دانشگاه آزاد اسلامی
واحد مبارکه

کنفرانس ملی

ECIT 2018

تحقیقات نوین در مهندسی برق، کامپیوتر و فناوری اطلاعات

بررسی و ارزیابی میزان کارایی در خصوص چهار متدولوژی رایج مهندسی نرم‌افزار عامل‌گرا

مریم قدیریان مارنانی
گروه برق و کامپیوتر، واحد مبارکه، دانشگاه آزاد اسلامی، مبارکه، اصفهان، ایران
mghadirian62@yahoo.com

چکیده

مهندسی نرم‌افزار عامل‌گرا[1]، یکی از جدیدترین ایده‌ها درحوزه مهندسی نرم‌افزار و نیز به عنوان رویکردی بعد از شی‌گرایی برای تولید نرم‌افزار بشمار می‌آید و امروزه در بسیاری از مسائل پیچیده به‌عنوان راه‌حل مناسب مطرح می‌باشد. در متدولوژی‌های مهندسی نرم‌افزار مبتنی بر عامل تمامی یا قسمتی از زیربخش‌هایی که در علم مهندسی نرم‌افزار مطرح هستند، پوشش داده می‌شوند. با وجود تکامل روزافزون متدولوژی‌های مهندسی نرم‌افزار مبتنی بر عامل هم‌چنان به علت نوپا بودن متدولوژی‌های مطرح شده، دربسیاری از آن‌ها مشکلات و نارسایی‌هایی دیده می‌شود، مشکلات پیش‌رو ما را نیازمند ارایه چهارچوب‌ها و تکنیک‌های ارزیابی در این زمینه می‌نماید، در این مقاله، چهارچوب‌های ارزیابی از پنج جنبه مفهوم, ابزار, عمل‌گرایی, چرخه حیات و جنبه های حمایتی متدولوژی‌ها را مورد سنجش قرار می‌دهد و هدف از آن مشخص نمودن نقاط ضعف و قوت متدولوژی‌های پیش‌رو به صورت کمی است. در این مقاله سعی شده است به بررسی و کارایی در خصوص چهار متدلوژی اصلی GAIA، PROMETHEUS، TROPOS و متدولوژی MASE پرداخته شود.

کلمات کلیدی

مهندسی نرم‌افزار عامل‌گرا، متدولوژی‌های عامل‌گرا، Gaia، Prometheus، Tropos، Mase

۱- مقدمه

دامنه‌ی وسیعی از مدل‌های مهندسی نرم‌افزار(مانند برنامه‌نویسی ساختاری، مدل‌های شی‌گرا و روش‌های مبتنی بر مؤلفه) با هدف تولید نرم‌افزار یا کاهش پیچیدگی های آن، مطرح می‌باشد. در حالی که هیچ‌کدام از مدل‌های مهندسی نرم‌افزار نمی‌توانند به طور کامل تکنولوژی نرم‌افزار را با نیازهای تجاری هماهنگ کنند، مهندسی نرم‌افزار مبتنی بر عامل، به‌عنوان رویکرد بعدی تولید نرم افزار(بعد از شی‌گرایی و یکی از جدیدترین و قدرتمندترین ایده ها درحوزه مهندسی نرم‌افزار بشمار می‌آید که قدرت و مزایای تکنیک‌های دیگر مهندسی نرم‌افزار مانند شی‌گرایی را افزایش می‌دهد و امروزه در بسیاری از مسائل پیچیده به‌عنوان راه‌حل مناسب مطرح می‌باشد. هدف اصلی آن ایجاد متدولوژی‌ها و ابزارهایی است، که توسعه و پشتیبانی از نرم‌افزارهای مبتنی برعاملها را معقول و به صرفه نماید زیرا مهندسی نرم‌افزار شی‌گرا نمی‌تواند خاصیتهای مشابه عامل‌ها از قبیل هوشمندی و انعطاف پذیری، ارتباطات و قدرتمند بودن را در محیطهای پیچیده، پویا حساس و توزیع شده و اینترنت داشته باشد[1].

با توجه به قابلیت‌های دیدگاه مبتنی بر عامل از جنبه مهندسی نرم افزار، از قبیل نزدیک بودن به دیدگاه انسان برای شکستن مساله، افزایش سطح تجرید و بسته‌بندی بلوک‌های پایه، افزایش امکان استفاده مجدد، افزودن یافته‌های هوش‌مصنوعی به نرم‌افزار و افزایش توانایی حل مسایل پیچیده، مهندسی نرم افزار مبتنی بر عامل رویکرد بعدی مهندسی نرم‌افزار پس از شی‌گرایی معرفی شده است[2]. بنابر تعریف IEEE "مهندسی نرم-افزار بکارگیری یک رویکرد سیستماتیک، با قاعده و کمی برای تولید، عملیاتی نمودن و نگهداری نرم افزار است. روشی که منجر به تولید نرم افزار به صورت مهندسی می‌گردد "[3].

متدولوژی‌های عامل‌گرا به عنوان راهکارها و روش‌هایی همراه با ابزار و زبان ، در توسعه سیستم‌های مبتنی بر عامل مورد توجه توسعه‌دهندگان و محققین هستند. بر اساس گفتار و متن های ارایه گردیده از طرف توسعه-دهندگان متدولوژی‌ها، هر متدولوژی خصوصیات و روش‌هایی را دارد، که کاربرد آن را در حوزه خاصی مشخص می نماید.از این رو استفاده از هر متدولوژی نیازمند درک از نکات قدرت و ضعف آن متدولوژی خواهد بود.

با وجود تکامل متدولوژی‌های مهندسی نرم‌افزار مبتنی بر عامل همچنان به علت نوپا بودن متدولوژی‌های مطرح شده ، دربسیاری از آن‌ها مشکلات و نارسایی‌هایی دیده می‌شود، این مشکلات ما را نیازمند ارایه چهارچوب‌ها و تکنیک‌های ارزیابی می‌کند، چهارچوب‌های ارزیابی از پنج جنبه مفهوم, فرایند, عملگرایی, چرخه حیات, جنبه های حمایتی به متدولوژی‌ها می‌نگرد.

هدف این مقاله، بررسی چهار متدولوژی اصلی این روش یعنی متدولوژی GAIA، PROMETHEUS، TPOPOS و متدولوژی MASEو روش‌های حاصل از توسعه متدولوژی‌های شی‌گرا در مهندسی نرم افزار می‌باشد.

بر این اساس ، قسمت‌های مختلف مقاله بشرح زیر بخش بندی می‌شود:بخش (۲)مهندسی نرم‌افزار عامل‌گرا، بخش(۳)مقایسه شی و سیستم، (۴)تعریف متدولوژی (۵)متدولوژی‌ها و فرآیندهای تولید سیستم‌های مبتنی بر عامل که خود زیر بخش‌هایی جهت معرفی چهار متدولوژی تقسیم بندی شده است، بخش(۶)چهارچوب ارزیابی که خود به زیر بخش‌هایی جهت بیان مقایسه بین متدولوژی‌های مورد مطالعه تقسیم‌بندی شده است، بخش(۷)-نتیجه‌گیری بدست آمده از این مقاله و نهایتا مراجع و منابع مورد استفاده در این مقاله بیان شده است.

۲- مهندسی نرم افزار عامل گرا

مهندسی نرم افزار عامل‌گرا از جدیدترین، روبه‌رشدترین و قدرتمندترین ایده‌ها در حوزه‌ی مهندسی نرم‌افزار بشمار می‌آید و هدف آن ساختن متدلوژی‌ها و ابزارهایی است تا توسعه و نگهداری نرم افزارهای عامل به صورت ارزان ممکن شود. علاوه بر این نرم افزار حاصل انعطاف‌پذیر بوده و کارکردن با آن ساده و آسان و در ضمن از کیفیت و کارایی بالایی نیز برخوردار باشد.

۳- مقایسه شی و عامل

شی[۲]نوعی نگرش به دنیاست. ویژگی اصلی که دارد کل جهان را به صورت مجموعه‌ای از اشیاء نگاه می‌کند که دارای ویژگی‌ها و رفتارهای خاص خود هستند و با یکدیگر در تعاملند و قابل اثرگذاری و اثرپذیری در مقابل یکدیگرند.

عامل[۳] چیزی است که می‌تواند برای یا به جای کسی عمل کند یک عامل به عنوان هر چیزی که می‌تواند بوسیله‌ی حسگرهایش محیط اطراف را درک کرده، تحت تاثیر و حتی عمل خاصی را انجام دهد تعریف می شود.

وجوه تمایز عامل و شی در زیر بیان می‌شود:

- خودمختاری عامل‌ها و اشیا
- رفتار انعطاف‌پذیری
- نحوه مدل‌سازی

٤- تعریف متدولوژی

متدولوژی[۴] مجموعه‌ای از متدهاست که باعث اتصال و پوشش‌دهی تمام مراحل مختلف یک فرآیند می‌شود و هدف از آن بیان روشی جهت حل مسئله در یک فرآیند و ایجاد ارتباط بین متدها جهت تولید خروجی است.

۵- متدولوژی و فرآیندهای تولید سیستم های مبتنی بر عامل

هدف از ارائه متدولوژی‌ها آن است که طراحان سیستم بتوانند بصورت سیستماتیک نیازهای کاربران سیستم را به مدل طراحی که دارای جزئیات کاملی از سیستم است و به راحتی قابل پیاده‌سازیست تبدیل کنند. در ادامه به معرفی و بررسی چهار متدولوژی مهندسی نرم‌افزار مبتنی بر عامل در این مقاله می‌پردازیم.

۵-۱- متدولوژی MASE

متدولوژی MASE درسال۱۰۰۰ توسط" دلوچ "معرفی شد. از متدولوژی‌های قدیمی شیءگرا بحساب می‌آید که از چرخه کامل توسعه پشتیبانی می‌کند. این متدولوژی عامل‌ها را به عنوان موجودیت‌های لزوما خودمختار و هدف‌گرا نمی‌نگرد، بلکه به عنوان پروسه‌های نرم‌افزاری ساده‌ای در نظر میگیرد که با یکدیگر برای رسیدن به کل اهداف سیستم تعامل میکنند.

هدف اصلی این متدولوژی فراهم آوردن چرخه‌ی حیات کامل(از مشخصات اولیه تا پیاده‌سازی سیستم مبتنی بر عامل)برای کمک به توسعه-دهندگان سیستم چندعامله می‌باشد.

از آنجایی که این متدولوژی تاریخچه قدیمی دارد از اولین متدولوژی‌هایست که جهت تحلیل و طراحی سیستم‌های مبتنی بر عامل مورد استفاده قرار می‌گرفت و توانایی حمایت از خصوصیاتی چون بررسی ثبات و قابلیت ردیابی را دارد. هدف متدولوژی MASE هدایت طراح از مشخصات سیستم اولیه به پیاده‌سازی سیستم‌های مبتنی بر عامل می‌باشد.

اما مهمترین نقطه قوت آن وجود ابزاری با نام "Agent Tool" است که توانایی پشتیبانی از کلیه مراحل تحلیل و طراحی سیستم در متدولوژی را داراست. این متدولوژی دارای هفت مرحله است که شامل دو فاز

طراحی و تحلیل می‌باشد[4] . جدول(۱) این مراحل را به اختصار بیان می-نماید.

جدول(۱):مراحل متدولوژی MASE

فاز	مراحل	توضیحات
تحلیل	گرفتن اهداف	یافتن اهداف از روی نیازمندی‌های ایجاد سلسله مراتب اهداف
	بکارگیری مراحل کاربرد	تعریف سناریو طراحی دیاگران توالی
	پالایش نقش‌ها	طراحی مدل وظایف جهت رسیدن به هدف
طراحی	ایجاد کلاس‌های عامل	مشخص شدن نقش‌ها بر اساس اهداف طراحی دیاگرام کلاس عامل
	ساختار تعامل	مشخص شدن جزئیات ارتباط و دیاگرام وظایف طی مکالمه
	مونتاژ عامل‌ها	ریز طراحی ساختار عامل‌ها با دیاگرام معماری عامل‌ها
	استقرار سیستم	طراحی دیاگرام نصب(مشخص کننده تعداد، نوع و عامل در سازمان)

این متدولوژی برای سیستم‌های بسته مناسب است. سیستم بسته سیستمی است که مشخصات ان مخفی باقی می‌مانند و با سیستم‌های دیگر نمی‌تواند ارتباط برقرار نماید.

۲-۵- متدولوژی TROPOS

متدولوژیTropos، یکی از متدولوژی‌های نسبتا جدید در مهندسی نرم افزار مبتنی بر عامل است. توسط" برشیانی "و همکاران در سال ۲۹۹۱ معرفی گردید این متدولوژی به وسیله افراد دانشگاهی از کانادا و ایتالیا نوشته شده است و تمامی فازها چرخه حیات نرم‌افزار را می‌پوشاند. تفاوت اصلی این متدولوژی متدولوژیهای دیگر در نظر گرفتن نیازمندیهای ابتدایی در حوزه تحلیل و تشخیص برای ذینفعان است[5] .

روش *i ، مبنای متدولوژی Tropos قرار گرفته است و مدل‌سازی و تحلیل سیستم در آن بر اساس مفاهیم و نمودارهای ارائه شده در *i انجام میشود. در این مدل از مفاهیمی مانند هدف،تمایلات ،ارتباط‌ها و توانایی های عامل استفاده می‌شود تا آنچه برای هر یک از ذینفعان سیستم حائز اهمیت است، مشخص شود. در چهارچوب *i دو نوع مدل وجود دارد:

- ارتباطات استراتژیک: نشان دهنده ارتباط و وابستگی بین بازیگران سیستم است.
- عقلانیت استراتژیک: نحوه ارتباط و وابستگی اهداف بازیگران مدل میشود .

با استفاده از این دو مدل در کنار هم ،یک دیدگاه جامع از سیستم ارائه میگردد[6]. Tropos سمبل‌های کنشگرها، اهداف و وابستگی میان آنها را پیشنهاد می‌کند که در آن چهار دیدگاه تحلیل نیازمندی‌ها، ،طراحی معماری، طراحی تفضیلی و پیاده‌سازی مطرح است. جدول(۲) این مراحل را به‌اختصار بیان می‌نماید.

جدول(۲):مراحل متدولوژی TROPOS

مراحل	توضیحات
تحلیل نیازمندی ها	تحلیل اولیه: درک اولیه از مسئله، سازمان مشخص‌کردن ذینفعان و علایقشان
	تحلیل ثانویه: مدل‌سازی سیستم هدف و محیط
طراحی معماری	تعریف زیر سیستم‌ها، داده و جریان‌های کنترلی
طراحی تفضیلی	تعریف مشخصات یک عامل و ریز خصوصیات
پیاده‌سازی	فاز پیاده سازی بر اساس ابزار Auml

این متدولوژی بر اساس سه ایده اصلی میباشد[7] :

- اهداف و برنامه‌ریزی‌ها در همه فازهای از تحلیل اولیه تا پیاده-سازی مورد استفاده است و از معماری عامل BDI1 استفاده می کند
- همه‌ی فازهای اولیه تحلیل نیازمندی‌ها را پوشش میدهد.
- به طور تکراری و بر اساس سطوح ادراکی به محصولات اجرایی،

روش انتقالی برای ساخت مدلهای سیستم را ، ارائه میدهد.

این متدولوژی مناسب هر گونه نرم‌افزاری نیست و برای نرم-افزارهای سیستمی که ذیفعان در آن زیاد واضح نیستند مناسب نیست اما برای محیط‌های باز و پویا می‌تواند تولید نرم‌افزای مناسب، مستحکم و قابل اطمینان را تضمین کند. ابزارهای جدید مدلسازی این متدولوژی Taomو Openomeمی‌باشد.

۳-۵- متدولوژی GAIA

این متدولوژی، در سال ۲۹۹۹ توسط" وولدریج "معرفی شد و یکی از نخستین متدولوژی‌های ارائه شده بحساب می‌آید. این متدولوژی عامل‌گرا تنها دو مرحله تحلیل و طراحی را پوشش می‌دهد و با گستره های جدید مطرح شده تا حدی طراحی جزییات را نیز در بر میگیرد[8] .

گایا به تحلیلگر این امکان را می‌دهد تا از کسب نیازمندی‌ها شروع به شناخت مسئله و سیستم نماید و بعد از تحلیل به طراحی سطح پایین برسد که مستقیما قابل پیاده‌سازیست . جدول(۳) این مراحل را به‌اختصار بیان می‌نماید.

جدول(۲):مراحل متدولوژی GAIA

مراحل	توضیح	خروجی
تحلیل	قدم اول: پیدا کردن نقش‌ها در سیستم قدم دوم: مشخص کردن الگوی تعاملات و تبادلات بین نقش‌ها مختلف	مدل محیطی،مدل اولیه نقشها، مدل اولیه تعاملات، مجموعه قواعد سازمانی
طراحی	مدل عامل: مشخص کردن انواع عامل‌ها و نمونه‌های آن‌ها در سیستم مدل خدمت: مشخص کردن فعالیت-های هر عامل و معاهدات هر نقش نهایتا ایجاد مدل از روی مدل‌های عامل‌ها و تعاملات بینشان	ساختار سازمانی کلی، مدل نقشها و تعاملات کامل شده، مدل عاملها و مدل سرویسها

لازم به توضیح است در مرحله تحلیل واکنش متقابل بین نقش‌ها وجود دارد که هر نقش ۴ صفت دارد :

- مسئولیت‌پذیری : نقش باید چیز خوبی را به سیستم اضافه کند و مسئولیت‌ها از اتفاق افتادن یک چیز بد جلوگیری می‌کنند
- مجوز ها : تعیین می‌کنند که چه نقشی اجازه چه کاری را دارد به طور دقیق‌تر به چه اطلاعاتی دسترسی دارد
- فعالیت ها: وظایفی هستند که یک نقش بدون داشتن کنش متقابل با سایر نقش‌ها انجام می‌دهد.
- پروتکل ها: پروتکل ها الگوی اصلی کنش ها هستند

متدولوژیGAIA برای طیف وسیعی از سیستم‌های چندعاملی بکار می‌رود و در سطح خرد و کلان عامل‌ها را بررسی می‌نماید. ابزار جدیدی که برای مدلسازی در آن مطرح شده‌است GAIA4E می‌باشد.

٤-٥- متدولوژی PROMETHUES

این متدولوژی در سال ۲۹۹۲ توسط" کرونکا" معرفی گردید و با استفاده از چهارچوب BDI[۵] ، بر روی ساخت عامل‌ها تمرکز دارد و مورد نیاز افراد کم تجربه می‌باشد.

مراحلی که این متدولوژی مورد پوشش قرار می‌دهد: مشخصات سیستم، طراحی معماری و طراحی جزییات هستند در جدول(۴) این مراحل به‌اختصار بیان می‌شود

جدول(٤):مراحل متدولوژی PROMETHUES

مراحل	توضیح
مشخصات سیستم	مدلسازی اهداف سیستم و سناریوها را پوشش می‌دهد
طراحی معماری	مدلسازی عامل‌ها، بررسی سیستم و پروتکل‌های میان عامل‌ها را با AUML نشان می‌دهد
طراحی جزئیات	تمرکز بر روی توسعه ساختار داخلی هر عامل در قالب قابلیت‌ها

کاربردهای این متدولوژی به بیان زیر می‌باشد:

- حمایت از کدنویسی با بازراهای موجود
- خطی بودن فرایند
- عدم مشکلات مدل آبشاری
- سناریوهای ان با UML مشخص می‌شود
- ابزارهای مدلسازی آن JDE و PDT هستند

٦- چهارچوب ارزیابی

چهارچوب ارزیابی مطرح شده در این مقاله شامل مفهوم, ابزار مدل‌سازی , عمل‌گرایی, مدل فرآیند و پوشش چرخه حیات, جنبه های حمایتی می‌باشد.
هر کدام از این جنبه‌ها دارای خصوصیاتیست بشرح زیر است:

- جنبه مفهوم: خودمختار، همزمانی، سازگاری، قابلیت، معماری، زبان ارتباط، وابستگی و کنش‌گرایی
- جنبه ابزار مدل‌سازی: مدل فرایند، نیازمندی، تحلیل نیازمندی، طراحی معماری، طراحی جزئیات، پیاده سازی و تست
- جنبه ابزار مدل‌سازی: نام ابزار، جمع‌آوری مدل، حمایت از هستان-شناسی، ذخیره زبان مدل‌سازی، ارزیابی، جمع‌آوری کد
- جنبه عمل‌گرایی: نداشت ابهام، دقت مدل، بیان‌کننده، سازگاری، سادگی نشانه‌ها، بازتعریف مستندات، مثال‌ها
- جنبه چرخه حیات:مدل فرآیند، نیازمندی‌ها،تحلیل نیازها،طراحی معماری، طراحی جزئیات، پیاده‌سازی، تست
- جنبه حمایتی: سیستم باز، امنیت، مقیاس‌پذیری، هستان شناسی، سازمان، پویایی سازمان، مدل دانش

در این مقاله هدف بررسی و ارزیابی متدولوژی‌های عامل‌گرا با رهیافتی وابسته به نظرات نویسنده‌گان مختلف است که به ارایه قالبی کلی از دیدگاه‌های متفاوت آنها پرداخته است. همچنین سعی شد از بازه های زمانی متفاوت متدولوژی‌هایی که مورد ارجاع بیشتری قرار گرفته باشند انتخاب شوند .در مقایسه متدولوژی‌ها از ارزیابی‌های پیشینی که در این زمینه انجام شده است نیز استفاده شده است[۹] .

لازم به توضیح است درقسمت های بعدی این بخش به مقایسه متدولوژی‌ها از دیدگاه‌های مختلف پرداخته شده‌است و از اعداد ۰ تا ۳ جهت بیان میزان حمایت از یک خصوصیت استفاده شده که عدد صفر ضعف این حمایت و عدد ۳ قوت آن را نشان می‌دهد.

۶-۱- مقایسه از لحاظ مفهوم

در مقایسه پوشش مفاهیم مرتبط با پیاده‌سازی سیستم‌های چند عامله می توان گفت که کم و بیش تمامی متدولوژی‌ها در این زمینه به تعریف راهکار پرداخته‌اند .در جنبه‌های ذهنی در همزمانی عاملها متدولوژیMASE نسبت به بقیه کاملتر است. در قابلیت ارتباط و زبان ارتباطی متدولوژی GAIA ارایه راهکار نپرداخته اند. در جدول(۵) مقایسه از لحاظ پوشش مفهوم بیان شده است.

جدول(۵) مقایسه از لحاظ پوشش مفهوم

متدولوژی	خودمختار	همزمانی	سازگاری	قابلیت	معماری	زبان ارتباط	وابستگی	کنش گرایی
MASE	۳	۳	۱	۲	۳	۱	۳	۳
GAIA	۳	۱	۲	۰	۲	۰	۲	۲
TROPOS	۳	۲	۲	۲	۳	۳	۳	۳
PROMETHEUS	۳	۳	۲	۱	۳	۲	۲	۳

۶-۲- مقایسه از بعد ابزار مدل‌سازی

متدولوژی GAIA خصوصیت متمایز نسبت به بقیه دارد که در نوع بازنمایی آن با استفاده از زبان صوری است، اما در خیلی از موارد هنوز راهکاری معرفی نکرده است .در زمینه ابزارهای پیاده سازی ابزار اریه گردیده به نام BOGORبرای متدولوژی MASE یک زبان برنامه نویسی را پشتیبانی نمی کند بلکه محیط را جهت اجرای همزمان فرایندها فراهم میکند .جدول(۶) مقایسه از نظر ابزار مدل‌سازی بیان شده است.

جدول(٦) مقایسه از لحاظ بعد ابزار مدل‌سازی

متدولوژی	نام ابزار	جمع‌آوری مدل	حمایت از هستان شناسی	ذخیره زبان مبنا سازی	ارزیابی	جمع‌آوری کد
MASE	Agent Tool	✗	✗	✓	✓	✓
GAIA						
TROPOS						
PROMETHEUS						
GAIA	GAIAE4	✗	✗	✗	✓	✓
TROPOS	UML	✓	✓	✓	✓	✗
PROMETHEUS	PDT	✗	✗	✓	✓	✓

لازم به توضیح است در فرآیند آبشاری، فرآیندها به گونه‌ای است که خروجی هر فاز ورودی برای فاز دیگر است. فرآیند تکاملی /افزایشی، پروژه در ابتدا از بخش‌های کوچک آغاز و به مرور زمان با رشد سیستم مشکلات مشخص و قبل از خرابی سیستم توسط فرضیه‌های نادرست مدل تکاملی / افزایشی کمک به برآورده شدن نیاز کاربران می‌کند.

۶-۳- مقایسه از دید عملگرایی

بنابر مقایسه انجام گرفته جنبه‌های عملگرایی متدولوژی‌هایی که سابقه بیشتری دارند وضعیت بهتری دارند .از لحاظ پوشش این جنبه ها متدولوژی GAIAخیلی ضعیف عمل کرده‌اند .در نهایت میتوان گفت متدولوژی‌های MASE، TROPOS ، در ارایه عملگرایی موارد بیشتری را پوشش داده‌اند . در جدول(۷) مقایسه از لحاظ دید عملگرایی بیان شده است.

جدول(۷) مقایسه از لحاظ دید عملگرایی

متدولوژی	نداشتن ابهام	دقت مدل	بیان کننده	سازگاری	سادگی- نشانه ها	بازتعریف	مستندات	مثالها
MASE	۲	۲	۲	۳	۳	۱	۳	۳
GAIA	۲	۲	۱	۱	۱	۰	۳	۰
TROPOS	۲	۲	۲	۳	۳	۳	۳	۲
PROMETHEUS	۲	۳	۳	۲	۰	۲	۲	۳

٤-٦- مقایسه از لحاظ مدل فرآیند و پوشش چرخه حیات

در مقایسه شرایط پوشش چرخه حیات میتوان گفت که متدولوژی TROPOSدربخش گرفتن نیازمندی‌ها نسبت به بقیه متدولوژی‌ها کاملتر عمل نموده‌اند.در تحلیل نیازمندی‌ها و طراحی معماری تقریبا تمام متدولوژیها روش‌ها و راهنمایی‌های مناسبی را عرضه کرده‌اند .در طراحی جزییات مربوط به ساختار عامل متدولوژیGAIA نسبت به بقیه ضعیف‌تر عمل کرده‌اند .در فازهای بعدی نسبت پوشش برای بعضی از متدولوژیهای کاهش مییابد یا به اتمام میرسد.

در فاز صحت سنجی متدولوژی PROMETHEUS به طور کامل از عملیات صحت سنجی در طراحی و نیازمندیها حمایت میکنند اما متدولوژی MASE بیشتر از صحت سنجی نیازمندیها حمایت میکند .در جدول(۸) مقایسه از لحاظ مدل فرآیند و پوشش چرخه حیات بیان شده است.

جدول(۸) مقایسه از لحاظ مدل فرآیند و پوشش چرخه حیات

متدولوژی	مدل فرایند	نیازمندی	تحلیل نیازمندی	طراحی معماری	طراحی جزئیات	پیاده سازی	تست
MASE	آبشاری	۱	۳	۳	۳	۲	۲
GAIA	آبشاری میان فازها و تکراری درون فازها	۰	۲	۳	۱	۰	۰
TROPOS	تکاملی/افزایشی	۳	۲	۲	۲	۲	۰
PROMETHEUS	آبشاری	۱	۳	۳	۳	۱	۳

مقایسه از لحاظ جنبه‌های حمایتی

ضعیف‌ترین حمایت‌ها از این خصوصیات از جانب متدولوژی PROMETHEUS است. در جدول(۹) مقایسه از لحاظ جنبه های حمایتی بیان شده است.

جدول(۹) مقایسه از لحاظ جنبه های حمایتی

متدولوژی	امنیت	مقیاس پذیری	هستان شناسی	سازمان	پویایی سازمان	مدل دانش
MASE	✔	✔	✔	✔	✖	✔
GAIA	✖	✔	✖	✖	✖	✔
TROPOS	✔	✔	✖	✖	✖	✖
PROMETHEUS	✖	✔	✖	✖	✖	✖

۷- نتیجه گیری

روش‌های عامل‌گرا یک شاخه ایجاد شده در مهندسی نرم‌افزار می‌باشد .جایی که تقاضاهای زیادی برای به کار بردن مدل عامل در کاربردهای صنعتی بزرگ و پیچیده در دامنه‌های مختلف کاربردی، وجود دارد .در دامنه‌های مختلف کاربردی و در کاربردهای صنعتی بزرگ و پیچیده قابلیت دسترسی متدولوژی‌های عامل‌گرا که از مهندسی نرم‌افزار، برای توسعه سیستمهای مبتنی بر عامل پشتیبانی می‌نمایندپر اهمیت است. متدولوژیهای مبتنی بر عامل به اندازه‌ای کامل و دقیق نیستند که تمامی جنبه‌ها وکاستی‌ها را پوشش دهنددر این مقاله با بررسی چهار متدولوژی مختلف و مقایسه آنها با یکدیگر نقاط قوت و ضعف، آنها مشخص شد ، تا بتوان متدولوژی مناسب را برای حوزه کاربرد مد نظر انتخاب نمود این بررسی در چهارچوبی مشخص که قابلیت ارزیابی عددی داشته باشد ارایه شد .البته در بعضی موارد منابع موجود امکان مقایسه عددی را به فراهم نکردند.

متدولوژیهای موجود در این مقاله برای روبرو شدن با پیچیدگی‌ها، راه‌کارهایی ارایه داده اند و انتخاب متدولوژی درست برعهده توسعه دهنده‌گان است که تا چه حد از مناسب بودن آن اطمینان دارند اما با این حال در انجام پروژه‌های مختلف انتخاب متدولوژی مناسب کاری مشکل است، که ارایه چهارچوبی برای انتخاب متدولوژی مبتنی بر مساله از کارهای آینده و در حال انجام است.

در ادامه در قالب یک جمع بندی کلی خصوصیات ، نقاط قوت و ضعف هر کدام از چهار متدولوژی بیان می‌گردد.جدول(۱۰) جمع‌بندی خصوصیات متدولوژی‌ها را نشان می‌دهد.

جدول(۱۰) جمع‌بندی خصوصیات متدولوژی‌ها

متدولوژی	خصوصیات
MASE	مبتنی بر شی‌گرایی،فرآیند آبشاری، حوزه کاری ایستا، عدم مدل خاص برای محیط، پشتیبانی از xml، مناسب سیستم‌های بسته، پشتیبانی از چرخه کامل توسعه، حمایت از بررسی ثبات و قابلیت ردیابی
GAIA	مبتنی بر شی‌گرایی،حوزه کاری ایستا، عدم مدل خاص برای محیط، پشتیبانی از xml، فرآیند آبشاری و تکاملی/آبشاری، عدم تعامل بین فازها، پوشش دهنده فازهای تحلیل و طراحی سیستم‌های چند عامله
TROPOS	محیط در قالب بازیگران، اهداف و وابستگی آنها مدل میشود، فرآیند تکاملی/آبشاری، پشتیبانی از زبان UML و AUML، مناسب سیستم‌های ناپایدار، حمایت از تمامی فازهای توسعه، نامناسب برای پروژه های سریع توسعه یافته
PROMETHEUS	فرآیند آبشاری و تکاملی/آبشاری، پشتیبانی از زبان UML بدون زبان استاندار آن، تمرکز روی ساخت عامل‌ها و توسط ابزارهای موجود کدنویسی را حمایت می‌کند، توجه به فعالیت‌های مهندسی نیازمندی‌ها

در این مقاله نقش متدولوژی‌های مبتنی بر عامل و خصوصیات ان‌ها بیان شد، هر چند که هیچ کدام از آن‌ها به اندازهای کامل و دقیق نیستند که تمامی جنبه‌ها ونیازها را پوشش دهند اما الگوی مناسبی برای مهندسین نرم‌افزار در انتخاب متدولوژی‌های عامل‌گرا باشد. [۱۱،۱۰]

مراجع

[1] Bernon, C., Cossentino, M., & Pavón, J. (2005). gent-oriented software engineering. *The Knowledge Engineering Review*, *20*(02), 99-116.

[2] F. Zambonelli, and A. Omicini, "Challenges and Research Directions in Agent-Oriented Software Engineering," *J. of Autonomous Agents and Multi-Agent Systems*, vol. 9, no. 3, pp. 253-283, 2004.

[3] R. Pressman, Software Engineering, a Practitioner's Approach, McGraw-Hill, Sixth Edition, 2005.

[4] S. A. DeLoach, M. Wood, and C. Sparkman, "Multiagent Systems Engineering," *The Int. Journal of Software Engineering and Knowledge Engineering*, vol. 11
no. 3, 2001.

[5] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos,and A. Perini, "TROPOS: An Agent-Oriented Software Development Methodology," *J. of Autonomous Agents and Multi-Agent Systems*, vol. 8, no.3, pp. 203-236, 2004.

[6] G. Grau, C. Cares, X. Franch, and F. J. Navarrete, "A
Comparative Analysis of i* Agent-Oriented Modelling Techniques," Eighteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'06), 2006.

[7] Giorgini, P., Mylopoulos, J., & Sebastiani, R. (2005). Goal-oriented requirements analysis and reasoning in the tropos methodology. *Engineering Applications of Artificial Intelligence*, *18*(2), 159-171.

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

[۹] علیرضایی، عبداله زاده، احسان، احمد،**"ارزیابی متدولوژی‌های عامل‌گرا"**،درس سمینار، کارشناسی ارشد، مهندسی نرم افزار، دانشگاه تهران

[۱۰] قندهاری، سعادت جو، زارع، عرفان، فاطمه، محمدعلی،**"تعیین میزان کارایی دو متدولوژی توسعه یافته پرکاربرد در حیطه مهندسی نرم‌افزار عامل‌گرا،"**همایش ملی پژوهش های مهندسی رایانه، ۲۰ آذر ۱۳۹۳، تهران

[۱۱] عبداله زاده، وفادار، احمد، شیوا، **موضوعات تحقیقاتی مطرح در مهندسی نرم‌افزار سیستم‌های مبتنی بر عامل،** نشریه علمی پژوهشی انجمن کتمپیوتر ایران، مجلد ۱۰، شماره ۱و۳(الف)، صفحات ۲۶-۳۹، ۱۳۹۱

زیرنویس‌ها

[1] Agent Oriented Software Engineering
[2]object
[3]agent
[4]Methodology
[5]Belief-Desire-Intention