Evaluate and comparission between Software Engineering Methodologies of Agent Systems

Asmaa Yaseen Hammo
asmahammo@uomosul.edu.iq
College of Comp. Sciences $ Math.

Maher Talal Alasaady
maher.alasaady@ntu.edu.iq
Northen Technical University

University of Mosul

Accepted 2011/11/02 **Received 2011/07/17**

ABSTRACT

Agent based systems software engineering is a new model in the field of software engineering. It provides new approaches for developing agent based systems. Numerous methodologies for agent based systems software engineering have been proposed. However, it is difficult to select a methodology for a specific project, and their application is still limited due to their lack of maturity and weakness for some of them. Evaluating and comparing their strengths and weaknesses is an important step towards developing better methodologies in the future, and help developers to select the most suitable methodology for project development. This research presents an evaluation and comparison of five agent based systems methodologies: Tropos, Gaia, Multi-agent Systems Engineering (MaSE), Multi Agent Systems Development (MASD) and Smart Multi Agent System (SmartMAS) methodology that is suggested by authors. The requirements of an agent based E-Commerce system were studied and applied on the development phases for these five methodologies, and toward a formal study, the evaluation and comparison were performed based upon a framework which addresses four major criteria: concepts and properties, notations and modeling, development process, and pragmatics. The results shows that the proposed methodology is give (85%) percentage of covering these criteria for the systems that were developed by using this methodology; this concluded that the proposed methodology is better than other methodologies.

Keywords: Agent system, SMART multiagent system, Tropos, Gaia, Multi-agent Systems Engineering (MaSE), Multi Agent Systems Development (MASD)

تقييم ومقارنة أساليب هندسة برمجيات نظم الوكلاء

أسماء ياسين حمو
كلية علوم الحاسوب والرياضيات
جامعة الموصل

ماهر طلال الأسعدي
هيئة التعليم التقني / الموصل

تاريخ استلام البحث: 2011/07/17 **تاريخ قبول البحث: 2011/11/02**

الملخص

تمثل هندسة برمجيات نظم الوكلاء أنموذجاً جديداً في حقل هندسة البرمجيات، وهي تقدم طرائق وأساليب جديدة لتطوير نظم الوكلاء. تم اقتراح العديد من أساليب هندسة برمجيات نظم الوكلاء، ولكن اختيار الأسلوب

المناسب لمشروع معين هي عملية صعبة، وأيضاً تطبيق هذه الأساليب لا يزال محدوداً بسبب الضعف وعدم النضوج للبعض منهم. عملية تقييم ومقارنة نقاط القوة والضعف لهذه الأساليب خطوة مهمة نحو تطوير أسلوب أفضل في المستقبل، وهذه العملية تساعد المطورين لاختيار الأسلوب المناسب لتطوير المشاريع. تم في هذا البحث تقييم ومقارنة خمسة أساليب لتطوير الوكلاء وهي Tropos و Gaia و Multi-agent Systems Engineering (MaSE) و Multi Agent Systems Development (MASD) وأسلوب Smart Multi Agent System (SmartMAS) المقترح من قبل الباحثين، إذ تمت دراسة متطلبات نظام التجارة الالكترونية المسند إلى الوكلاء وتطبيقها على مراحل التطوير للأساليب الخمسة السابقة، والاعتماد على إطار يحتوي على أربعة معايير رئيسية وهي، المفاهيم والخصائص، الترميزات والنمذجة، عملية التطوير، والناحية التطبيقية. أظهرت نتائج التقييم أن أسلوب SmartMAS يعطي نسبة (85%) من تغطية هذه المعايير للنظم التي يتم تطويرها باستخدام هذا الأسلوب، وهو الأفضل بالمقارنة مع الأساليب الأخرى.

الكلمات المفتاحية: نظام الوكلاء ، نظام SMART multiagent، Tropos، Gaia ، هندسة النظم متعددة الوكلاء (MaSE) ، تطوير أنظمة الوكلاء المتعددين (MASD).

1- المقدمة

تتميز تقنية الوكلاء بقدرتها على التعامل مع التعقيد والسلوك الموزع للنظم. لبناء مثل هذه النظم المعقدة، يحتاج المطور إلى أسلوب مناسب كأساس صلب لتطوير النظام بدءاً من مرحلة المتطلبات وصولاً إلى مرحلة البناء. تعرض أساليب هندسة برمجيات نظم الوكلاء معاني جديدة لتحليل وتصميم النظم البرمجية المعقدة، فهذه الأساليب لديها إمكانية محتملة لتحسين الممارسة الحالية لهندسة البرمجيات التقليدية ولتمديد مدى التطبيقات التي يمكن أن تعالج بشكل أفضل [11]. هناك العديد من الأساليب المقترحة لدعم مهندسي البرمجيات في تطوير نظم الوكلاء، مثل Gaia [20] وMaSE [19] وTropos [4] وMASD [2]، ولكن البعض منهم تم تصميمه بدون التقيد بجميع مراحل دورة حياة النظام، وأيضاً افتقاره في تمثيل أغلب خصائص الوكلاء، وأغلب هذه الأساليب لا تمثل مرحلة البناء أو تمثيلها بصورة غير واضحة وغير عملية.

تم في هذا البحث عرض بعض أساليب هندسة برمجيات نظم الوكلاء، وعرض بعض تقنيات وأطر تقييم ومقارنة هذه الأساليب، ثم إجراء عملية تقييم ومقارنة هذه الأساليب ومن ضمنها أسلوب SmartMAS المقترح من قبل الباحثين [1] لتشخيص نقاط القوة والضعف في هذه الأساليب. ومن خلال دراسة متطلبات نظام التجارة الالكترونية المسند إلى الوكلاء، وتطبيق هذه المتطلبات على مراحل التطوير لكل من الأساليب التي تم عرضها، تم وضع درجة تتراوح بين (1-7) للدلالة على تطابق الأسلوب مع معيار التقييم، إذ تمت عملية التقييم والمقارنة بالاعتماد على إطار يحتوي على أربعة معايير رئيسية تحتوي على معايير ثانوية وهي، المفاهيم والخصائص Concepts and Properties، الترميزات النمذجة Notations and Modeling، وعملية التطوير Development Process، والناحية التطبيقية Pragmatics.

2- أهداف البحث الهدف الأساسي للبحث هو تقييم ومقارنة أساليب هندسة برمجيات نظم الوكلاء، إذ تساعد نتائج التقييم والمقارنة المطورين لاختيار الأسلوب المناسب لمشروع معين، وأيضاً لتشخيص نقاط القوة والضعف في هذه الأساليب نحو تطوير أسلوب أفضل خالٍ من العيوب في المستقبل.

3- أساليب هندسة البرمجيات المعتمدة على الوكلاء Agent Based Software Engineering Methodologies (ABSE)

ABSE هي واحدة من أحدث نماذج البرمجة وهي انتقال كبير في هندسة البرمجيات باستخدام نهج جديد أكثر ملاءمة من تقنيات الكائنات الموجهة في بناء النظم البرمجية الموزعة والمعقدة [10]. ولكي يكون المفهوم قابلاً للتطبيق، هناك حاجة لإتباع نهجٍ موحدٍ في كل المراحل من خلال دورة حياة تطوير نظام الوكيل والذي يكون مماثلاً للعملية الموحدة Unified Process في تحليل وتصميم الكائنات الموجهة، وهذا النهج يسمى بالأسلوب Methodology. يمكن تعريف أسلوب هندسة البرمجيات هو مجموعة منظمة من المبادئ التوجيهية والمفاهيم أو الأنشطة التي تؤدي إلى مساعدة المطورين في القيام في تطوير البرمجيات [9]. دور أساليب تطوير نظم الوكلاء هي للمساعدة في جميع مراحل دورة حياة نظم الوكلاء وأيضاً إدارتها ويجب أن يوفر الأسلوب خارطة طريق لتوجيه المطورين في إنشاء نظم الوكلاء [18]. هناك عددٌ متزايدٌ من أساليب تطوير الوكلاء والتي تحاول أن تشمل جميع قضايا النظام ومن خلال جميع مراحل تطويره وتتنافس فيما بينها لتكون في المقدمة وتعطي النهج الرئيسي للمطورين.

تم في هذا البحث عرض الأساليب الأكثر شيوعاً لدراستها بشكل نظري من أجل تحديد نقاط القوة والضعف، فضلاً عن أوجه التشابه والاختلاف فيما بينها، ولذلك تم اختيار أربعة أساليب للتقييم والمقارنة بالإضافة إلى أسلوب مقترح من قبل الباحثين.

3-1 أسلوب Tropos

هو أسلوب لتطوير البرمجيات يتم فيه استخدام أنموذج الوكيل في عملية تطوير النظام. وهو يستخدم رموزاً لتمثيل الوكلاء والأهداف والمهام والتبعيات الاجتماعية لنمذجة وتحليل متطلبات النظام بنوعيها المبدئية والمتقدمة، والتصميم الهيكلي والتفصيلي [4].

يستند Tropos على فكرتين أساسيتين، الأولى، استخدام مفهوم الوكيل وجميع المفاهيم ذات الصلة (على سبيل المثال الأهداف والخطط)، التي تستخدم في جميع مراحل تطوير النظام، من التحليل المبكر وصولاً إلى التصميم التفصيلي، الثانية، تغطي المراحل المبكرة جداً من تحليل المتطلبات، وبالتالي تجعل الفهم أعمق للبيئة التي يجب أن يعمل فيها النظام بالإضافة إلى نوع التفاعلات التي يجب أن تحدث بين الوكلاء [11].

3-2 أسلوب Gaia

وهو أسلوب لتطوير الوكلاء يعتمد على نمذجة المفاهيم التنظيمية، يقترح أن المطورين يفكرون ببناء نظم الوكلاء على أساس عملية التصميم التنظيمي، أي أن التنظيم الحسابي للوكيل يمكن أن يصوّر على شكل مشابه لتنظيم الإنسان والذي يتكون من عدة أدوار مختلفة مع وظائف عديدة تتفاعل مع بعضها البعض. تنقسم المفاهيم الرئيسية للعناصر المستخدمة في Gaia إلى فئتين: الأولى مجردة، والثانية ملموسة. الكيانات المجردة هي تلك التي تظهر في عملية التطوير ولكن ليس لها تحقيق مباشر في النظام. الأجزاء الملموسة هي تلك التي تظهر في العملية ولديها نسخ مباشرة مثيلة لها في النظام في وقت التشغيل. استناداً إلى التصنيف المنطقي المذكور أعلاه، تتكون العناصر في النماذج المصممة لمرحلتين رئيسيتين في التطوير: الأولى هي مرحلة التحليل، والتي تستوعب معظم

المفاهيم المجردة، وتتألف من نمذجة الأدوار وأنموذج التفاعل، الثانية هي مرحلة التصميم، التي تولد العناصر الملموسة وهي تتألف من نموذج الوكيل وأنموذج الخدمة ونموذج التعارف [20].

3-3 أسلوب MaSE

يوفر هذا الأسلوب المساعدة في تطوير نظم الوكلاء من خلال مرحلتين رئيسيتين وهما: التحليل والتصميم. يوجد في كل مرحلة سلسلة من الخطوات لنمذجة النظام إذ يتم إنشاء النماذج ذات الصلة. النماذج التي تكونت في الخطوات السابقة يكون ناتج إخراجها مرجعاً لمُدخلات النماذج في الخطوة اللاحقة. التبادلات بين الخطوات يمكن تتبعها إلى الأمام وإلى الخلف. تتكون مرحلة التحليل من ثلاث خطوات وهي، تعريف الأهداف، تطبيق حالات الاستخدام، تنقية الأدوار، وهناك أربع خطوات في مرحلة التصميم وهي، إنشاء فئات الوكيل، بناء المحادثات، تجميع فئات الوكيل، وتصميم النظام [19].

4-3 أسلوب MASD

هو أسلوب لتطوير نظم الوكلاء يتعامل مع مفهوم الوكيل بصورة مجردة وعالية المستوى، يستخدم الأسلوب بعض التقنيات مثل خرائط حالة الاستخدام Use Case Maps (UCMs) [Buhr, 1998]، ومخططات حالة استخدام Use Case Diagrams (UCDs)، وأيضاً يدعم هذا الأسلوب البساطة وسهولة الاستخدام وكذلك التتبع. يتألف أسلوب MASD من أربع مراحل رئيسية وهي، مرحلة متطلبات النظام، مرحلة التحليل، مرحلة التصميم، ومرحلة البناء. يتم التعامل في هذا الأسلوب مع جزئين وهما هيكلية الوكيل بصورة مفردة وهيكلية النظام بصورة عامة. تكون نتائج التحليل والتصميم عبارة عن نماذج عديدة للحقائق والأهداف والخطط [2].

5-3 أسلوب SmartMAS

وهو أسلوب مقترح من قبل الباحثين، تم تطويره بالاعتماد على أساليب تطوير الوكلاء الأخرى وبعض التقنيات المستحدثة، إذ تم دمج التقنيات ذات المزايا القوية وإهمال التقنيات الضعيفة. يحتوي هذا الأسلوب على أربع مراحل رئيسية وهي: المتطلبات، التحليل، التصميم، البناء.

في مرحلة المتطلبات يوصف النظام بمستوى عالٍ لتكوين مخطط الفاعل، ويستخدم هذا المخطط لوصف النظام والمكونات والمهام التي يجب أن يؤديها كل مكون داخل النظام، ومفهوم النظام من وجهة نظر المستخدم. تقسم مرحلة المتطلبات إلى خطوتين رئيسيتين وهي المتطلبات المبدئية والمتقدمة. في الخطوة الأولى يمثل النظام بشكل مخطط بسيط، وفي الخطوة الثانية يتم تفصيل المخطط من خلال: إدخال فاعل النظام، وتجزئة الأهداف، وتكوين المخطط النهائي، وتحليل التبعيات.

الهدف من مرحلة التحليل هو تحويل متطلبات النظام بشكل واضح لتمثيله بشكل مبدئي وتهيئته لمرحلة التصميم، إذ تقوم هذه المرحلة بتحليل متطلبات النظام باستخدام مخطط الفاعل. تقسم مرحلة التحليل إلى خطوتين رئيسيتين: الأولى هي تكوين وصف لهيكلية الوكيل بشكل منفرد، والثانية هي وصف للنظام بشكل كلي.

يتم في مرحلة التصميم تمثيل النماذج التي تم بناؤها في مرحلة التحليل بصورة مفصلة. إذ تراجع هذه النماذج مع إضافة بعض التفاصيل وفقاً لمواصفات مرحلة البناء. الهدف الرئيسي من مرحلة التصميم هو الحصول على تصميم الوكيل الهيكلي وتصميم النظام والتي يمكن استخدامها في مرحلة البناء، بالإضافة إلى الحصول على

أنماط يمكن إعادة استخدامها في التطبيقات الأخرى المشابهة أو من خلال مراجعتها لغرض التحديث والصيانة والتعديل من قبل المطور، والكثير من فوائد التصميم الأخرى. تتمحور هذه المرحلة في ثلاث خطوات وهي: بناء مخطط التصميم، تعريف حاوية الوكيل، تكوين أنموذج الاتصالات.

وأخيراً في مرحلة البناء تجري عملية التطوير الفعلية للنظام من خلال البدء بكتابة الرموز البرمجية، وبناء النظام على أساس مواصفات مرحلة التصميم. هناك عدة أطر ومنصات لتطوير نظم الوكلاء، والأسلوب المقترح يدعم البعض منها مثل JACK و Jadex، وبعض المنصات الأخرى التي تستخدم هيكلية BDI ومقاييس FIPA [1].

4- تقنيات التقييم Evaluation Techniques

وهي التقنيات التي تستخدم لغرض تقييم ومقارنة أساليب تطوير البرمجيات. هنالك العديد من التقنيات المقترحة في هذا المجال والتي يمكن من خلالها التوصل إلى النتائج المناسبة من خلال دمجها مع أطر التقييم، بعض هذه التقنيات تستخدم لتقييم أساليب ABSE مثل:

1. المقارنة المستندة على الميزات Feature Based Comparison

وهي المقارنة المستندة على مجموعة من الميزات التي يتم فحصها في الأساليب، ميزة هذه التقنية هي تطبيقها بنجاح في كثير من الحالات التي يمكن القيام بها بشكل مستقل عن المصادر الخارجية والتحيز وذلك بسبب وضوح هذه الميزات [15].

2. النمذجة الوصفية Meta-Modeling

يستخدم نهج النمذجة الوصفية لتقييم تحليل وتصميم الأساليب باستخدام نهج رسمي لمقارنة الأساليب. تجري عملية التقييم بالخطوات التالية (1) بناء أنموذج وصفي عام للأسلوب الأساسي (المطلوب) (2) بناء أنموذج وصفي للأسلوب الذي تم تقييمه مسبقاً (3) مقارنة ما سبق ذكره من النماذج [5].

3. المقاييس Metrics

تحلل تقنية المقاييس تعقيد الأسلوب، وهذا يتم من خلال تحليل عدد عناصر البناء، إذ أن عناصر البناء هي البنية التحتية للأسلوب، على سبيل المثال في UML تكون الفئة والكائن والروابط هي أمثلة لعناصر البناء [14]. يقوم هذا الأسلوب باتخاذ القرار على أساس عناصر البناء التي سوف تُحسب وأيضاً درجة تجزئتها، بالإضافة إلى أن هناك حاجة إلى الكثير من العمل التجريبي للتحقق من صحة القياسات.

4. تقنيات التقييم التجريبية Empirical Evaluation Techniques

وهي تشمل الدراسات الاستقصائية والبحثية والمختبرية والتجارب الميدانية، فضلاً عن دراسات الحالة Case Studies. عادة ما تتطلب هذه التقنيات مشاركة العديد من الممارسين والأشخاص، وهو أمر صعب ويتطلب وجود العديد من الموارد.

5- أطر التقييم Evaluation Frameworks

هنالك العديد من أطر تقييم ومقارنة أساليب ABSE، في كل إطار يتم استخدام تقنية مختلفة، فعلى سبيل المثال، استخدم Sabas إطاراً للتقييم يحتوي على إطار متعدد الأبعاد في كل بعد يوجد معيار للتقييم، على سبيل المثال: طرائق التمثيل، خاصية التنظيم، دعم الوكالة، خاصية التعاون، والتقنية المستخدمة، إذ يستخدم المعيار مؤشراً لتقييم ومقارنة هذه الأساليب. نتائج المقارنة تكون على شكل رموز توضع في مصفوفة ذات بعدين، تحتوي في صفوفها على المعيار وفي أعمدتها على أسم الأسلوب، والرموز المستخدمة تكون على الشكل الأتي، الرمز (Y) للدلالة على دعم الأسلوب لهذا المعيار، والرمز (N) للدلالة على عدم دعمه، و الرمز (P) للدلالة على احتمالية دعم الأسلوب لهذا المعيار، والرمز (_) للدلالة على عدم القدرة على التقييم لعدم توفر المعلومات الكافية [12].

واستخدم Sturm أربعة أبعاد لتجزئة الاختبار إلى أربعة أجزاء. هذه الأبعاد تحتوي على المفاهيم والخصائص، الترميزات النمذجة، وعملية التطوير، والناحية التطبيقية [16].

وقام Dam بوضع إطار يحتوي على الأجزاء كما سبق ذكرها، مع تحليل المعيار وقياسه لمختلف الأساليب وإمكانية وضع درجة تتراوح بين (1-7) كقيمة تحدد تطابق هذا المعيار مع الأسلوب المقابل، وفي هذه الحالة ممكن للقارئ أن يلاحظ إمكانيات هذه الأساليب ودرجاتهم التي تحدد قابلية أو ضعف أو قوة تطبيق هذا الأسلوب للمعيار، إذ أن الدرجات الأعلى تدل على دعم الأسلوب للمعيار بصورة أكبر [7].

فيما وضع Shehory إطاراً يستخدم خصائص هندسة البرمجيات التقليدية وخصائص ABSE معياراً لتقييم الأساليب، في كل حقل يوجد معيار مرافق للتقييم، ويستخدم تعليق لفظي لكل معيار مع استخدام الرموز التالية، الرمز (+) للدلالة على جودة دعم المعيار، والرمز (*) يدل على دعمه بصورة مقبولة، والرمز (NS) يدل على عدم دعم الأسلوب للمعيار [13].

6- تقييم ومقارنة أساليب هندسة برمجيات نظم الوكلاء

سوف نشير في هذا البحث لأسلوب ABSE على أنه مجموعة كاملة من المبادئ التوجيهية والأنشطة وهي: دورة الحياة الكاملة[1]، مجموعة شاملة من المفاهيم والنماذج، مجموعة كاملة من التقنيات (القواعد والمبادئ التوجيهية)، مجموعة من الترميزات، وطرائق إعادة الاستخدام. إذ يرتبط أحد هذه المبادئ التوجيهية أو الأنشطة مع أحد المعايير الرئيسية الأربعة وهي المفاهيم والخصائص، الترميزات والنمذجة، وعملية التطوير، والناحية التطبيقية.

أجريت عملية التقييم والمقارنة بين الأساليب باستخدام التقنية المعتمدة على الميزات بسبب دعم هذه التقنية للمفاهيم الرئيسية لعملية التطوير والتي تكون البنية الأساسية في كل من البرمجيات التقليدية ونظم الوكلاء. تمت عملية التقييم والمقارنة من خلال ثلاث مراحل، في المرحلة الأولى تم عرض هذه المفاهيم في جدول يوضح المعايير اللازمة لتطبيق هذا المفهوم، وفي المرحلة الثانية تم سرد شرحٍ مختصرٍ لدعم كل أسلوب للمعيار الموجود في الجدول، وفي المرحلة الأخيرة تم عرض نتائج التقييم ولكل أسلوب في جدول ثنائي الأبعاد، تحتوي صفوفه على المعايير التي ذكرت في الجدول الأول وتحتوي الأعمدة على أسماء الأساليب المطلوب تقييمها، وفي خلايا الجدول

[1] سوف نطلق على دورة الحياة الكاملة للأسلوب الذي يغطي مراحل التطوير الرئيسية وهي المتطلبات والتحليل والتصميم والبناء، دون الأخذ بنظر الاعتبار مراحل الفحص والصيانة والمراحل الأخرى.

وضعت درجة تتراوح بين (1-7) كقيمة تحدد تطابق هذا المعيار مع الأسلوب المقابل، إذ أن الدرجات الأعلى تدل على دعم الأسلوب للمعيار بصورة أكبر، مع وضع مخطط يصور بشكل واضح النسبة المئوية لمعدل الدرجات الخاصة بكل أسلوب. وضعت الدرجات على أساس دراسة متطلبات نظام التجارة الالكترونية المسند على الوكلاء وتجربته على مراحل التطوير لكل من الأساليب أعلاه، بالإضافة إلى عرض ومناقشة النتائج.

الأساليب التي تمت مقارنتها هي Gaia وMaSE وTropos وMASD، بالإضافة إلى أسلوب SmartMAS.

1-6 المفاهيم والخصائص Concepts and Properties

المفهوم هو فكرة مجردة أو فكرة مشتقة أو مستمدة من حالات محددة ضمن نطاق المشكلة، والخاصية هي قدرة خاصة أو ميزة. تتناول هذه الفقرة مسألة ما إذا كان الأسلوب يلتزم بالمفاهيم الأساسية (مفاهيم وخصائص) للوكلاء والنظم المتعددة الوكلاء. ومن أجل تنفيذ مثل هذا التقييم نحن بحاجة إلى تحديد هذه المفاهيم. يوضح الجدول (1) معايير المفاهيم والخصائص التي يجب أن يتم تقييم أسلوب تطوير الوكلاء على أساسها [17].

الجدول (1). معايير المفاهيم والخصائص

المعيار	الوصف
تجريد الوكيل Agent Abstraction	مقدرة الأسلوب لوصف وتعريف الوكلاء بدرجة عالية من التجريد.
الخواص العقلانية Mental Properties	امتلاك الوكلاء على الخواص العقلانية ولديهم آلية خاصة لتحقيق الرغبات من خلال الأهداف.
الاستقلالية Autonomy	قدرة الوكيل للعمل من دون تدخل خارجي.
التكيف Adaptation	مرونة الوكيل للتعامل مع التغييرات الحيوية التي تحصل على البيئة.
التواصل Communication	تعريف بروتوكول أو آلية معينة للتفاعل بين الوكلاء.
التعاون Collaboration	قدرة الوكيل للتعامل مع الوكلاء الآخرين لتحقيق الأهداف.
التزامن Concurrency	مقدرة الوكيل على تنفيذ المهام بصورة متزامنة.

1. أسلوب Tropos

يعرّف الوكلاء من خلال الفاعلين في المخطط التنظيمي. يوجه الوكيل حالاته العقلية ومقدراته من التحليل إلى التصميم التفصيلي باستخدام هيكلية BDI من خلال الأهداف القوية والأهداف الضعيفة والمهام. تكون خواص الاستقلالية والتكيّف مضمّنة في فئة الوكيل. صفة التواصل والتعاون متوفرة في العلاقات التبعية. خاصية التزامن متوفرة في التصميم الذي يتم فيه جمع كل المقدرات إلى أنواع الوكيل، وفئات الوكيل يمكن أن توفر العديد من الخدمات بشكل آني.

2. أسلوب Gaia

يمثّل الوكيل على شكل دور واحد أو أكثر ويتم نمذجة الدور من خلال استخدام خصائص الحياة Liveness أو خصائص السلامة Safety. تعرّف خاصية الاستقلالية ضمنياً من خلال هذه الخصائص. لا تتوفر الخواص العقلانية في هذا الأسلوب ولكن يتم استخدام خصائص الحياة والسلامة لتوضيح سلوك الوكيل. يمكن تعريف خاصية التكيف من خلال المعادلات الوظيفية الاختيارية لخصائص السلامة. يوجد تعريف لأنموذج

تفاعل الخدمات بين الوكلاء، ولكن لا يوجد تعريف للاتصال بين الوكلاء. خاصية التعاون ممكن أن تنجز من خلال الخدمات بين الأدوار. أي دور يمكن أن يختار عمله بصورة آلية طبقا للحالة في ذلك الوقت.

3. أسلوب MaSE

يتم بناء فئات الوكيل من خلال أنموذج الدور. يكون هذا الدور مضمّناً في أهدافه ومهامه المرتبطة. هذه المهام تكون مستقلة في إنجاز الأهداف من خلال التواصل مع الوكلاء الآخرين. لا يستطيع الوكيل الاستجابة على بيئته إذا تغيرت أهدافها ولهذا السبب خاصية التكيف لا يمكن تحقيقها. يمكن إنجاز خواص التزامن والتعاون من خلال أنموذج المهام ومن خلال التواصل بين الوكلاء.

4. أسلوب MASD

يتم تعريف الوكيل من خلال الفاعلين في مخططات UCDs والمكونات في مخططات UCMs، وينقل هذا التعريف إلى الدور الذي يكون مضمّناً في أهدافه ومهامه المرتبطة. تعرّف الحقائق والأهداف والخطط في نماذج لتشكيل هيكلية BDI. تكون المهام مستقلة في إنجاز الأهداف من خلال الدور. ممكن للوكيل الاستجابة إلى بيئته إذا تغيرت أهدافها وذلك من خلال قيم الحقائق الحيوية.

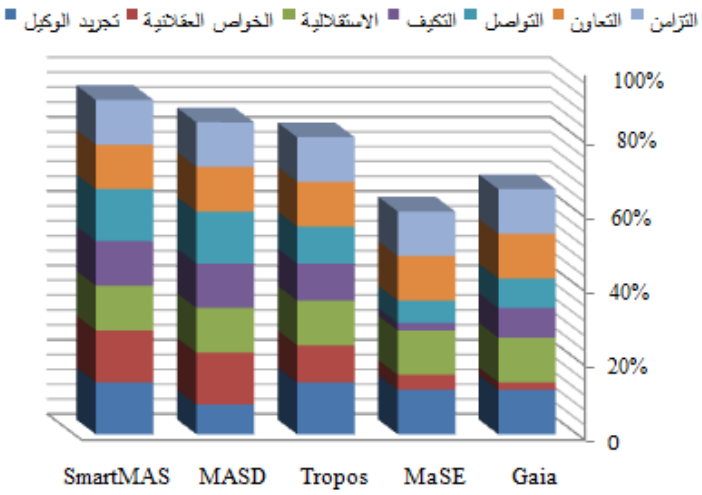
5. أسلوب SmartMAS

يتم تعريف الوكيل من خلال الفاعلين في مخطط الفاعل ومن خلال تعيين الأدوار المناسبة لهم. تعرّف الحقائق والأهداف والخطط في نماذج لتشكيل هيكلية BDI. يتم تعريف الأهداف، والمهام التي تكون مستقلة في إنجاز الأهداف من خلال الوكيل. ممكن للوكيل الاستجابة لبيئته إذا تغيرت أهدافها وذلك من خلال قيم الحقائق الحيوية. وجود نموذج للتواصل بين الوكلاء وتعريف التفاعلات. يمكن إنجاز خواص التزامن والتعاون من خلال نموذج الخطط ومن خلال التواصل بين الوكلاء ومن خلال التبعية.

يوضح الجدول (2) نتائج تقييم ومقارنة الأساليب لمعايير المفاهيم والخصائص، ويصور الشكل (1) مخطط النسبة المئوية لمعدل نتائج تقييم الأساليب للمفاهيم والخصائص.

الجدول (2). نتائج تقييم ومقارنة الأساليب لمعايير المفاهيم والخصائص

	Gaia	MaSE	Tropos	MASD	SmartMAS
تجريد الوكيل	6	6	7	4	7
الخواص العقلانية	1	2	5	7	7
الاستقلالية	6	6	6	6	6
التكيف	4	1	5	6	6
التواصل	4	3	5	7	7
التعاون	6	6	6	6	6
التزامن	6	6	6	6	6
نتائج تقييم الأساليب للمفاهيم والخصائص	**65%**	**61%**	**81%**	**85%**	**91%**



الشكل (1). معدل نتائج تقييم الأساليب لمعايير المفاهيم والخصائص

2-5 الترميزات والنمذجة Notations and Modeling

الترميزات هي الرموز التقنية المستخدمة لتمثيل العناصر داخل النظام. النمذجة هي مجموعة من النماذج التي تصور النظام على مستويات مختلفة من التجريد وجوانب مختلفة للنظام. خلال عملية التطوير تكون الرموز والنماذج المتناسقة والواضحة دليلاً لمعرفة سلوك الوكيل وأيضاً لتسهيل التواصل بين المطورين، بالإضافة إلى تسهيل التعقيدات من خلال فهمها وبنائها. يوضح الجدول (3) معايير الترميز والنمذجة التي يجب أن يلتزم بها الأسلوب [3].

الجدول (3). معايير الترميز والنمذجة

المعيار	الوصف
سهولة الفهم Accessibility	السهولة أو البساطة في فهم واستخدام الأسلوب يعزز قدرات كل من الخبراء والمبتدئين على استخدام المفاهيم الجديدة.
قابلية التعبير Expressiveness	المقدرة على تقديم مفاهيم النظام الذي يشير إلى: الهيكلية، المعرفة المغلفة، تدفق البيانات والتحكم، الأنشطة المتداخلة، القيود المفروضة على الموارد مثل (الوقت، المعالج، الذاكرة)، معمارية النظام، التنقل في الوكلاء، تفاعل النظام مع النظم الخارجية، وتعريفات واجهة المستخدم.
إدارة التعقيد Complexity Management	القدرة على التعامل مع مستويات مختلفة من التجريد (أي مستويات مختلفة من التفاصيل). في بعض الأحيان هناك حاجة إلى متطلبات عالية المستوى بينما في حالات أخرى هناك حاجة إلى مزيد من التفصيل. على سبيل المثال، في دراسة تصميم النظام المتعدد الوكلاء، ممكن فهم الوكلاء الذين يقعون ضمن النظام، ولكن ليس بالضرورة معرفة ماهي صفاتهم وخصائصهم.
التركيبات والتزايدية Modularity & Incremental	القدرة على تعريف النظام بأسلوب تدريجي وتكراري. على سبيل المثال، عندما تضاف متطلبات جديدة يجب أن لا تؤثر على المواصفات الحالية، ولكن يمكن استخدامها.
قابلية التنفيذ والفحص Executability	القدرة على إجراء محاكاة أو توليد نموذج أولي Prototype لبعض الجوانب من المواصفات. وهذه القدرة يمكن أن تعرض بعض من المقدرات المحتملة في النظام التي

	سوف يتم نمذجتها.
التنقية Refinement	عملية النمذجة التي تقوم بتنقية العوامل إلى كيانات مبسطة لتسهيل التعامل معها.
التتبع Traceability	المقدرة على تتبع التبعيات بين النماذج.
الدقة Preciseness	هي سمة من سمات عدم الغموض، وهي تتيح للمستخدمين تجنب سوء التفسير للنماذج الحالية.

1. أسلوب Tropos

يحتوي هذا الأسلوب على مجموعة من الرسومات التخطيطية الواضحة والسهلة التعبير والتي تجعل من السهل فهم الأسلوب. لا تتوفر الجزئية والتزايدية في مرحلة التحليل، ولكنه يوفر أنماط الهيكلية الاجتماعية في مرحلة التصميم، وهذا بدوره يمكن أن يساعد في تجزئة بعض الأنماط. وأيضاً توفير عملية تجزئة الأهداف والمهام الذي يساعد على إدارة التعقيد. يمكن تحقيق التنقية من خلال تحليل الأهداف أو المهام إلى أهداف الثانوية أو مهام الثانوية.

2. أسلوب Gaia

يحتوي هذا الأسلوب على نماذج ومخططات جديدة لتدفق السيطرة التي من الممكن أن تعبر عن مفاهيم النظام ولكنها تكون معقدة أكثر. تكون الأدوار التنظيمية في الفكرة الأساسية لتشكيل الجزئية. تصف النماذج وظائف الأدوار وبروتوكولات التواصل. لا يوجد هناك تنقية للأدوار، ولكن يمكن لمسؤوليات الوكيل أن تجزأ إلى بروتوكولات وأنشطة للتعاون مع الوكلاء الآخرين. التتبع يكون واضحاً في عملية الانتقال من الأدوار إلى أنواع الوكيل.

3. أسلوب MaSE

يحتوي هذا الأسلوب على نماذج ومخططات UML في عملية التحليل والتصميم وهذا يدل على قابلية التعبير وسهولة الفهم. يمكن تشكيل الجزئية من خلال تجميع الأهداف والأدوار لبناء أنموذج الأسلوب. عملية التنقية تكون واضحة في نمذجة الأهداف والأدوار. يمكن الحصول على التتبع بسبب الانتقال بين أنموذج الدور وأنموذج فئة الوكيل.

4. أسلوب MASD

يحتوي هذا الأسلوب على مخططات UCDs و UCMs للتعبير عن المفاهيم في مرحلة المتطلبات، وتستخدم النماذج على شكل جداول واضحة في مرحلة التحليل، هذا يؤدي إلى قابلية التعبير ولكن في نفس الوقت يزيد من التعقيد بسبب كثرة المخططات والنماذج. لا تتوفر في هذا الأسلوب الجزئية والتزايدية في مراحل التحليل والتصميم، ولكنه يوفر أنموذج الأدوار وهذا بدوره يمكن أن يساعد في تجزئة النظام ولكن بدون أنماط. يمكن تحقيق التنقية من خلال تنقية الأدوار في مرحلة التحليل.

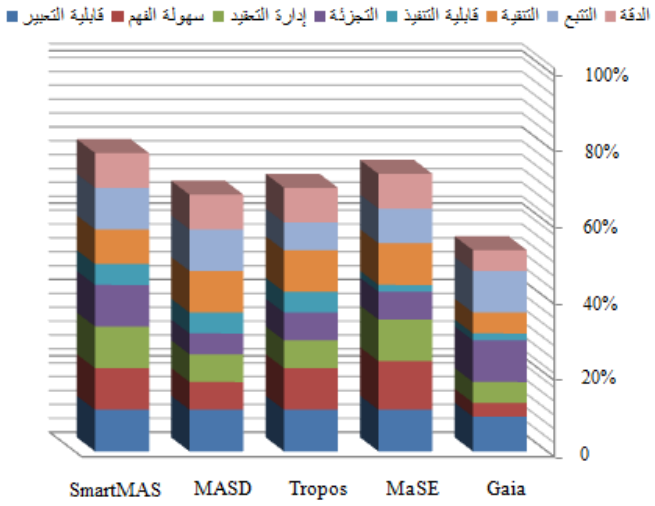
5. أسلوب SmartMAS

تستخدم الرسومات التوضيحية في مرحلة المتطلبات والأنماط للتصميم الهيكلي الموجودة في أسلوب Tropos، والنماذج في مرحلة التحليل من أسلوب MASD، وهذا بدوره يجمع نقاط القوة في هذه الأساليب وزيادة قوة التعبير وسهولة الفهم. وجود عملية تجزئة الأهداف والمهام يساعد على إدارة التعقيد. تتوفر الجزئية والتزايدية في مرحلة المتطلبات المتقدمة، بالإضافة إلى توفير أنماط الهيكلية الاجتماعية في مرحلة التصميم. يمكن تحقيق التنقية من خلال التحليل المنطقي للأهداف أو المهام إلى الأهداف الثانوية أو المهام الثانوية.

يوضح الجدول (4) نتائج تقييم ومقارنة الأساليب لمعايير الترميز والنمذجة، ويصور الشكل (2) مخطط النسبة المئوية لمعدل نتائج تقييم الأساليب للترميزات والنمذجة.

الجدول (4). نتائج تقييم ومقارنة الأساليب لمعايير الترميز والنمذجة

	Gaia	MaSE	Tropos	MASD	SmartMAS
قابلية التعبير	5	6	6	6	6
سهولة الفهم	2	7	6	4	6
إدارة التعقيد	3	6	4	4	6
التجزئة	6	4	4	3	6
قابلية التنفيذ	1	1	3	3	3
التنقية	3	6	6	6	5
التتبع	6	5	4	6	6
الدقة	3	5	5	5	5
نتائج تقييم الأسلوب للترميزات والنمذجة	52%	66%	68%	66%	76%



الشكل (2). معدل نتائج تقييم الأساليب لمعايير الترميزات والنمذجة

3-5 عملية التطوير Development Process

عملية التطوير هي عبارة عن سلسلة من الإجراءات والتغييرات والوظائف التي عندما تنجز تكون النتيجة نظاماً محوسباً. وهي تستخدم دليلاً توجيهياً لجميع الفعاليات خلال مراحل التطوير. يوضح الجدول (5) معايير عملية التطوير في الأسلوب [7].

الجدول (5). معايير عملية التطوير

المعيار	الوصف
تغطية دورة الحياة Lifecycle coverage	عملية التحقق من تغطية دورة حياة البرمجيات الخاصة بأسلوب معين، حيث أن في كل أسلوب هناك دورة حياة مختلفة وقد تحتوي على عناصر مهمة في جميع مراحل التطوير وتؤثر فيها، أو قد تدمج مرحلتين أو تلغى مراحل معينة.
مواصفات النظام System Specification	مقدرة الأسلوب على توفير الطرق لغرض وضع صيغة لتعريف مواصفات النظام.
التصميم الهيكلي Architecture Design	مقدرة الأسلوب على توفير آلية لتسهيل عملية التصميم من خلال استخدام الأنماط أو تجميع الأجزاء.
سياق التطوير Development Context	وهي تحدد ما إذا كان الأسلوب مفيد في تكوين البرامج الجديدة، وعملية إعادة الهندسة Reengineering، وتكوين النماذج الأولية، والتصميم أو مكونات إعادة الاستخدام.
أدوات البناء Implementation Toolkits	توفير الأسلوب على مقترحات بكيفية بناء الوكلاء في النظام.

1. أسلوب Tropos

خلال دورة حياة النظام لا يغطي الأسلوب مرحلة البناء بصورة تفصيلية. يتعامل هذا الأسلوب مع المتطلبات المبدئية والتي يشارك فيها الزبون لتحديد مواصفات النظام. يحتوي على التصميم الهيكلي الذي يتكون من مجموعة أنماط ونظم فرعية. يمكن الاستفادة من الأنماط في عملية إعادة الاستخدام. يوفر هذا الأسلوب مخططات يمكن أن تتعامل مع أدوات البناء التي تتعامل مع هيكلية BDI.

2. أسلوب Gaia

يتعامل هذا الأسلوب مع مرحلتي التحليل والتصميم فقط خلال دورة الحياة. يتم التعرف على مواصفات النظام في مرحلة التحليل من خلال تحليل الأدوار. لا يحتوي على مرحلة التصميم الهيكلي. يمكن إعادة استخدام أنواع الوكيل أو الأدوار. لا يعطي تفاصيل بكيفية التعامل مع أدوات البناء.

3. أسلوب MaSE

خلال دورة حياة النظام لا يغطي الأسلوب مرحلة المتطلبات ولا يعطي تفاصيل بناء النظام في مرحلة البناء. يتم التعرف على مواصفات النظام في مرحلة التحليل من خلال مخططات UCD وتحليل الأدوار. يحتوي على تفاصيل فئات الوكيل والمحادثات في مرحلة التصميم. يمكن إعادة استخدام فئات الوكيل أو مجاميع فئات

الوكيل. يعطي الأسلوب صورة واضحة لكيفية التعامل مع بعض أدوات البناء التي تتعامل مع هيكلية الكائنات الموجهة وذلك من خلال استخدامه الفئات.

4. أسلوب MASD

يغطي الأسلوب جميع مراحل دورة حياة النظام الأساسية. يتم التعرف على مواصفات النظام في مرحلة المتطلبات من خلال مخططات UCDs ومخططات UCMs. لا يحتوي هذا الأسلوب على مرحلة التصميم الهيكلي. يمكن الاستفادة من المقدرات في عملية إعادة الاستخدام ولكن بنفس النظام، ومن الصعوبة إعادة الاستخدام لنظم أخرى مشابهة. يوفر MASD نماذج يمكن أن تعطي صورة واضحة لكيفية التعامل مع جميع أدوات البناء التي تتعامل مع هيكلية BDI.

6. أسلوب SmartMAS

يغطي هذا الأسلوب جميع مراحل دورة حياة النظام الأساسية. يتعامل هذا الأسلوب مع المتطلبات المبدئية والتي يتم فيها مشاركة الزبون لتحديد مواصفات النظام. ويحتوي على التصميم الهيكلي الذي يتكون من مجموعة من الأنماط والنظم الفرعية. يمكن الاستفادة من عملية إعادة الاستخدام وفي نفس النظام باستخدام المقدرات، ومن الأنماط في إعادة الاستخدام لنظم أخرى مشابهة. يوفر نماذج تعطي صورة واضحة لكيفية التعامل مع جميع أدوات البناء التي تتعامل مع هيكلية BDI.

يوضح الجدولين (6، 7) نتائج تقييم ومقارنة الأساليب لمعايير عملية التطوير، ونتائج تقييم ومقارنة الأساليب لتغطية دورة حياة النظام على التوالي، ويصور الشكل (3) مخطط النسبة المئوية لمعدل نتائج تقييم الأساليب لعملية التطوير.

الجدول (6). نتائج تقييم ومقارنة الأساليب لمعايير عملية التطوير

	Gaia	MaSE	Tropos	MASD	SmartMAS
تغطية دورة الحياة	4	4	5	6	6
مواصفات النظام	5	5	7	5	7
التصميم الهيكلي	3	5	6	1	6
سياق التطوير	5	6	5	3	5
أدوات البناء	1	6	5	7	7
نتائج تقييم الأسلوب لعملية التطوير	51%	74%	80%	63%	88%

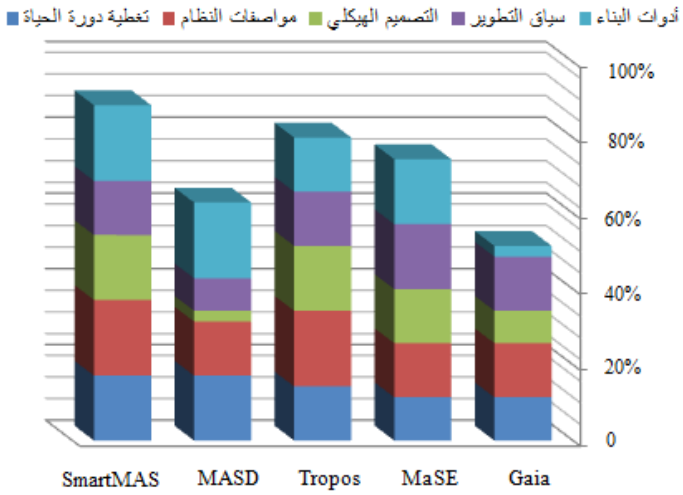
الجدول (7). نتائج تقييم ومقارنة الأساليب لتغطية دورة حياة النظام

	Gaia	MaSE	Tropos	MASD	SmartMAS
المتطلبات	N	N	Y	Y	Y
التحليل	Y	Y	Y	Y	Y
التصميم	Y	Y	Y	Y	Y

البناء	N	No Details	No Details	Y	Y
الفحص	N	N	N	N	N
الصيانة	N	N	N	N	N

4-5 الناحية التطبيقية Pragmatics

تشير الناحية التطبيقية إلى التعامل مع الجوانب العملية لاستخدام الأسلوب. تتناول هذه الفقرة الناحية التطبيقية لاعتماد الأسلوب لمشروع معين أو داخل المؤسسة، يوضح الجدول (8) المعايير اللازمة من الناحية التطبيقية [7].



الشكل (3). معدل نتائج تقييم الأساليب لمعايير عملية التطوير

الجدول (8). معايير الناحية التطبيقية

المعيار	الوصف
الخبرة المطلوبة Required Expertise	وهي الخبرة المطلوبة لتعلم الأسلوب، الصفة التي تميز الكثير من الأساليب هي مستوى التطور الرياضي المطلوب للاستغلال الكامل للأسلوب. بالإضافة إلى الخبرة المطلوبة في لغات النمذجة، والخبرة في تطبيق المنصة الخاصة بهذا الأسلوب.
النمذجة المناسبة Modeling Suitability	هل استخدام الأسلوب مناسب لهيكلية معينة؟ على سبيل المثال BDI.
نطاق التطبيق Domain applicability	هل استخدام الأسلوب مناسب لمجال تطبيق معين؟ على سبيل المثال في الوقت الحقيقي ونظم المعلومات. يجب دراسة هذه المسألة للتحقق ما إذا كان الأسلوب ينطبق لنطاق المسألة المطلوب.
الأدوات المتوفرة Tools available	ماهي الأدوات والموارد المتوفرة التي تستخدم مع الأسلوب في جميع المراحل.
التوسيع Scalability	هل يمكن استخدام هذا الأسلوب للتعامل مع التطبيقات مختلفة الأحجام؟ على سبيل المثال، هل يمكن أن توفر نسخة خفيفة لأبسط المشاكل؟ يجب دراسة هذه المسألة للتحقق ما إذا كان الأسلوب المناسب لمعالجة النطاق المطلوب.

1. أسلوب Tropos

بسبب عدم ذكر التفاصيل الكافية لمرحلة البناء يكون من الصعب تنفيذ النظام بالإضافة إلى الحاجة للخبرة المناسبة لتطوير نظام بصورة جيدة. بالرغم من تعامل هذا الأسلوب مع مخططات UML في مرحلة التصميم، ولكن لا يوجد هناك أدوات متوفرة لتساعد المصممين في مرحلة التحليل وجزء من مرحلة التصميم.

2. أسلوب Gaia

يحتاج هذا الأسلوب إلى الخبرة الكافية لغرض فهم خصائص الحياة والسلامة وتحويلها إلى صيغة تنفيذية. ولكنه لا يحتوي على أدوات للتحليل والتصميم، وهذا يجعل المصممين يبحثون عن طريقة مناسبة لإكمال المخططات بأنفسهم وهذا يتحقق من خلال خصائص الحياة والسلامة.

3. أسلوب MaSE

لا يحتاج هذا الأسلوب إلى خبرة كبيرة وذلك لأنه يتعامل مع مخططات UML. يتعامل مع أداة الوكيل Agent Tool التي تساعد في المصممين في تطوير النظام، يمكن لهذه الأداة أن تولّد العلاقات بين العناصر بصورة آلية، وعملية تدقيق النماذج [8].

4. أسلوب MASD

في مرحلة البناء يتم وصف النظام بصورة تفصيلية وهذا يؤدي إلى سهولة التعامل مع أدوات التطوير وعدم الحاجة إلى خبرة كبيرة في البناء. يستخدم هذا الأسلوب مخططات UCDs التي يتوفر لها العديد من الأدوات، ولكن لا يحتوي على أدوات لمخططات UCMs في مرحلة المتطلبات أو أدوات في مراحل التحليل والتصميم.

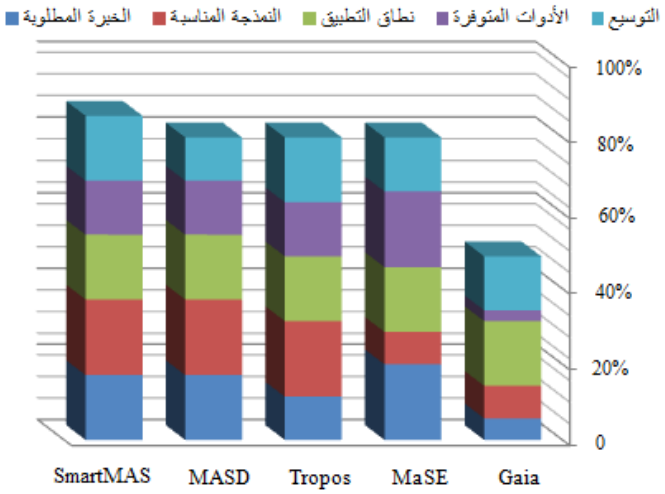
7. أسلوب SmartMAS

في مرحلة البناء يتم وصف النظام بصورة تفصيلية وهذا يؤدي إلى سهولة التعامل مع أدوات التطوير وعدم الحاجة إلى خبرة كبيرة في البناء. لا يحتوي هذا الأسلوب على أدوات للتحليل والتصميم.

يوضح الجدول (9) نتائج تقييم ومقارنة الأساليب لمعايير الناحية التطبيقية، ويصور الشكل (4) مخطط النسبة المئوية لمعدل نتائج تقييم الأساليب للناحية التطبيقية.

الجدول (9). نتائج تقييم ومقارنة الأساليب لمعايير الناحية التطبيقية

	Gaia	MaSE	Tropos	MASD	SmartMAS
الخبرة المطلوبة	2	7	4	6	6
النمذجة المناسبة	3	3	7	7	7
نطاق التطبيق	6	6	6	6	6
الأدوات المتوفرة	1	7	5	5	5
التوسيع	5	5	6	4	6
نتائج تقييم الأسلوب للناحية التطبيقية	**48%**	**80%**	**80%**	**80%**	**86%**



الشكل (4). معدل نتائج تقييم الأساليب لمعايير الناحية التطبيقية

6- الخلاصة والاستنتاجات

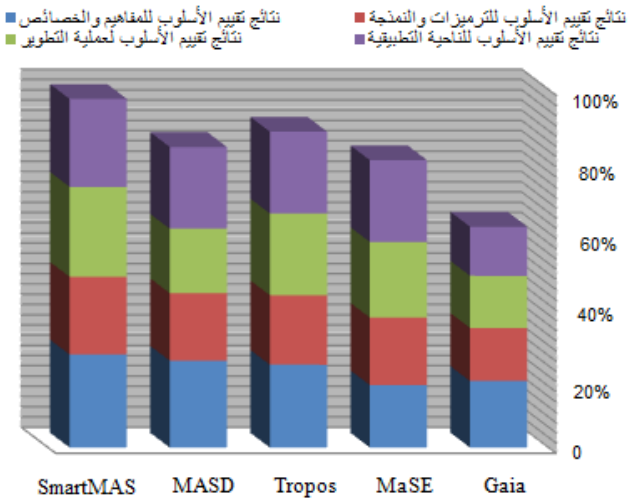
تم في هذا البحث عرض بعض أساليب ABSE، ثم عرض تقنيات وأطر تقييم ومقارنة هذه الأساليب، ثم إجراء عملية تقييم ومقارنة خمسة أساليب لتطوير نظم الوكلاء من ضمنها أسلوب مقترح من قبل الباحثين من خلال استخدام التقنية المعتمدة على الميزات والتي تحتوي على أربعة معايير رئيسية للتقييم وهي المفاهيم والخصائص، الترميزات والنمذجة، عملية التطوير، والناحية التطبيقية، وتم استخدام إطار يحتوي في صفوفه على معايير التقييم وفي أعمدته على أسماء الأساليب، إذ يتم تحليل المعيار وقياسه لمختلف الأساليب ووضع درجة تتراوح بين (1-7) كقيمة تحدد تطابق هذا المعيار مع الأسلوب المقابل، تم وضع الدرجات على أساس دراسة متطلبات نظام التجارة الالكترونية المسند على الوكلاء وتجربته على مراحل التطوير لكل من الأساليب السابقة. يوضح الجدول (10) نتائج تقييم الأساليب، ويصور الشكل (5) معدل نتائج تقييم الأساليب لجميع المعايير.

نلاحظ أن الأسلوب المقترح يعطي نسبة (85%) من تغطية هذه المعايير للنظم التي يتم تطويرها باستخدام هذا الأسلوب، وهو الأفضل بالمقارنة مع الأساليب الأخرى. هذه النتائج تساعد المطورين لاختيار الأسلوب المناسب من خلال اختيار الخصائص المطلوبة لتطوير مشروع معين، بالإضافة إلى أنها تعرض نقاط القوة والضعف في هذه الأساليب مما يساعد في تطوير أسلوب خالي من العيوب في المستقبل.

الجدول (10). نتائج تقييم الأساليب

SmartMAS	MASD	Tropos	MaSE	Gaia	
91%	85%	81%	61%	65%	**نتائج تقييم الأسلوب للمفاهيم والخصائص**
76%	66%	68%	66%	52%	**نتائج تقييم الأسلوب للترميزات والنمذجة**
88%	63%	80%	74%	51%	**نتائج تقييم الأسلوب لعملية التطوير**
86%	80%	80%	80%	48%	**نتائج تقييم الأسلوب للناحية التطبيقية**

ناتج المعدل العام للتقييم	54%	70%	77%	73%	85%



الشكل (5). معدل نتائج تقييم الأساليب لجميع المعايير

7- الأعمال المستقبلية

بالنسبة للأعمال المستقبلية هناك ناحيتان لتطوير العمل في هذا البحث:

1. أساليب ABSE في تطور دائم ولذلك فأن من المهم إجراء التقييم والمقارنة للأساليب الجديدة مع بعضها البعض ومع الأساليب القديمة لتحديد الأسلوب الأفضل والاتجاه نحو أسلوب موحد يحمل جميع نقاط القوة، ولذلك فأن هذا العمل يتم تكراره في المستقبل.
2. عملية تقييم تعقيد الأساليب من خلال حساب تعقيد العناصر البنائية التي يطورها الأسلوب وبالاعتماد على البيانات الإحصائية التي يتم حسابها من خلال المعادلات.

المصادر

[1] حمو، أسماء ياسين، الأسعدي، ماهر طلال، 2011، "أسلوب مقترح لهندسة برمجيات نظم الوكلاء"، بحث مقبول للنشر في مجلة الرافدين لعلوم الحاسوب والرياضيات، جامعة الموصل.

[2] Abdelaziz, T., Elammari, M., and Branki, C., 2008, "MASD: Towards a Comprehensive Multi-agent System Development Methodology" Springer-Verlag Berlin Heidelberg, pp. 108-117.

[3] Ardis, M., Chaves, J., Jagadeesan, L., Mataga, P., Staskauskas, M., Olnhausen, J., 1996, "A Framework for Evaluating Specification Methods for Reactive Systems, Experience Report", IEEE Trans. Software Engineering, Vol. 22, No. 6, pp 378-389.

[4] Bresciani, P., Giorgini, P., Hiunchiglia, F., Mylopoulos, J., Perini, A., 2001, "TROPOS: An Agent-Oriented Software Development Methodology", Technical Report #DIT-02-0015, AAMAS Journal.

[5] Brinkkemper, S., Hong, S., Goor, G., 1993, "A formal approach to the comparison of object-oriented analysis and design methodologies", in Proc.of the Twenty-Sixth Hawaii Intl. Conf., Vol. 4 , pp. 689-698.

[6] Buhr, R., 1998, "Use Case Maps as Architectural Entities for Complex Systems", IEEE Transactions on Software Engineering vol. 24(12), pp.1131-1155.

[7] Dam, K., Winikoff, M., 2003, "Comparing Agent-Oriented Methodologies", Fifth International Bi-Conference Workshop on Agent-Oriented Information Systems(AOIS-2003), Melbourne, Australia, at AAMAS'03, pp. 78-93.

[8] DeLoach, S., 2001, "Analysis and Design using MaSE and agent Tool", Proceedings of the 12th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2001). Miami University, Oxford, Ohio, pp. 45-57.

[9] Jeffrey M., 1997, "An introduction to software agents", Jeffrey M. Bradshaw, editor, Software Agents, AAAI Press / The MIT Press, vol. 43,3 p.

[10] Lind, J., 2000, "Issues in agent-oriented software engineering", In P. Ciancarini and M. Wooldridge, editors, AOSE, volume 1957 of LNCS, Springer, pp. 45-58.

[11] Parandoosh, F., 2007, "Evaluating Agent-Oriented Software Engineering Methodologies", IEEE International Workshop on Soft Computing Applications, Gyula, Hungary – Oradea, Romania, pp 21–23.

[12] Sabas, A., Badri, M., Delisle S., 2002, "A Multidimentional Framework for the Evaluation of Multiagent System Methodologies", Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI-2002), Orlando (Florida, USA), Volume I, pp. 211-216.

[13] Shehory, O., Sturm, A., 2001, "Evaluation of Modeling Techniques for Agent-Based Systems", AGENTS01, Montreal, Quebec Canada, p.91-103.

[14] Siau, K., Rossi, M., 1998, "Evaluation of Information Modeling Methods – A Review", in Proc. 31 Annual Hawaii International Conference on System Science, pp. 314-322.

[15] Siau, K., Cao, Q., 2001, "Unified Modeling Language: A Complexity Analysis, Journal of Database Management", Vol. 12, No. 1, pp. 26-34.

[16] Sturm, A., Shehory, O., 2003, "A Framework for Evaluating Agent-Oriented Method-ologies", Fifth International Bi-Conference Workshop on Agent-Oriented Information System (AOIS-2003), Springer, pp. 94-109.

[17] Tran Q., Low G., Williams M., 2003, "A Feature Analysis Framework for Evaluating Multi-agent System Development Methodologies", ISMIS 2003, pp. 613-617.

[18] Tveit, A., 2001, "A survey of Agent-Oriented Software Engineering", First NTNU CSGSC.

[19] Wood, M., DeLoach, S., 2001, "An Overview of the Multi-agent Systems Engineering Methodology", in Agent-Oriented Software Engineering. P. Ciancarini, M. Wooldridge, (Eds.) Lecture Notes in Computer Science. Vol. 1957, Springer Verlag, Berlin.

[20] Wooldridge, M., Jennings, N. R. and Kinny, D., 2000, "The Gaia Methodology for Agent-Oriented Analysis and Design", Journal of Autonomous Agents and Multi-Agent Systems, vol. 3, pp. 285-312.