و ابزارهای پشتیبانی‌کننده آن‌ها برای سطح آگاهی و کاربردهای جاری و قابلیت دسترسی به آن‌ها، تقریبا برای همه متدولوژی‌ها مجانی است و مستندات آن‌ها قابل دسترس است.
نوع ویژگی۲: ویژگی‌های تکنیکی
۱- قابلیت کاربرد دامنه: از دیدگاه افراد متخصص و خبره هیچ محدودیتی برای دامنه کاربردهای این دو متدولوژی وجود ندارد. این دامنه‌ها، برای سیستم‌های مبتنی بر عامل، که دارای نرم‌افزارهای خودمختار، قابل اعتماد و قدرتمند می‌باشند، مناسب است.
۲- مقیاس‌پذیر بودن: این پارامتر توسط هیچ یک از دو متدولوژی پوشش داده نمی‌شود. از دیدگاه افراد متخصص و خبره هیچ یک از دو متدولوژی راجع به چگونگی معرفی مؤلفه‌های جدید در سیستم راهکاری ارائه نداده‌اند.
۳- توزیع‌پذیری: از دیدگاه افراد متخصص و خبره، متدولوژی ADELFE به‌طور ضمنی از پارامتر توزیع‌پذیری پشتیبانی می‌کند. از دیدگاه آن‌ها متدولوژی PASSI حالتی استثناء دارد، به‌طوری که مرحله طراحی سیستم در این متدولوژی، به توسعه‌دهنده امکان طراحی و قرار دادن عامل‌ها در شبکه را می‌دهد. دلیل این موضوع وجود مدل استقرار در این متدولوژی می‌باشد.

۵- مطالعه موردی: سیستم اتاق بازرگانی

در این مقاله، سیستم اتاق بازرگانی در قالب بستر اینترنتی تعریف شده و بخش‌های مختلفی که برای عملیات تجاری لازم است را ارائه می‌دهد. در این سیستم هئیت‌های تجاری با مراجعه به سایت، می‌توانند اطلاعات مربوط به بخش‌های سرمایه‌گذاری را مشاهده کنند و پس از بررسی نسبت به سرمایه‌گذاری در بخش مورد نظر تصمیم بگیرند. هنگام درخواست برای سرمایه‌گذاری بیان شرایط، اولویت‌ها، میزان سرمایه‌گذاری و برنامه و نحوه فروش بین‌المللی محصول (در صورت وجود) بایستی توسط هئیت مورد نظر مشخص شود، همچنین داشتن دو سوم مبلغ کف مشخص شده توسط اتاق بازرگانی در بخش مورد نظر برای سرمایه‌گذاری نیز بایستی در حساب ارائه شده توسط هئیت وجود داشته باشد. هئیت تجاری می‌تواند در درخواست خود، شروطی را لحاظ نماید. به طور مثال با توجه به قیمت نوسانی بازار میزان سودهای تعلق گرفته را برای ماه‌های خاص تغییر دهد، و این موضوع به نوعی استراتژی هئیت تجاری محسوب می‌شود. از طرف دیگر در صورتی که هئیت بخواهد بخشی از سرمایه‌گذاری را به‌صورت مدت‌دار انجام دهد، می‌تواند اعلام کند. پس از انجام درخواست تجارت و سرمایه‌گذاری از سوی هئیت تجاری، کارشناسان اتاق بازرگانی به بررسی صحت و سقم درخواست‌های انجام شده

می‌پردازد و در صورتی که اطلاعات کامل باشند، آن‌ها را در سیستم وارد می‌کند. پس از ورود درخواست، سیستم با توجه به شرایط، اولویت‌ها و میزان سرمایه‌گذاری بر اساس خطوط کلی نظام اقتصادی، درخواست‌ها را سامان می‌دهد و با توجه به محدودیت‌های اعمال شده توسط طرفین، هئیت‌های تجاری می‌توانند در طول ۷۲ ساعت آینده لیستی از مواردی را که با توجه به شرایط درخواستی در آن‌ها امکان سرمایه‌گذاری وجود دارد را مشاهده نمایند. پس از مشاهده لیست، هئیت‌ها می‌توانند بر اساس اولویت-های خود، مواردی را که مد نظرشان است را انتخاب و بر روی آن‌ها درخواست بدهند و یا انصراف خود را اعلام کنند. در صورتی که درخواستی از سوی هئیت تجاری داده شود، متناسب با برنامه‌های اتاق بازرگانی، تاریخی برای سفر هئیت مذکور به کشور و بازدید از پروژه‌های مورد نظر تعیین می‌شود. هئیت تجاری می‌تواند تا یک ماه قبل از تاریخ تعیین شده، تاریخ ورود خود را تغییر دهد و تاریخ جدید مد نظر خود را به اتاق بازرگانی اعلام کند. اتاق بازرگانی با توجه به تاریخ تعیین شده می‌تواند اعلام نظر کند و در صورت تأیید نهایی برنامه سفر هئیت تجاری به کشور تنظیم خواهد شد.

۶- معرفی متدولوژی AP

AP یک متدولوژی توسعه نرم‌افزاری عامل‌گرا است که به‌صورت ترکیبی از دو متدولوژی ADELFE و PASSI و همچنین مدل‌های پیشنهادی کنشگر گسترش‌یافته، هدف مجزا، قابلیت و برنامه‌ریز و مدل سیستم گسترش‌یافته به دست آمده است. در تعیین مراحل فازهای متدولوژی AP پنج ایده اصلی مورد استفاده قرار گرفته است که عبارتند از: الف) تصور و درک از عامل و همه تصورات ذهنی (اهداف، برنامه‌ریزی‌ها و موارد مشابه) در مراحل فازهای توسعه نرم‌افزار از تحلیل اولیه تا طراحی مورد استفاده قرار گرفته است. ب) به منظور درک کامل از کنشگرهای سیستم و تعامل میان آن‌ها، مدل کنشگر گسترش‌یافته به فاز تحلیل در متدولوژی AP اضافه شده است. پ) به منظور تعیین اهداف جزئی موجود در سیستم که همه بخش‌های آن را پوشش می‌دهد، مدل هدف مجزا در نظر گرفته شده است. ت) برای تعیین قابلیت‌های عامل‌ها به‌صورت جزئی و همچنین چگونگی اجرای این قابلیت‌ها به‌وسیله‌ی عامل‌ها، مدل سرویس در فاز طراحی متدولوژی AP پیشنهاد شده است. ث) به منظور درک کلی و جزئی محدوده سیستم، از مدل سیستم گسترش‌یافته در متدولوژی استفاده شده است.

تحلیل و طراحی عامل‌گرا مجموعه‌ی زیادی از مفاهیم را شامل می‌شود و این موضوع فهم همه‌ی جنبه‌های مدل تحلیل و طراحی را از دیدگاه خاص مشکل می‌سازد. به همین دلیل در متدولوژی AP، تعدادی مدل که روی جنبه‌های مختلف تأکید دارند، تعریف شده است. این مدل-ها جنبه‌های مختلفی را در برمی‌گیرند، اما به تنهایی کامل نیستند، بنابراین با کنار هم قرار دادن آن‌ها، یک دیدگاه کامل و قابل فهم نسبت به سیستم حاصل می‌شود. همان-طور که در بخش (۴-۲) تشریح شد، در انتخاب مدل‌های متدولوژی AP از یک چارچوب ارزیابی جامع که چهار ناحیه‌ی عمده مهندسی نرم‌افزار عامل‌گرا شامل مفاهیم و ادراک‌ها، زبان مدل‌سازی، فرآیند و عمل‌گرایی را پوشش می‌دهد، استفاده شده است. بر اساس این چارچوب و پارامترهای آن میزان پشتیبانی هر یک از متدولوژی‌های مورد بررسی و همچنین مدل‌های پیشنهادی کنشگر گسترش‌یافته، هدف مجزا، قابلیت و برنامه‌ریز و مدل سیستم گسترش‌یافته از پارامترهای ارزیابی بر اساس نظر خبرگان مورد بررسی قرار گرفته است. بنابراین مدل‌های استفاده شده در متدولوژی AP مفاهیم عامل‌گرایی را تا حد زیادی پوشش می‌دهد و این مدل‌ها هیچ‌گونه هم‌پوشانی با یکدیگر ندارند. در ادامه به منظور سنجش توانایی متدولوژی پیشنهادی، از آن در یک مطالعه موردی که در بخش پنجم تشریح گردید استفاده شده است.

۶-۱-تحلیل سیستم اتاق بازرگانی با استفاده از فاز-های متدولوژی پیشنهادی

در این قسمت روشی برای یکسان‌سازی دو متدولوژی ADELFE و PASSI به‌وسیله‌ی ترکیب نقاط قوت و نیز اجتناب از محدودیت‌های آن‌ها، ارائه می‌گردد. در حقیقت از برخی از قسمت‌های این متدولوژی‌ها، برای ایجاد متدولوژی جدید، با توجه به چارچوب ارائه شده در بخش‌های قبل استفاده می‌شود. استفاده از روش ترکیبی می‌تواند تأثیر مناسبی بر روی متدولوژی پیشنهادی به منظور پوشش حداکثری فعالیت‌های مهندسی نرم‌افزار عامل‌گرا داشته باشد و در توسعه‌ی نسل بعدی متدولوژی‌های عامل‌گرا مؤثر باشد. در این مقاله برای

یکسان‌سازی متدها از روش فرآیند توسعه استفاده می‌گردد و با توجه به اهمیت فازهای تحلیل و طراحی در ایجاد محصول نرم‌افزاری با کیفیت و قابلیت اطمینان بالا، بر روی این دو فاز تمرکز خواهیم نمود.

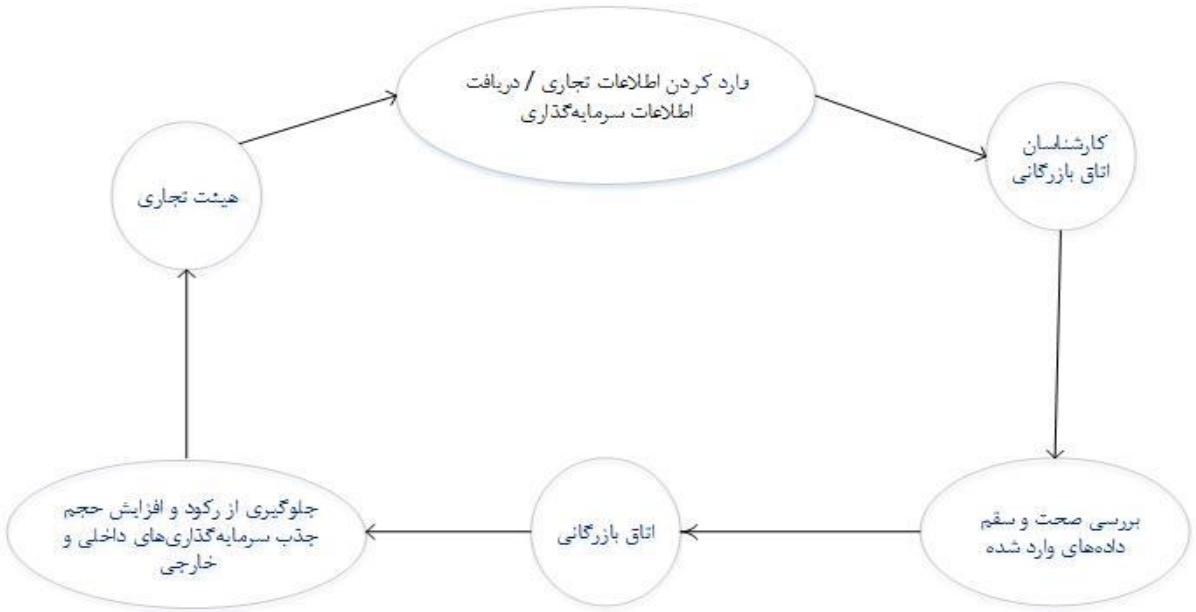
- فازهای متدولوژی پیشنهادی

- **فاز تحلیل**

متدولوژی پیشنهادی در فاز تحلیل شامل مدل کنشگر گسترش‌یافته، هدف مجزا، مدل دانش، مدل محیط و مدل نقش می‌باشد که این پنج مدل پشتیبانی قدرتمندی برای مشخص نمودن کنشگرها، اهداف، وظیفه و دانش سیستم، به‌دست آوردن محیط و تعریف نقش‌های کلیدی در سیستم ارائه می‌دهند. این مدل‌ها آگاهی توسعه‌دهندگان از نیازمندی‌های سیستم را ارتقا می‌دهند و ورودی‌هایی برای مرحله‌ی بعدی یعنی طراحی فراهم می‌آورند. تحلیل و طراحی در متدولوژی AP، با استفاده از زبان مدل‌سازی AUML [۳۸] صورت می‌گیرد. در مدل‌های دانش، محیط و نقش این متدولوژی ARA [۳۹] عمل می‌کند. بنابراین در بخش تحلیل تنها مدل‌های جدید کنشگر گسترش یافته و هدف مجزا تشریح می‌گردد.

- **مدل کنشگر گسترش‌یافته**

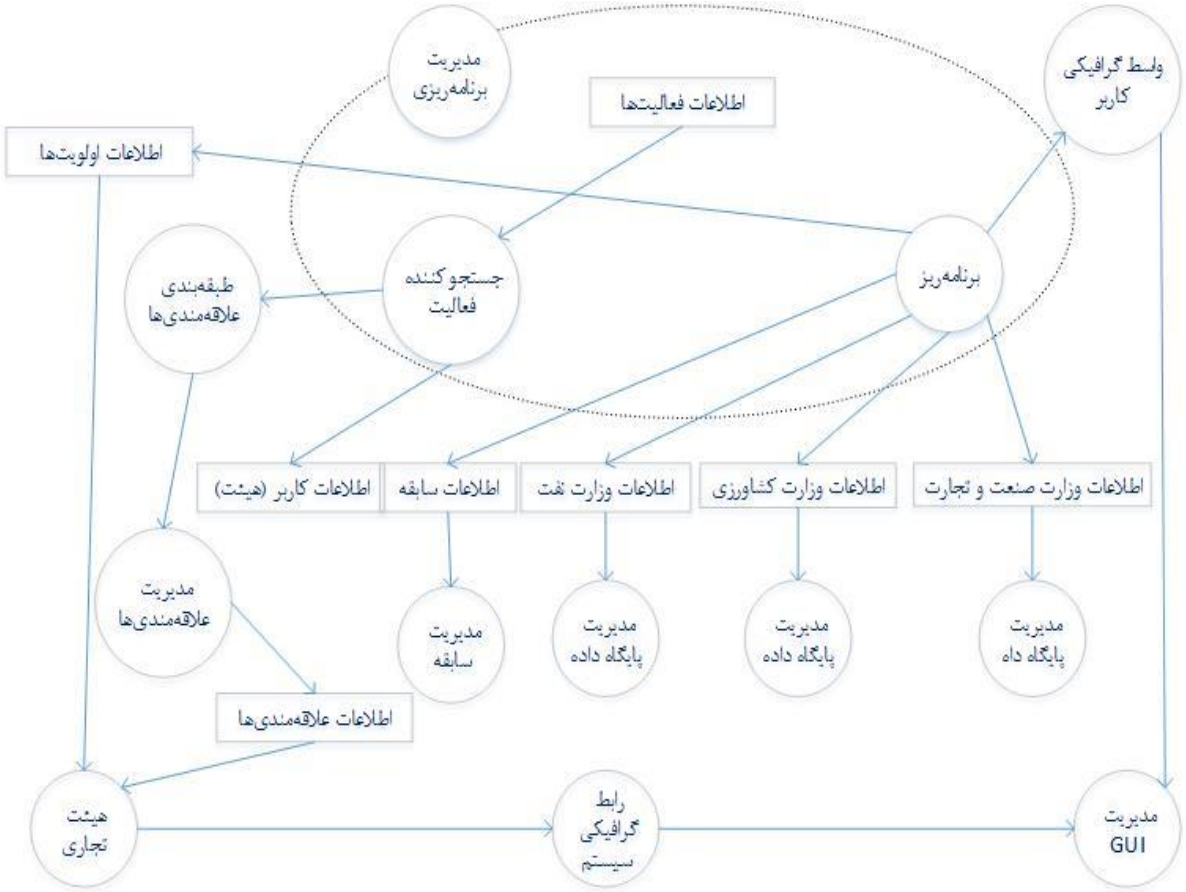
سازمان‌دهی و مشخص نمودن کنشگرهای سیستم، یکی از مراحل مهم در فاز تحلیل محسوب می‌شود. در این مدل، موجودیت‌های فیزیکی مؤثر در سیستم شناسایی می‌شوند. به منظور رسیدن به مدل کنشگر گسترش‌یافته ابتدا نمودار کنشگر سیستم را مدل می‌کنیم و پس از آن بر اساس مدل به دست آمده، مدل کنشگر گسترش‌یافته را به‌دست می‌آوریم. شکل (۲) مدل کنشگر سیستم اتاق بازرگانی را نشان می‌دهد.



شکل ۲- مدل کنشگر سیستم اتاق بازرگانی

همان‌طور که در شکل (۲) مشاهده می‌شود، سیستم اتاق بازرگانی دارای سه کنشگر اتاق بازرگانی، هئیت تجاری و کارشناسان اتاق بازرگانی است. هر کدام از کنشگرهای شناسایی شده در سیستم دارای اهدافی می‌باشند. هدف کنشگر هئیت تجاری، دسترسی به اطلاعات اولیه در مورد فرصت‌های سرمایه‌گذاری، هدف کارشناسان اتاق بازرگانی بررسی صحت و سقم اطلاعات وارد شده و هدف کنشگر اتاق بازرگانی رونق بخش تجاری، خروج از رکود و جذب حداکثری سرمایه‌گذاری‌های داخلی و خارجی است. با استفاده از نمودار کنشگر می‌توان ارتباط بین کنشگرهای سیستم را که یکی از ضعف‌های عمده در سایر متدولوژی‌های عامل‌گرا است را به روشنی نمایش داد.

بر اساس نمودار کنشگر شکل (۲)، یک دیدگاه کلی از کنشگرها و اهداف آن‌ها به‌دست آمد. حال در قالب شکل (۳) به تشریح جزئیات کنشگرها و اهداف آن‌ها بر اساس نمودار کنشگر گسترش‌یافته می‌پردازیم. این مدل می‌تواند میزان خطای تحلیل‌گران سیستم را (نسبت به سایر متدولوژی‌های عامل‌گرا) به حداقل برساند.

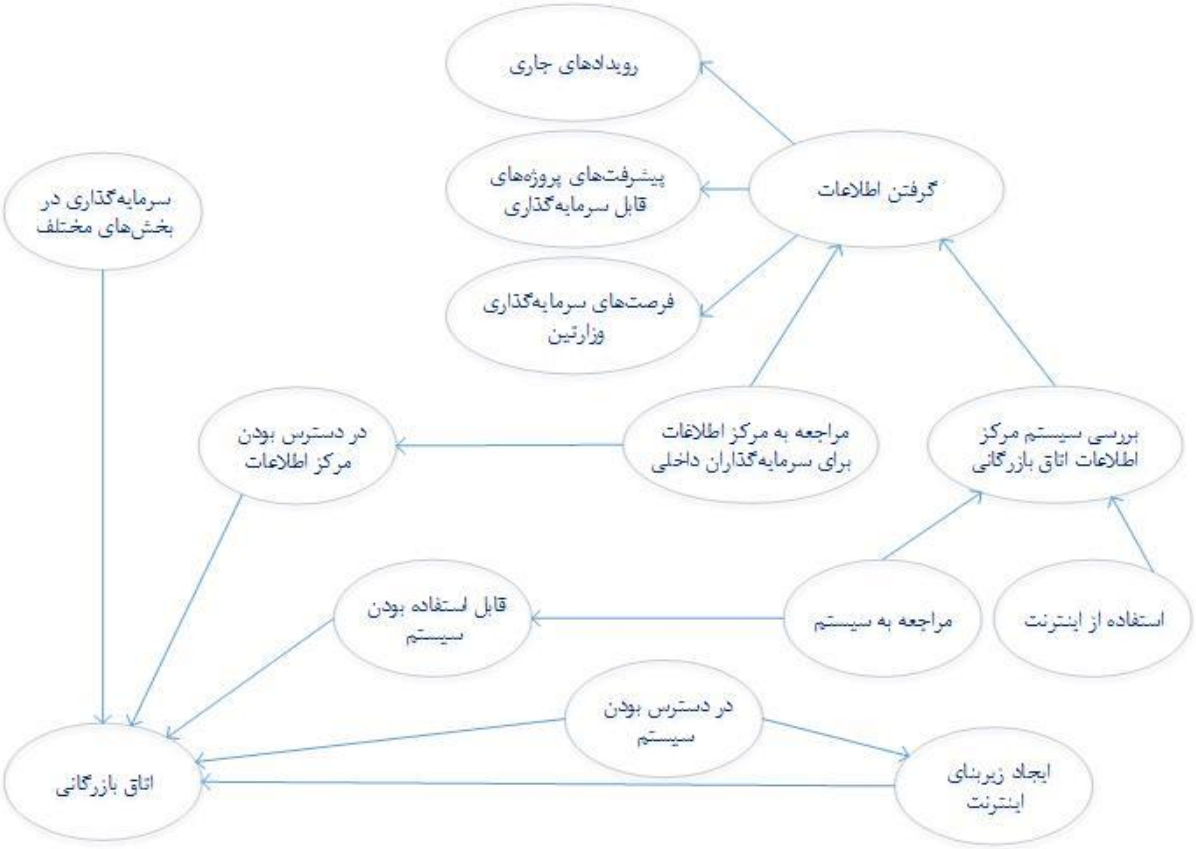


شکل ۳- مدل کنشگر گسترش‌یافته (مدیریت برنامه‌ریزی)

همان‌طور که در شکل (۳) مشاهده می‌شود، کنشگر مدیریت علاقه‌مندی، مسؤل دسته‌بندی علاقه‌مندی‌های هئیت تجاری است. این دسته‌بندی توسط مدیریت برنامه‌ریزی برای جستجوی فعالیت‌های مرتبط در پایگاه‌داده هئیت تجاری می‌باشد، بعلاوه مدیریت برنامه‌ریزی بر اساس اولویت‌های هئیت تجاری و مقایسه آن با سابقه ثبت شده از کنشگر مدیریت سابقه، فعالیت‌های معینی که می‌تواند، اجرا شود را به‌دست می‌آورد. مدیریت برنامه‌ریزی همچنین از اطلاعات وزارتین صنعت و تجارت، کشاورزی و نفت (بر اساس اولویت‌های نظام اقتصادی کشور)، فعالیت‌های مناسب برای اجرا را استخراج می‌کند (در واقع حیطه‌های مختلف برای سرمایه‌گذاری از پایگاه-داده‌های وزارتین واکشی می‌شود). همه اطلاعات جمع‌آوری شده به همراه اطلاعات محدودیت‌های تعیین شده از سوی هئیت تجاری و نیز اطلاعات وزارتین برای شکل‌دهی به برنامه‌ریزی عملی و منسجم، به‌کار می‌رود. رابطه‌ها از کاربر (هئیت تجاری) به سیستم و برعکس، به مدیریت GUI وابسته است. کنشگر مدیریت برنامه‌ریزی نیز به دو کنشگر جستجوکننده فعالیت‌ها و برنامه‌ریز شکسته می‌شود. جستجوکننده فعالیت‌ها، مسؤل جستجوی فعالیت‌هایی است که مربوط به علاقه‌مندی‌های کاربر می‌شود، و برنامه-ریز مسؤل ایجاد برنامه‌ریزی توسط اطلاعات به‌دست آمده از فعالیت‌های جستجو و اطلاعات دیگر کاربر (اولویت‌ها، پروژه‌ها و موارد مشابه) و نیز اطلاعات وزارتین و مدیریت سابقه می‌باشد.

- **مدل هدف جزئی**

سازمان‌دهی و مشخص نمودن اهداف، مرحله‌ی مهمی در استخراج نیازمندی‌ها است. در این مدل، اهداف جزئی سیستم در سطح‌های مختلف به طور جداگانه برای عوامل مؤثر در سیستم مشخص می‌شود. شکل (۴) مدل هدف جزئی هئیت تجاری را نشان می‌دهد. بر اساس شکل (۴)، هئیت تجاری برای دستیابی به اهداف خود می‌تواند از رویدادها، فرصت‌های سرمایه‌گذاری و پیشرفت‌های پروژه-های قابل سرمایه‌گذاری اطلاعات دریافت کند.



شکل ۴- مدل هدف جزئی برای هئیت تجاری

اطلاعات این محل‌ها و وقایع می‌تواند از اینترنت و یا کتابچه راهنما به‌دست آید. برای به‌دست آوردن این اطلاعات هئیت تجاری می‌تواند به سیستم اتاق بازرگانی مراجعه کند. برای این مراجعه دو پارامتر قابلیت دسترسی و قابلیت استفاده از سیستم بسیار مهم است، که هر دو پارامتر با داشتن زیربنای اینترنت و همچنین مدیریت مناسب اطلاعات می‌توانند مورد پوشش قرار بگیرند. می‌توان مدل هدف جزئی را برای اتاق بازرگانی نیز در نظر گرفت. هدف اصلی اتاق بازرگانی رونق بخش تجاری، خروج از رکود و جذب حداکثری سرمایه گذاری‌های داخلی و خارجی است. این هدف می‌تواند به زیر اهدافی شامل ارائه تبلیغات هدفمند، فراهم آوردن اطلاعات و توسعه سرویس مناسب تقسیم شود. زیر هدف فراهم آوردن اطلاعات شامل ایجاد سیستم اتاق بازرگانی و ساخت یک مرکز اطلاعات از سایر موقعیت‌های سرمایه‌گذاری وزارتین می‌باشد. از طرفی اتاق بازرگانی به منظور جذب حداکثری سرمایه بایستی تبلیغات هدفمند داخلی و بین‌المللی را افزایش دهد تا در خلال آن بتواند به توسعه بخش‌های خصوصی و دولتی بپردازد. زیر هدف سرویس مناسب نیز شامل توسعه و بهبود منابع انسانی، امکانات، امنیت و موارد مشابه می‌باشد. شکل (۵) مدل هدف جزئی اتاق بازرگانی را نمایش می‌دهد. در مرحله‌ی تحلیل، بر اساس مدل‌های کنشگر گسترش‌یافته و هدف جزئی نیازمندی‌های سیستم و همچنین قواعد و محدودیت‌های آن استخراج گردید. همچنین با استفاده از مدل‌های دانش، محیط و نقش که در متدولوژی AP دقیقا مشابه با متدولوژی ARA است و در [۳۹] به تفضیل مورد بررسی قرار گرفته است، میزان دامنه و تعامل بین نقش‌ها توسط عامل‌ها استخراج می‌گردد. در ادامه در مورد مراحل فاز طراحی در متدولوژی پیشنهادی بحث خواهیم نمود.

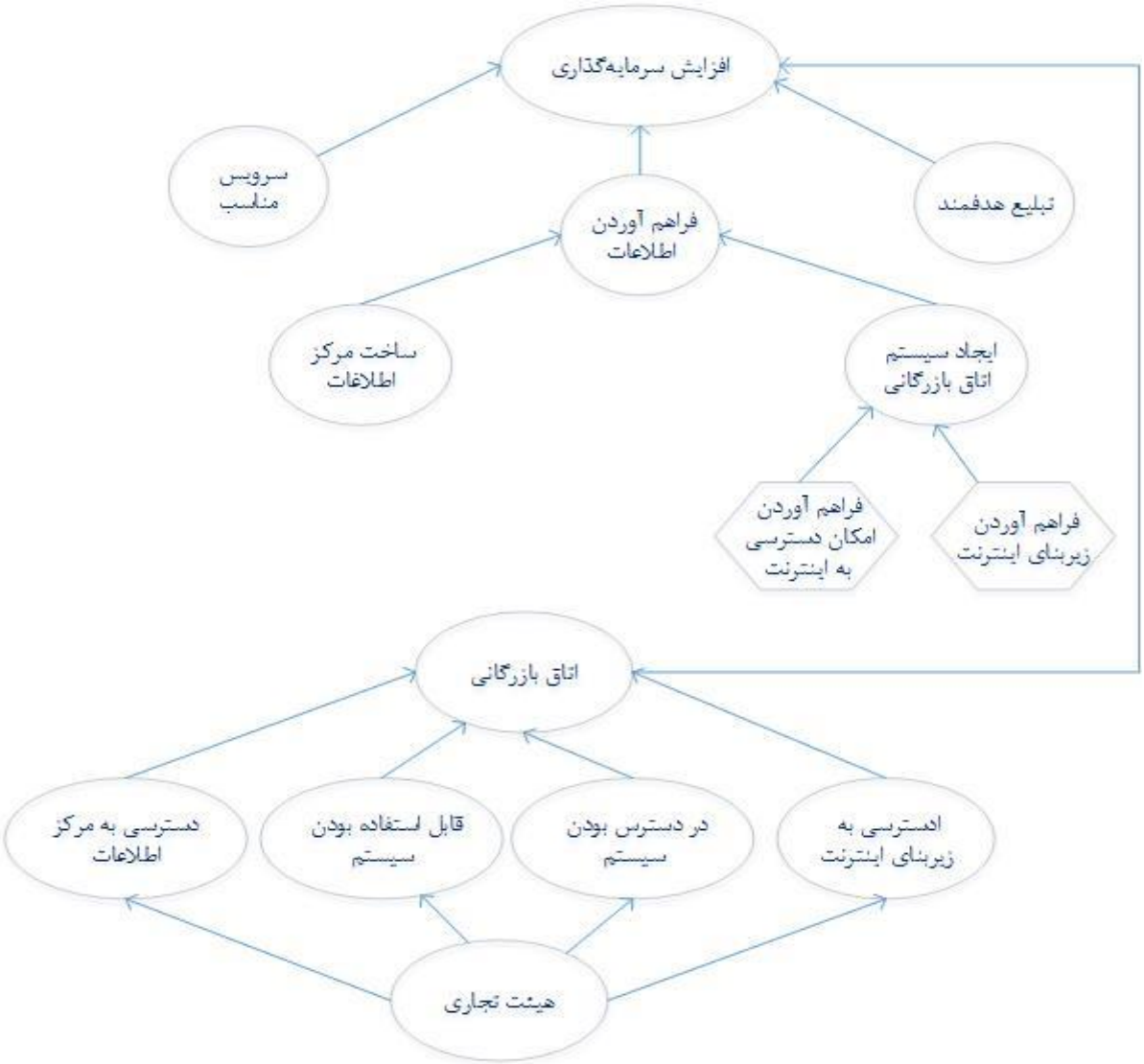
- **فاز طراحی**

متدولوژی پیشنهادی در فاز طراحی شامل: مدل عامل، مدل تعامل، مدل سرویس و مدل سیستم گسترش‌یافته است. در مدل عامل، نقش‌ها به هر عامل نگاشت می‌شوند، مدل تعامل مشخص کننده‌ی روابط بین عامل‌ها به منظور انجام نقش‌ها می‌باشد. مدل سرویس نیز به منظور مدل کردن قابلیت‌های عامل و برنامه‌ریزی برای چگونگی انجام این

قابلیت‌ها به‌کار می‌رود. مدل جدید سیستم گسترش یافته، جزئیات جامع و کاملی از سیستم و محدوده عملکرد آن به تیم تحلیل نشان می‌دهد. در فاز طراحی مدل عامل متدولوژی AP با مدل عامل متدولوژی ARA [۳۹] یکسان است اما مدل تعامل آن در جزئیات با متدولوژی ARA تفاوت دارد. بنابراین در فاز طراحی مدل‌های تعامل، سرویس و سیستم گسترش‌یافته را مورد بررسی قرار می‌دهیم. مدل‌های جدید سرویس و سیستم گسترش‌یافته دامنه کاربرد و همچنین قابلیت مدل‌سازی جزئیات متدولوژی AP را نسبت به سایر متدولوژی‌های عامل‌گرا افزایش داده است.

- **مدل تعامل**

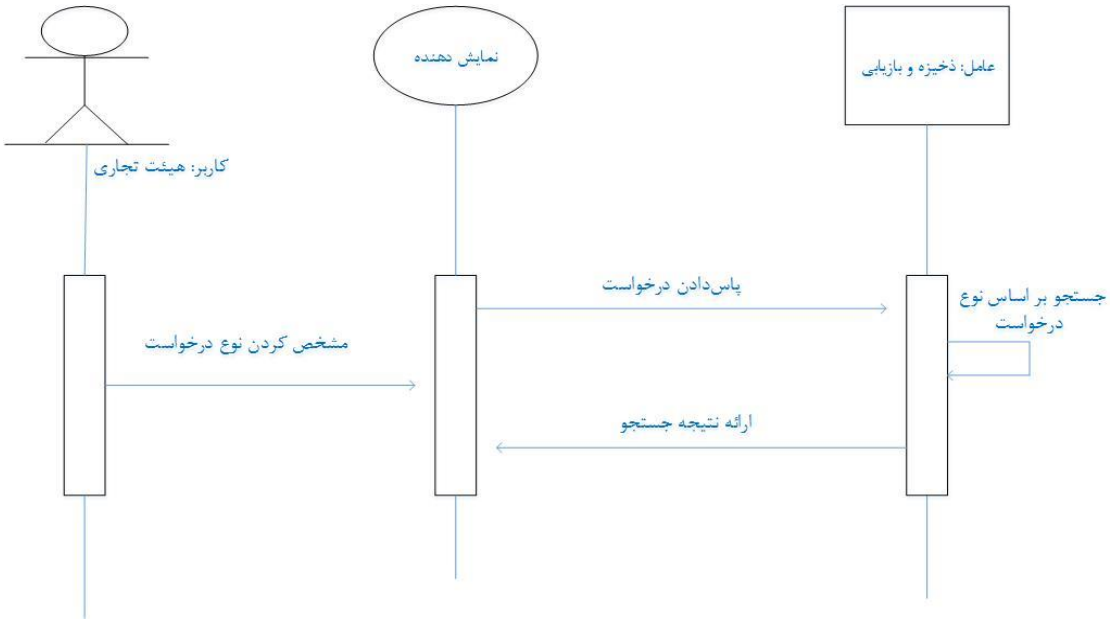
مدل تعامل، مشخص‌کننده روابط بین عامل‌ها برای انجام نقش‌ها می‌باشد، به‌عبارت دیگر نحوه انجام گرفتن نقش‌ها، به‌وسیله عامل‌ها با استفاده از مدل تعامل، مدل می‌شود. در مدل نقش پنج نقش برای سیستم تعیین گردید (بر اساس [۳۹])، اولین نقش، نقش جستجوی اطلاعات می‌باشد که مدل تعامل این نقش در شکل (۶) نمایش داده شده است.



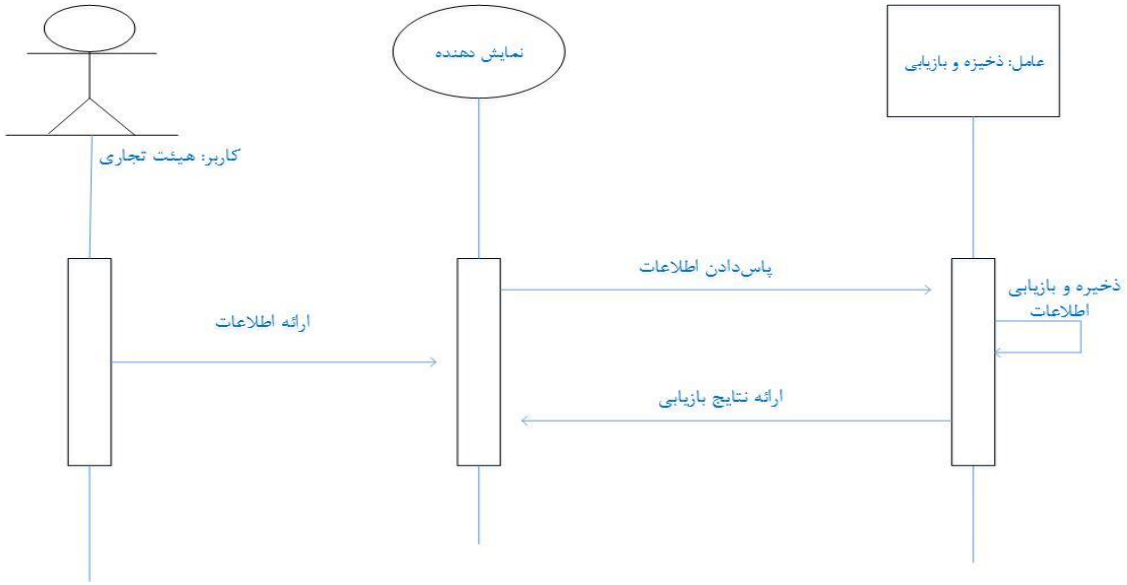
شکل ۵- مدل هدف جزئی برای اتاق بازرگانی

بر اساس شکل (۶)، کاربر(هئیت تجاری) ابتدا نوع درخواست خود را مشخص می‌کند، سپس این درخواست در اختیار عامل جستجو قرار می‌گیرد و این عامل، جستجو را بر اساس نوع درخواست از پایگاه‌داده سیستم انجام و نتیجه را بازگشت می‌دهد. یکی دیگر از نقش‌هایی که در مدل نقش تعیین گردید، نقش ذخیره و بازیابی می‌باشد،

که شکل (۷) نمودار تعامل این نقش را نشان می‌دهد.



شکل۶- مدل تعامل نقش جستجو اطلاعات

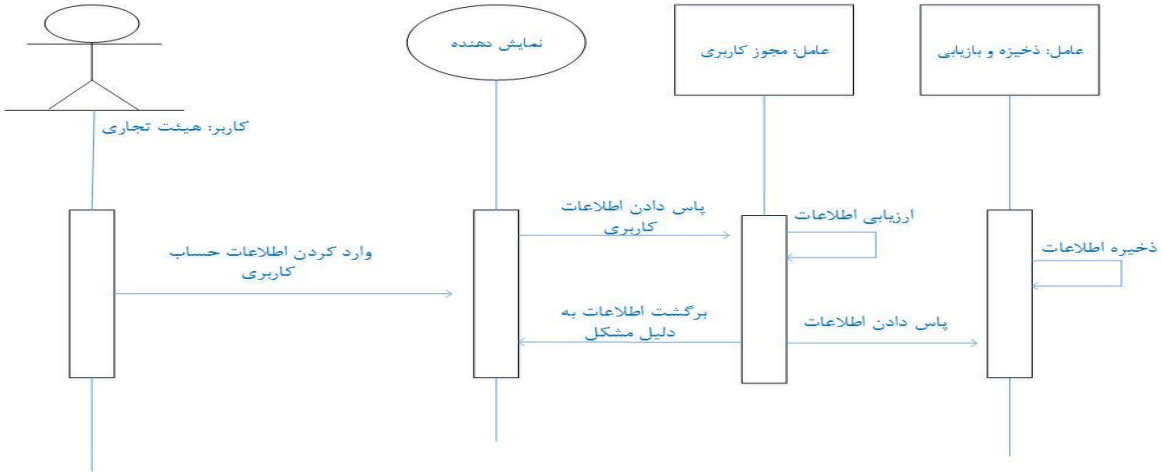


شکل۷- مدل تعامل نقش ذخیره و بازیابی

بر اساس شکل (۷)، کاربر(هئیت تجاری) ابتدا می‌بایست اطلاعاتی را که می‌خواهد ذخیره و بازیابی شوند را ارئه نماید، پس از آن این اطلاعات در اختیار عامل ذخیره و بازیابی قرار می‌گیرد و این عامل، عمل ذخیره و بازیابی را بر اساس نوع اطلاعات انجام می‌دهد و در صورتی که عملیات بازیابی باشد، نتیجه را بازگشت می‌دهد.

در ادامه در مورد مدل‌های تعامل نقش مجوز کاربری بحث خواهیم نمود. مدل تعامل نقش مجوز کاربری از دو حیطه، ذخیره اطلاعات کاربری و تأیید اعتبار اطلاعات کاربری مورد بررسی می‌باشد. از نظر ذخیره اطلاعات کاریری، کاربر

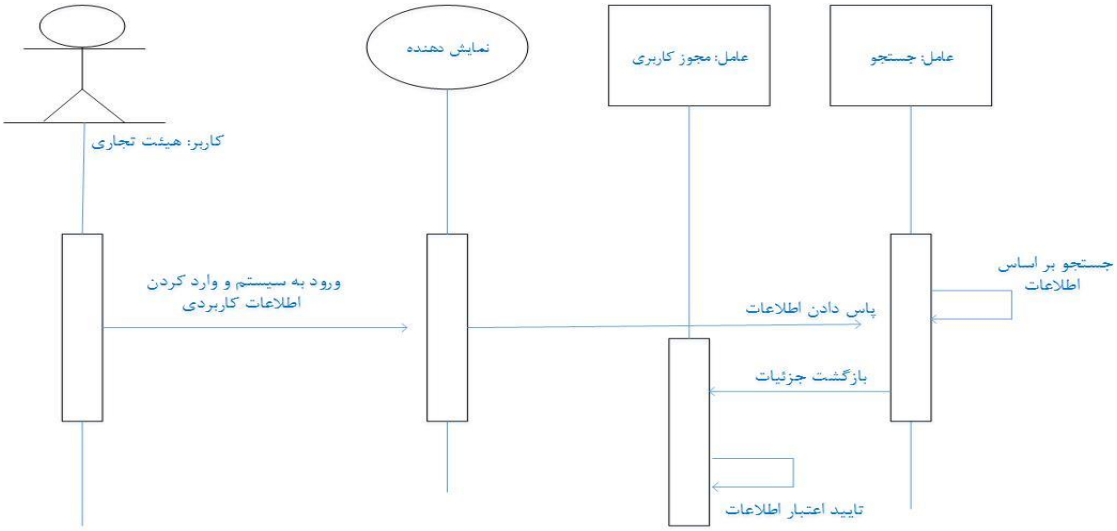
ابتدا اطلاعات حساب کاربری خود را وارد می‌نماید و این اطلاعات در اختیار عامل مجوز کاربری قرار می‌گیرد، این عامل، ارزیابی اطلاعات وارد شده را (از نظر این‌که کلمه و رمز عبور تکراری نباشد) انجام می‌دهد، سپس در صورتی که این قاعده رعایت نشده باشد، پیغام خطا (با توجه به نوع مشکل) و برگشت اطلاعات صورت می‌گیرد و در صورت نبود مشکل، اطلاعات را به عامل ذخیره و بازیابی پاس می‌دهد و عامل ذخیره و بازیابی، اطلاعات را ذخیره می‌نماید. شکل (۸) مدل تعامل نقش مجوز کاربری در حیطه ذخیره را نشان می‌دهد.



شکل۸- مدل تعامل نقش مجوز کاربری در حیطه ذخیره

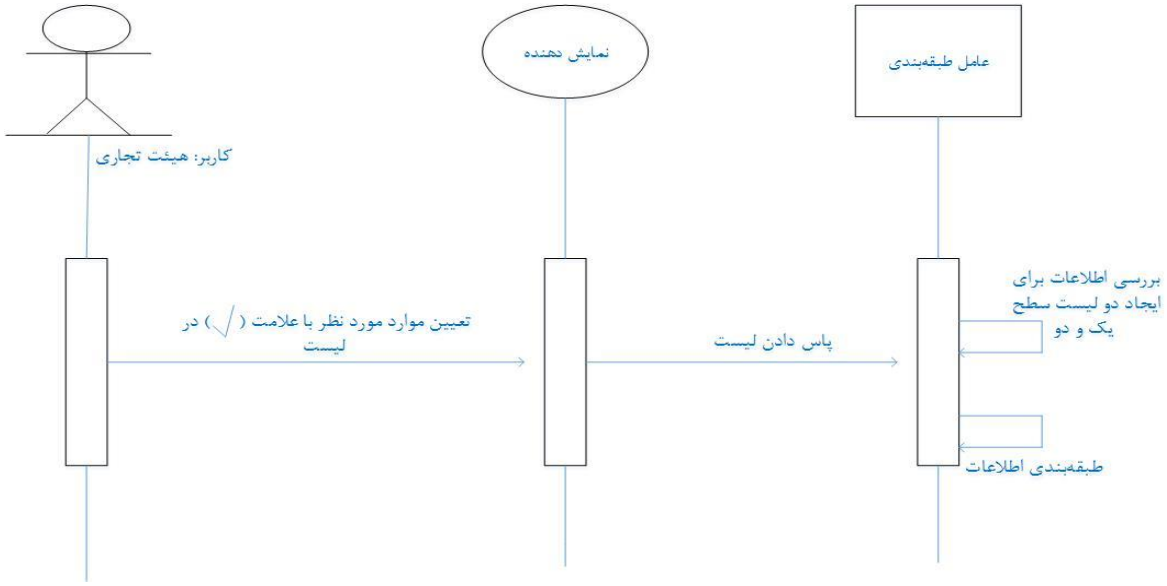
البته باید توجه داشت که کاربر (هئیت تجاری) فقط در زمانی که می‌خواهد درخواست سرمایه‌گذاری بدهد لازم است که در سیستم ثبت نام کند، بعد از این‌که درخواست انجام شد در صورتی که در درخواست و مدارک مشکلی نداشته باشند، کارشناس رمز و کلمه‌ی عبور جداگانه به کاربران اختصاص می‌دهد و دیگر نیازی به رمز و کلمه عبور قبلی نمی‌باشد و کاربر تنها با استفاده از رمز و کلمه عبور کارشناس می‌تواند از سیستم استفاده کند.

در حیطه تأیید اعتبار اطلاعات کاربری، کاربر به منظور ورود به سیستم، اطلاعات کاربری خود را وارد می‌کند و این اطلاعات در اختیار عامل جستجو قرار می‌گیرد و عامل جستجو بر اساس اطلاعات وارده، عمل جستجو در پایگاه‌داده سیستم را انجام می‌دهد. شکل (۹) مدل تعامل نقش مجوز کاربری در حیطه تأیید اعتبار را نشان می‌دهد.



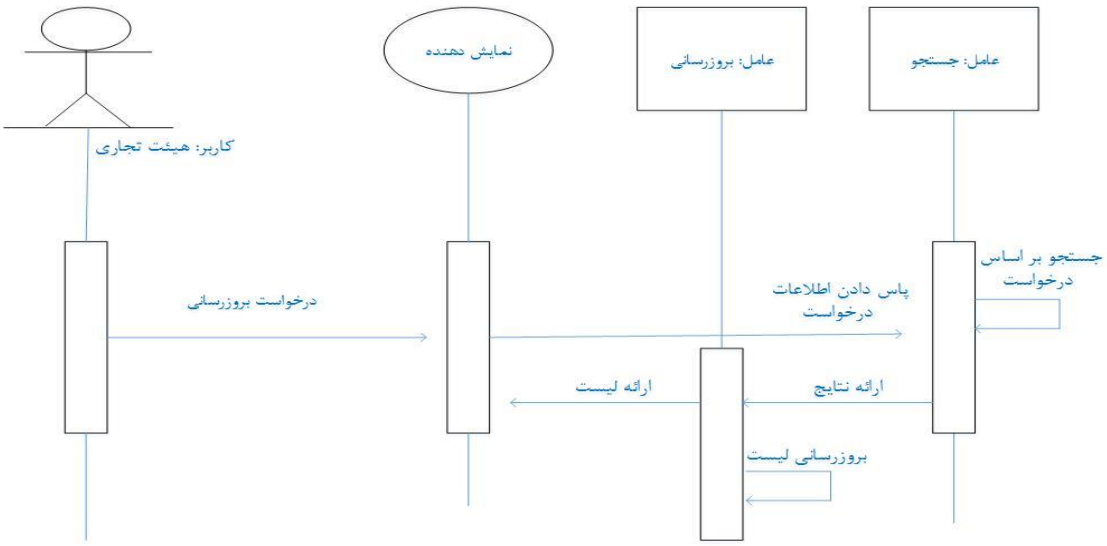
شکل۹- مدل تعامل نقش مجوز کاربری در حیطه تأیید اعتبار

نقش دیگری که در مدل نقش تعیین گردید، نقش طبقه‌بندی اطلاعات می‌باشد. در مدل تعامل این نقش، ابتدا کاربر در لیست ارائه شده از سرمایه‌گذاری‌ها، مواردی که مد نظرش است را مشخص می‌کند، پس از آن این اطلاعات در اختیار عامل طبقه‌بندی قرار می‌گیرد و این عامل پس از تفکیک اطلاعات به دو سطح یک (علاقه‌مندی‌های کاربر) و سطح دو (دیگر موارد)، اطلاعات را طبقه‌بندی می‌نماید. شکل (۱۰) مدل تعامل نقش طبقه‌بندی اطلاعات را ارائه می‌دهد.



شکل ۱۰- مدل تعامل نقش طبقه‌بندی اطلاعات

در مورد نقش به‌روزرسانی، کاربر ابتدا باید نوع به‌روزرسانی (به‌روزرسانی اطلاعات سرمایه‌گذاری، به‌روزرسانی علایق کاربر) را مشخص نماید (فرض بر این است که کاربر اجازه به‌روزرسانی موارد دیگر را ندارد). پس از مشخص شدن نوع به‌روزرسانی، درخواست به عامل جستجو داده می‌شود و این عامل بر اساس نوع به‌روزرسانی، اطلاعات را جستجو می‌کند و نتایج را در اختیار عامل به‌روزرسانی می‌گذارد و عامل به‌روزرسانی عملیات به‌روزرسانی لیست را انجام می‌دهد و نتایج را در اختیار کاربر قرار می‌دهد. شکل (۱۱) مدل تعامل نقش به‌روزرسانی را ارائه می‌دهد.



شکل ۱۱- مدل تعامل نقش به‌روزرسانی

- **مدل سرویس**

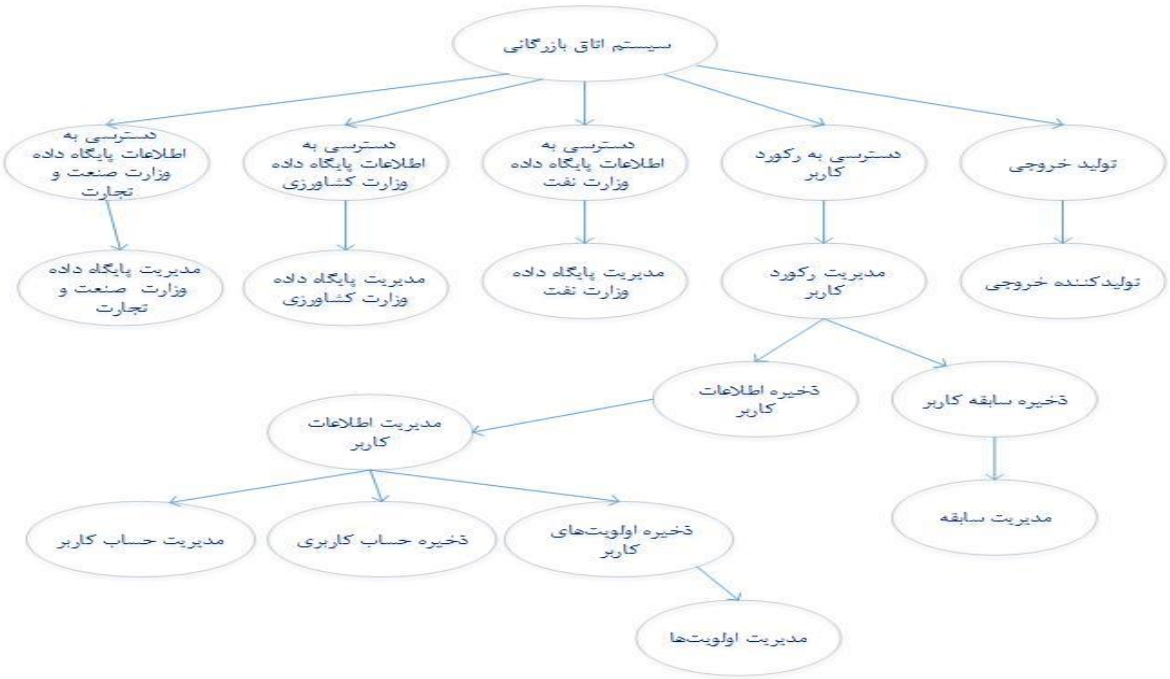
مدل سرویس با توجه به بخش قواعد و محدودیت‌های مدل دانش در فاز تحلیل تعیین می‌شود. دراین مدل هر نقش با حداقل یک سرویس مرتبط خواهد شد. برای هر سرویس باید ورودی‌ها و خروجی‌ها، پیش‌شرط‌ها و پس‌شرط‌ها مشخص شوند. ورودی‌ها و خروجی‌ها به راحتی از مدل تعامل استخراج می‌شوند. پیش‌شرط‌ها و پس‌شرط‌ها در این مدل نشان‌دهنده محدودیت‌های روی هر سرویس می‌باشد و از مسؤلیت‌های نقش‌ها استخراج می‌گردد. جدول ۲ مدل سرویس در سیستم اتاق بازرگانی را نشان می‌دهد.

جدول ۲- مدل سرویس سیستم اتاق بازرگانی

نام سرویس	ورودی	خروجی	پیش شرط‌ها	پس شرط‌ها
لیست کردن اطلاعات سرمایه گذاری	کد پروژه	نمایش کلیه‌اطلاعات سرمایه گذاری	وارد کردن مجوز کاربری	ثبت زمان رویداد انجام شده در اطلاعات سابقه
به‌روزرسانی اطلاعات	وارد کردن اطلاعات جدید	اصلاح اطلاعات مورد نظر	وارد کردن اطلاعات مورد نظر برای ویرایش	ثبت زمان رویداد انجام شده در اطلاعات سابقه
طبقه‌بندی اطلاعات	اطلاعات مربوط به علایق/ غیر علایق	نمایش ثبت اطلاعات در لیست یک / دو	مشخص کردن نوع اطلاعات	ثبت زمان رویداد انجام شده در اطلاعات سابقه

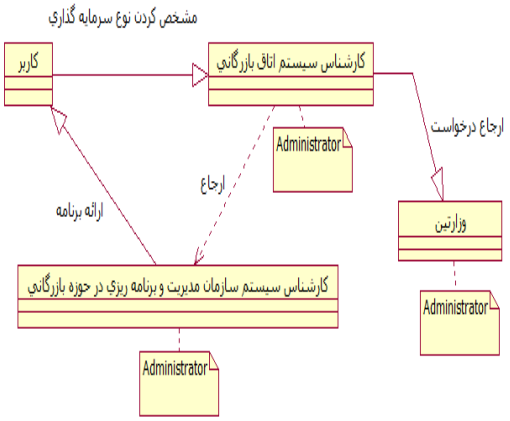
- **مدل سیستم گسترش‌یافته**

مدل سیستم گسترش‌یافته، یک دید کلی از فاز تحلیل سیستم می‌دهد و نقش‌ها، کنشگرها و ارتباطات بین آن‌ها را با توجه به خروجی‌های تولید شده در فاز تحلیل، مدل می‌کند. در این مدل زیرنقش‌های هر نقش، پایگاه‌داده‌های داده در سیستم و نحوه استفاده هر نقش از آن‌ها مشخص می‌شود. ارائه این مدل در متدولوژی AP می‌تواند اشتباهات تیم تحلیل را تا حد زیادی کاهش دهد، چرا که با دید جزئی‌تری ارتباطات بین بخش‌های مختلف در نظر گرفته می‌شود. شکل (۱۲) مدل سیستم گسترش‌یافته اتاق بازرگانی را نمایش می‌دهد.



شکل۱۲- مدل سیستم گسترش‌یافته سیستم اتاق بازرگانی

برای مدل نمودن کلاس‌های سیستم از مدل کلاس عامل- شی استفاده می‌گردد. بر اساس مدل کلاس عامل- شی در فاز تحلیل نقش‌های شناسایی شده بر اساس ویژگی‌های خود می‌توانند به عامل یا شی تبدیل شوند. برای تصمیم-گیری راجع به این انتخاب طراح سیستم می‌تواند از اطلاعات فراهم شده در فاز تحلیل استفاده نماید. شکل (۱۳) نمودار کلاس عامل- شی سیستم را نشان می‌دهد.



شکل۱۳- نمودار کلاس عامل- شی

متدولوژی پیشنهادی بر اساس نتایج تحلیل و طراحی سیستم اتاق بازرگانی، از دیدگاه افراد متخصص و خبره، نسبت به دو متدولوژی مورد بررسی، پشتیبانی بسیار مناسبی از پارامترهای خودمختاری، هدف‌گرا بودن، واکنش‌داربودن و قابلیت کاربرد دامنه دارد. همچنین از دیدگاه خبرگان، مدل‌های اضافه‌شده به متدولوژی ترکیبی پیشنهادی (مدل کنشگر گسترش‌یافته، مدل هدف جزئی، مدل سرویس و مدل سیستم گسترش‌یافته) باعث افزایش همگرایی بین فازهای تحلیل و طراحی شده است.

۷- نتیجه‌گیری و راهکارهای آینده

در این مقاله، یک متدولوژی ترکیبی توسعه‌یافته برای تحلیل و طراحی سیستم‌های مبتنی بر عامل معرفی شد. در این متدولوژی با افزودن مدل‌های جدید کنشگر گسترش‌یافته، هدف جزئی، سرویس و سیستم گسترش‌یافته، قابلیت مدل کردن ویژگی‌های خاص سیستم‌های مبتنی بر عامل در متدولوژی فراهم شده است، علاوه بر آن‌که متدولوژی از دیدگاه معیارهای خودمختاری، هدف‌گرا بودن، واکنش‌دار بودن قابلیت کاربرد دامنه از نگاه خبرگان بهبود یافته است. استفاده از راهکار ترکیبی در متدولوژی پیشنهادی باعث شد که دو هدف عمده، استفاده از استانداردهای کاری و تعریف مجدد بلوک‌های اصلی برآورده شود. این روش می‌تواند به‌عنوان یک راهکار مناسب برای توسعه نسل بعدی متدولوژی‌های عامل‌گرا مورد استفاده قرار بگیرد.

به منظور ادامه تحقیق در این زمینه لازم است متدولوژی AP در ساخت سیستم‌های گوناگون در دامنه‌های مختلف و با سطوح مختلف پیچیدگی مورد استفاده قرار گیرد و توانایی‌های متدولوژی مورد ارزیابی قرار گیرد. بدین ترتیب امکان ارزیابی مدل‌های ارائه شده و بهبود آن‌ها فراهم می‌گردد.

ویژگی‌های بارز معیارهای مهندسی نرم‌افزار، کمیت‌هایی است که برای این معیارها مطرح می‌شود و فرآیند ارزیابی سیستم‌های تولید شده را به یک فرآیند مهندسی تبدیل می‌نماید. بنابراین لازم است معیارهای مهندسی نرم‌افزار به-صورت کمّی بیان شوند و فاکتورهای کمّی نمودن آن‌ها در هر متدولوژی مشخص شود. به علت جدید بودن دیدگاه مهندسی نرم‌افزار مبتنی بر عامل، معیارهایی که برای ارزیابی متدولوژی‌های مبتنی بر عامل ارائه شده‌اند، بسیار محدود هستند. این معیارها به‌صورت کیفی مطرح شده‌اند و راهکارهای کمّی نمودن آن‌ها در ارزیابی‌های انجام شده بیان نشده است. به منظور ارتقا فرآیند مهندسی نرم‌افزار مبتنی بر عامل لازم است، روش‌های کمّی کردن معیارهای مهندسی نرم‌افزار مبتنی بر عامل مورد تحقیق قرار گیرد.

مراجع

[1] F. Zambonelli, "Key Abstractions for IoT-Oriented Software Engineering", IEEE Software, Vol.34, No. 1, 2017, pp. 38-45.

[2] L. Padgham, and J. Thangarajah, "Agent Oriented Software Engineering: Why and How", VNU Journal of Science: Natural Sciences and Technology, Vol. 27, No. 3, 2016, pp.190-204.

[3] X. Mao, Q. Wang, and S. Yang, "A survey of agent-oriented programming from software engineering perspective", Web Intelligence, Vol. 15, No. 2, 2017, pp. 143-163.

4] C. W. Shiang, J. J. Meyer, and K. Taveter, "Agent-Oriented Methodology for Designing Cognitive Agents for Serious Games", Engineering Multi-Agent Systems, 39, 2016.

[5] A. A. Lopez-Lorca, G. Beydoun, R.Valencia-Garcia, and R. Martinez-Bejar, "Supporting agent oriented requirement analysis with ontologies", International Journal of Human-Computer Studies, 87, 2016, pp. 20-37.

[6] G. Rodríguez, Á. Soria, and M. Campo, "Artificial intelligence in service-oriented software design", Engineering Applications of Artificial Intelligence, 53, 2016, pp. 86-104.

[7] A. D. Scott, "O-MaSE: An Extensible Methodology for Multi-agent Systems", Agent-Oriented Software Engineering: Reflections on Architectures, Methodologies, Languages, and Frameworks, 2014, pp. 173-191.

[8] A. D. Scott ,C. Juan, and G. Ojeda, "The O-MaSE Methodology", Handbook on Agent-Oriented Design Processes, Springer Berlin Heidelberg, 2014, pp.253-285.

[9] G. Picard, and M. P.Gleizes, "The ADELFE methodology", Methodologies and Software Engineering for Agent Systems, 2004, pp. 157-175.

[10] M. Cossentino, "From requirements to code with the PASSI methodology", Agent-oriented methodologies, 2005, pp. 79-106.

[11] Y. L. Hsu, C. H. Lee, and V. B. Kreng, "The application of Fuzzy Delphi Method and Fuzzy AHP in lubricant regenerative technology selection", Expert Systems with Applications, Vol 37, No.1, 2010, pp. 419-425.

[12] M. Wooldridge, "Reasoning about Rational Agents", The MIT Press: Cambridge, MA, 2000.

[13] F. Zambonelli, N. R. Jennings, A. Omicini, and M. J. Wooldridge, "Agent-oriented software engineering for internet applications", Coordination of Internet Agents Springer Berlin Heidelberg, 2001, pp. 326-346.

[14] T. Juan, A. Pearce, and L. Sterling, "ROADMAP: extending the Gaia methodology for complex open systems", Proceedings of the first international joint conference on Autonomous agents and multi agent systems: part, ACM, 2002, pp. 3-10.

[15] J. C.García-Ojeda, A. E. Arenas, and de J. Jesús Pérez-Alcázar, "Paving the way for implementing multi agent systems: integrating Gaia with agent-UML", Agent-Oriented Software Engineering V, Springer Berlin Heidelberg, 2005, pp. 179-189.

[16] J. Gonzalez-Palacios, and M. Luck, "Extending Gaia with agent design and iterative development. In Agent-Oriented Software Engineering", Springer Berlin Heidelberg, 2007, pp. 16-30.

[17] S. A. DeLoach, M. F. Wood, and C. H. Sparkman, "Multi agent systems engineering", International Journal of Software Engineering and Knowledge Engineering, Vol. 11, No. 3, 2001, pp. 231-258.

[18] J. DiLeo, T. Jacobs, and S. DeLoach, "Integrating ontology into multi agent systems engineering", Air univ Maxwell AFB al center for aerospace doctrine research and education, 2006.

[19] S. A. DeLoach, "Modeling organizational rules in the multi-agent systems engineering methodology", Advances in Artificial Intelligence, Springer Berlin Heidelberg, 2002, pp. 1-15.

[20] A. L. Self, and S. A. DeLoach, "Designing and specifying mobility within the multi agent systems engineering methodology", Proceedings of the 2003 ACM symposium on Applied computing, ACM, 2003, pp. 50-55.

[21] J. C. Garcia-Ojeda, S. A. DeLoach, W. H. Oyenan, and J. Valenzuela, "O-MaSE: a customizable approach to developing multi agent development processes", Springer Berlin Heidelberg, 2007, pp. 1-15.

[22] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An agent-oriented software development methodology", Autonomous Agents and Multi-Agent Systems, Vol. 8, No. 3, 2004, pp. 203-236.

[23] P. Giorgini, J. Mylopoulos, and R. Sebastiani, " Goal-oriented requirements analysis and reasoning in the tropos methodology", Engineering Applications of Artificial Intelligence, Vol. 18, No. 2, 2005, pp.159-171.

[24] I. J. Jureta, S. Faulkner, and P. Y. Schobbens," Allocating goals to agent roles during mas requirements engineering", Agent-Oriented Software Engineering, Springer Berlin Heidelberg, 2006, pp. 19-34.

[25] A. U. Mallya, and M. P. Singh, "Incorporating commitment protocols into Tropos" Agent-Oriented Software Engineering, Springer Berlin Heidelberg, 2005, pp. 69-80.

[26] J. Lind, "Iterative Software Engineering for Multiagent Systems: The MASSIVE Method", LNCS 1994, Springer-Verlag, 2001.

[27] G. Caire, W. Coulier, F. Garijo, J. Gomez, J. Pavón, F. Leal, and P. Massonet, "Agent oriented analysis using MESSAGE/UML", Agent-oriented software engineering II ,Springer Berlin Heidelberg, 2001, pp. 119-135.

[28] J. Pavón, and J. Gómez-Sanz, "Agent oriented software engineering with INGENIAS", Multi-Agent Systems and Applications III ,Springer Berlin Heidelberg, 2003, pp. 394-403.

[29] C. Iglesias, M. Garijo, J. Gonzales, and J. R. Velasco, "Analysis and Design of Multi-agent Systems using MAS Common KADS", Proceedings of the Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL'97), LNCS 1365, Springer-Verlag, 1998, pp. 313–326.

[30] M. Gervais, "ODAC: An Agent-Oriented Methodology based on ODP", Journal of Autonomous Agents and Multi-Agent Systems, Vol. 7, No.3, 2003, pp. 199–228.

[31] ISO/IEC X.900 1995, IS 10746-x ITU-T Rec. X90x, ODP Reference Model Part x.

[32] J. Odell, "Objects and agents compared", Journal of object technology, Vol. 1, No.1, 2002, pp. 41-53.

[33] J. Lind, "Issues in agent-oriented software engineering", Agent-Oriented Software Engineering Springer Berlin Heidelberg, 2001, pp. 45-58.

[34] L. Cernuzzi, G. Rossi, and L. Plata, "On the evaluation of agent oriented modeling methods", Proceedings of Agent Oriented Methodology Workshop, Seattle" ,Vol. 29, No. 2, 2002, pp. 1-12.

[35] P. Cuesta, A. Gómez, J. C. González, and F. J. Rodríguez, "A framework for evaluation of agent oriented methodologies", Proceedings of the Conference of the Spanish Association for Artificial Intelligence, Vol. 147, No. 4, 2003, pp. 151-152.

[36] E. Ghandehari, F. Saadatjoo, and M. A. Z. Chahooki, "AMA: a compound methodology for designing and implementing agent-based systems", Advances in Computer Science: an International Journal, Vol. 3, No. 5, 2014, pp. 107-114.

[37] C. E. Lin, K. M. Kavi, F. T. Sheldon, K. M. Daley, and R. K. Abercrombie,"A methodology to evaluate agent oriented software engineering techniques", System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on ,IEEE, 2007, pp. 60-60.

[38] B. Bauer, J. P. Müller, and J. Odell, " Agent UML: A formalism for specifying multiagent software systems", International journal of software engineering and knowledge engineering, Vol. 11, No. 03, 2001, pp. 207-230.

[39] E. Ghandehari, F. Saadatjoo, and M. A. Z. Chahooki, "Method integration: An approach to develop agent oriented methodologies", Journal of AI and Data Mining, Vol. 3, No.1, 2015, pp. 59-76.