A Proposed Methodology for Software Engineering of Agent Systems

Asmaa Yaseen Hammo
asmahammo@uomosul.edu.iq
College of Computer Sciences and Mathematics/University of Mosul

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

Received 2011/05/11 Accepted 2011/06/21

ABSTRACT

In recent years, computer systems that make use of software agents are increased, due to the increased size and complexity of these systems, as well as, because the benefits provided by agents to facilitating the developing process and requirements representation for these systems. Traditional software engineering methodologies are not enough to be used in the development of such systems, because the special characteristics for software agent, therefore there are many proposed Agent Based Software Engineering (ABSE) methodologies to support software engineers in the agents based systems development, However, some of them were designed without adhering to all main stages of software development life cycle, as well as they are lacking for representation of most agents features, additionally, most of them are not represents the implementation phase or represented it but in unclear and unpractical manner.

To address these issues, this research is investigated the existing methodologies and diagnose its strengths and weaknesses, then construct a proposed ABSE methodology depending on combining of strengths for existing methodologies, and delete or modify its weaknesses. This methodology is covered all main stages of software development life cycle, and dealing with problems in the existing methodologies. The requirements phase and part of the design phase were rely on Tropos methodology, whereas the implementation phase and part of the analysis and design phases were rely on MASD methodology, these stages has been modified by deleting its weaknesses and add some other novel techniques.

To improve the new proposed methodology, a running agent based e-commerce system has been developed depending on stages of this methodology, and implement it with Java programming language and XML using the Jadex platform

Keywords: MASD, Agent system, Tropos, AMART, Jadex.

أسلوب مقترح لهندسة برمجيات نظم الوكلاء

أسماء ياسين حمو
كلية علوم الحاسوب والرياضيات
جامعة الموصل

ماهر طلال الأسعدي
هيئة التعليم التقني /الموصل

تاريخ استلام البحث: 2011/05/11 **تاريخ قبول البحث: 2011/06/21**

الملخص

تزايدت في السنوات الأخيرة النظم الحاسوبية التي تحقق الاستفادة من الوكلاء البرمجية، بسبب زيادة حجم وتعقيد هذه النظم، وللفوائد التي تقدمها هذه الوكلاء في تسهيل عملية التطوير والتمثيل لمتطلبات هذه النظم. لم تعد أساليب هندسة البرمجيات التقليدية كافية لاستخدامها في تطوير مثل هذه النظم بسبب المميزات الخاصة للوكيل البرمجي، فاقترحت العديد من الأساليب لدعم مهندسي البرمجيات في تطوير نظم الوكلاء، ولكن البعض منها تم تصميمه بدون التقيد بجميع مراحل دورة حياة النظام، ويفتقر إلى تمثيل أغلب خصائص الوكلاء، كما أن أغلب هذه الأساليب لا تمثل مرحلة البناء أو أن تمثيلها لا يتم بصورة واضحة وعملية.

لتشخيص هذه القضايا، تم في هذا البحث التحقق من الأساليب الحالية وتشخيص نقاط القوة والضعف فيها، ثم بناء أسلوب مقترح لهندسة برمجيات نظم الوكلاء اعتماداً على دمج نقاط القوة في هذه الأساليب وحذف أو تعديل نقاط الضعف مع إضافة بعض التقنيات الأخرى المستحدثة ليغطي هذا الأسلوب جميع مراحل دورة حياة النظام ومعالجة المشاكل الموجودة في الأساليب الحالية.

في مرحلة المتطلبات وجزء من مرحلة التصميم أعتمد على أسلوب Tropos، أما مرحلة البناء وجزء من مرحلة التحليل والتصميم فأعتمد على أسلوب MASD، وتم تعديل هذه المراحل بحذف نقاط الضعف وإضافة بعض التقنيات المستحدثة. لبرهنة الأسلوب الجديد المقترح تم تطوير تطبيق التجارة الالكترونية المعتمد على الوكلاء بالاعتماد على مراحل التطوير في هذا الأسلوب، وبرمجته بلغة جافا واللغة الرمزية الممتدة XML باستخدام منصة Jadex.

الكلمات المفتاحية: MASD ، نظام الوكيل ، Tropos، AMART، Jadex.

1- المقدمة:

الوكيل Agent هو كيان برمجي يمتلك خصائص ذكية مثل الاستقلالية والتفكير والتنقل والاجتماعية والقدرة على التعلم والتعاون والتفاوض، مما يتيح له إنجاز عمله من دون الحاجة إلى التدخل المباشر أو التوجيه من قِبل الإنسان أو الكيانات الأخرى، والتعاون بشكل تبادلي والتواصل مع الوكلاء الآخرين ومع بيئته لإنجاز المهام الخاصة التي لا يمكن أن تؤديها البرمجيات التقليدية. يطلق على النظام الذي يتكون من عدة وكلاء بالنظام متعدد الوكلاء Multi Agent System (MAS)، حيث أن كل وكيل في النظام لديه معلومات و قدرات غير مكتملة لغرض حل المسائل، وبالتالي كل وكيل لديه وجهة نظر محدودة و لا يوجد تحكم عام للنظام وتكون البيانات غير مركزية والمعالجة تكون غير متزامنة [20]. لعملية التواصل بين الوكلاء يتم استخدام لغة تواصل الوكلاء Agent Communication Language (ACL)، التي تحدد أنواع الرسائل المتبادلة ومعانيها، كما أنها أيضاً تحدد قواعد دلالات اللغة، مثل KQML [14]، و FIPA ACL [16].

جذبت نظم الوكلاء الكثير من الانتباه خلال السنوات الماضية، وذلك لأنها قدمت أنموذجاً جديداً لتطوير النظم البرمجية، ودعماً لذلك فقد تم اقتراح العديد من أساليب هندسة البرمجيات لتطوير هذه النظم [3] مثل Gaia [27]، MASE [26]، Prometheus [25]، Tropos [9]، MASD [1]، ولكن البعض منهم تم تصميمه بدون التقيد بجميع مراحل دورة حياة النظام، والبعض الآخر يفتقر إلى تمثيل أغلب خصائص الوكلاء، وأغلب هذه الأساليب لا تمثل مرحلة البناء أو تمثلها بصورة غير واضحة وغير عملية.

تم اقتراح العديد من الأطر لتقييم ومقارنة أساليب تطوير نظم الوكلاء مثل [4] [13] [3] [2] [21]. إذ تم الاتفاق في اغلب هذه الأطر على أنه بالرغم من أن هذه الأساليب وضعت على أسس قوية، ولكنها مازالت تعاني من عدد من نقاط الضعف، وقد تم ذكر نقاط الضعف هذه بالصيغة التالية: لا يوجد أي من أساليب تطوير نظم الوكلاء الحالية مقبولاً بوصفه معياراً، ولا يوجد أي واحد منهم يستخدم بطريقة واسعة. حتى الآن لم يتم بناء مقاييس لهذه الأساليب، والأبحاث التي اختبرت وقارنت خصائص هذه الأساليب اقترحت أنه لا يوجد أي منهم مناسب تماماً للتنمية الصناعية للنظم المتعددة الوكلاء [12]. بالإضافة إلى ذلك، لا يوجد أي من هذه الأساليب كاملة، أي بمعنى تغطية جميع الأنشطة اللازمة المشاركة في دورة حياة تطوير البرمجيات Software Development Life Cycle (SDLC)، أو تكون قادرة على تقديم الدعم الكامل لاحتياجات التنمية الصناعية لنظم الوكلاء. وعلاوة على ذلك، فإن معظم أساليب تطوير الوكلاء تعاني من فجوة بين نماذج التصميم ولغات البناء الحالية [12]. تم في هذا البحث اقتراح أسلوب جديد لتطوير نظم الوكلاء يجتاز نقاط الضعف في بقية الأساليب ويكون مبنياً على أساس المفاهيم والخصائص والنماذج والناحية العملية والتطبيقية.

2- أهداف البحث

يهدف البحث إلى تطوير أسلوب جديد لهندسة برمجيات نظم الوكلاء يحمل الصفات التالية:

1. يكون شاملاً يغطي جميع مراحل تطوير النظام الأساسية.
2. يغطي أغلب خصائص الوكلاء.
3. التعامل مع مفهوم الوكيل بدرجة عالية من التجريد.
4. التعامل مع مرحلة المتطلبات بدرجة مفصلة ذلك لأهمية هذه المرحلة لبقية المراحل ولمشاركة الزبون في هذه المرحلة.
5. التوافق مع مقاييس Foundation of Intelligent Physical Agent (FIPA) وهيكلية Beliefs, Desires, Intentions (BDI).
6. التعامل مع مرحلة البناء بشكل تفصيلي لغرض تنفيذ النظام بسهولة وتوافق النظام قيد التطوير مع أغلب منصات وأطر تطوير الوكلاء.

3. هيكلية BDI

تحتوي وكلاء BDI الاتجاهات العقلانية وهي الحقائق Beliefs والرغبات Desires والنوايا Intentions التي يمكن أن توظف لنمذجة قدراتها المعرفية، وأيضاً لتحديد حالتها الداخلية وتزويدها بالتفكير والاستنتاج [24]. تمثل الحقائق معلومات عن البيئة الحالية، وتمثل الرغبات الإجراءات وبدائلها الممكنة أو الأهداف التي يستطيع الوكيل تنفيذها، أما النوايا فهي تمثل اهتمام الوكيل أو تركيزه أو خططه التي تتحول إلى الإجراءات لتنفيذ الأهداف [8].

4- مقاييس FIPA

منظمة FIPA هي منظمة دولية تكرس جهودها لتعزيز صناعة الوكلاء الذكية من خلال تطوير المقاييس لدعم العمل المشترك بين الوكلاء والتطبيقات المسندة إلى الوكيل. تضم هذه المقاييس بنية الوكلاء وطريقة

تجريدها، طرق التواصل بين الوكلاء، لغات محتوى الرسائل بين الوكلاء، هيكلية الرسائل، بروتوكولات التفاعل بين الوكلاء، والكثير من المقاييس الأخرى [15].

5- أساليب هندسة برمجيات نظم الوكلاء

هندسة برمجيات نظم الوكلاء هي واحدة من أحدث نماذج البرمجة وهي انتقال كبير في هندسة البرمجيات باستخدام نهج جديد أكثر ملائمة من تقنيات الكائنات الموجهة في بناء النظم البرمجية الكبيرة والمعقدة. ولكي يكون المفهوم قابلاً للتطبيق، هناك حاجة لإتباع نهج موحد في كل المراحل خلال دورة حياة تطوير نظام الوكيل والذي يكون مماثلاً للعملية الموحدة Unified Process في تحليل وتصميم الكائنات الموجهة، وهذا النهج يسمى بالأسلوب. يمكن تعريف أسلوب هندسة البرمجيات هو مجموعة منظمة من المبادئ التوجيهية والمفاهيم أو الأنشطة التي تؤدي إلى مساعدة المطورين في القيام في تطوير البرمجيات [19].

1. أسلوب Tropos

هو أسلوب لتطوير البرمجيات ويتم فيه استخدام أنموذج الوكيل البرمجي في عملية تطوير النظام. 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 المبدئي 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 [23]. يحتوي الأسلوب على أربع مراحل رئيسية [9]:

أ- **المتطلبات المبدئية Early Requirements**: يتم في هذه المرحلة فهم المشكلة من خلال دراسة للوضع الحالي، وناتج هذه المرحلة هو الأنموذج التنظيمي الذي يشمل الفاعلين وأهداف وتبعيات كل منهم. المتطلبات المبدئية تشمل مخططين رئيسيين وهما، مخطط الفاعل ومخطط الهدف، إذ يتكون مخطط الهدف من تنقية مخطط الفاعل من خلال التركيز على أهداف فاعل واحد فقط.

ب- **المتطلبات المتقدمة Late Requirements**: في هذه المرحلة يوصف النظام ليكون ضمن بيئته التشغيلية جنباً إلى جنب مع صفاته ومهامه. النظام يكون على شكل فاعل واحد لديه عدد من التبعيات مع الفاعلين الآخرين في التنظيم، وهذه التبعيات تحدد متطلبات النظام الوظيفية وغير الوظيفية.

ج- **التصميم الهيكلي Architectural Design**: يتم في هذه المرحلة تعريف هيكلية النظام العامة التي تتكون من النظم الفرعية التي تكون مترابطة من خلال البيانات والتحكم وغيرها من التبعيات، وتتمحور هذه المرحلة في ثلاث خطوات وهي، تعريف الهيكلية الشاملة. تعريف المقدرات Capabilities. تعريف مجموعة من أنواع الوكيل Agent Types وتوزيع كل واحد منهم أو أكثر على المقدرات.

د- **التصميم التفصيلي Detailed Design**: يتم في هذه المرحلة تعريف سلوك كل عنصر هيكلي على حدة في مزيد من التفاصيل، ويتم تعريف أهداف وحقائق ومقدرات الوكيل بالتفصيل، جنباً إلى جنب مع التفاعل بينهما.

يعتمد Tropos على إطار (i* Framework) كنظرية أساسية لتحليل المتطلبات [22]، إذ يقدم i* بعض المفاهيم مثل الفاعلين، والأهداف، والتبعيات التي تهدف إلى نمذجة الهياكل الاجتماعية.

مزايا أسلوب Tropos:

- يحتوي على مرحلة المتطلبات المبدئية والتي تكون بشكل مبسط وتكون مفهومة للمستخدم العادي، حيث يكون من الممكن أن يشترك الزبون والمستخدم النهائي في هذه المرحلة وتقديم مقترحاته.
- يتم تعريف الأهداف والخطط في مراحل متقدمة جداً من عملية التطوير، مما يمكن الاستفادة منها في عملية التعديل بصورة مبكرة ولا يؤثر على بقية المراحل.
- اعتماد الأسلوب على التبعيات بين الفاعلين وبين أهدافهم في النظام، إذ يمكن الاستفادة منها في أولويات تنفيذ المهام.
- يحتوي على مخطط يمثل النظام الشامل والنظم الفرعية في مرحلة التصميم، وهذا المخطط يعطي وصفاً شاملاً للنظام الكلي.

مساوئ أسلوب Tropos:

- الطريقة المستخدمة في تعريف حقائق الوكيل هي عن طريق الاستنتاج المعتمدة على النتائج السابقة، وهذا يؤدي إلى تراكم الأخطاء في حال حصولها من البداية، بالإضافة إلى عدم وجود نمذجة لهذه الحقائق.
- عدم وجود نماذج للأهداف والخطط لتعريفها بشكل واضح في عملية البناء الفعلي للنظام.
- عدم وجود أنموذج أو هيكلية أو بروتوكول لتعريف التواصل بين الوكلاء.
- عدم وجود تعريف لخدمات الوكلاء في النظام عن طريق دليل الخدمات.
- عدم وجود تفصيل لمرحلة البناء وهذا يؤدي إلى فجوة بين مرحلة التصميم التفصيلي وعملية البناء الفعلي للنظام.

2. 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) [9]، ومخططات حالة استخدام Use Case Diagrams (UCDs)، وأيضاً يدعم هذا الأسلوب البساطة وسهولة الاستخدام. يتألف أسلوب MASD من أربع مراحل رئيسية وهي، مرحلة متطلبات النظام، مرحلة التحليل، مرحلة التصميم، ومرحلة البناء [1].

أ- **مرحلة متطلبات النظام System Requirement Phase:** يتم في هذه المرحلة وصف تفاصيل سيناريوهات النظام بشكل عالي المستوى من خلال أنموذج سيناريو النظام. يستخدم في هذه المرحلة تقنيات UCMs و UCDs لوصف سيناريو النظام. هذه التقنيات تساعد على اكتشاف مكونات النظام مثل (الوكلاء، الكائنات، الموارد، الأدوار). ناتج هذه المرحلة يكون على شكل أنموذج يسمى أنموذج سيناريو النظام.

ب- **مرحلة التحليل Analysis Phase:** يتم في هذه المرحلة وصف هيكلية الوكيل فضلاً عن وصف هيكلية النظام. في مرحلة هيكلية الوكيل يتم وصف النماذج التالية: الأدوار، الوكيل، الحقائق، الأهداف، الخطط، المشغلات. إذ أن دور الوكيل يمثل سلوك الوكيل، وتوفير وسيلة لتحديد وضع الوكيل في النظام، ويتم تحديد مسؤوليات كل دور، وتحديد أنشطة كل مسؤولية. ثم يتم الكشف عن الوكلاء في النظام ويعين لهم الأدوار

المناسبة. أنموذج الوكيل يتألف من خطوتين. الأولى، تعريف الوكلاء داخل النظام وتنقية الأدوار لتناسب مقدرات الوكلاء. والثانية، تعريف الحقائق، والأهداف والخطط والمشغلات. في مرحلة هيكلية النظام يتم إعداد أنموذج لتفاعل الوكلاء مع بعض في النظام باستخدام سيناريوهات UCMs. ويتم إعداد أنموذج علاقات الوكيل ليجسد العلاقات بين الوكلاء، ويتم إعداد أنموذج خدمات الوكيل لكشف الخدمات التي ينبغي أن تتوفر لكل وكيل في النظام.

ج- **مرحلة التصميم Design Phase**: يتم في هذه المرحلة تمثيل مفصل للنماذج وتحويلها إلى تركيبات التصميم. وهذه التركيبات تكون تمهيداً لبناء النظام، إذ تتم مراجعة النماذج التي تم إعدادها في مرحلة التحليل وفقا لمواصفات البناء. تحتوي مرحلة التصميم على ثلاث خطوات وهي، تكوين حاوية الوكيل، تكوين التواصل المتبادل بين الوكلاء، وتكوين دليل الخدمات.

د- **مرحلة البناء Implementation Phase**: يتم في هذه المرحلة بناء النظام اعتمادا على مواصفات المراحل السابقة من خلال النماذج التي يمكن نقلها إلى مرحلة البناء. هذه النماذج لها مجموعة مواصفات التصميم لتعرض كيفية تنظيم نظام الوكلاء ومكوناته.

مزايا أسلوب MASD:

- يحتوي على نماذج لكل من الحقائق والأهداف والخطط والأدوار، وهذه النماذج توضح مكونات النظام وتسهل عملية البناء الفعلي للنظام.
- يحتوي على أنموذج التواصل التبادلي بين الوكلاء، مما يسهل عملية إعداد صيغ الرسائل المتبادلة بين الوكلاء في النظام.
- يحتوي على حاوية الوكيل التي تضم مكونات الوكيل مما يسهل عملية البناء الفعلي للنظام.

مساوئ أسلوب MASD:

- مرحلة المتطلبات تكون معقدة نسبياً، ولا تأخذ بنظر الاعتبار الزبون الذي سوف يشترك في هذه المرحلة وهذا المستخدم احتمال أن يكون ليس لديه الخبرة في هذا المجال.
- يتم تعريف حقائق الوكيل على مرحلتين، إذ يتم تعريف الشروط السابقة واللاحقة في مرحلة مخططات UCDs ومرحلة مخططات UCMs ويتم تحويلها إلى حقائق الوكيل، وهذا يؤدي في بعض الأحيان إلى عدم تطابق هذه الحقائق في المرحلتين، بالإضافة إلى زيادة عدد الحقائق التي تكون فائضة عن الحاجة.
- تعتمد مدخلات مرحلة التحليل في جزء منها على مخرجات مرحلة المتطلبات، وفي الجزء الأخر تعتمد على مخرجات المرحلة نفسها، وهذا يؤدي إلى تزايد الأخطاء في حالة أن المدخلات تكون خاطئة، وحدوث إرباك وعدم المقدرة على معرفة أصل الخطأ وتصحيحه، إذ من المفروض أن تعتمد على مرحلة المتطلبات بشكل كلي.
- يتم تعريف الأهداف والخطط من خلال مراجعة نماذج UCMs وUCDs ومن خلال المهام، ولكن هذه المخططات قابلة للتعديل والمراجعة باستمرار، وهذا سوف يؤدي إلى تغيير الأهداف والخطط في مرحلة التحليل، إذ من المفروض أن يتم تعريف الأهداف والخطط مباشرة في مرحلة المتطلبات.

- يعتمد هذا الأسلوب على المشغلات المحددة مسبقاً في تحديد أولوية تنفيذ الأهداف أو الخطط أو من خلال تحديد هذه الأولويات يدوياً، ولا يأخذ بنظر الاعتبار التبعيات بين الوكلاء والأهداف لتحديد الأولويات، إذ هناك احتمال أن تتغير الأولويات في وقت التشغيل بالاعتماد على الموارد وعلى تنفيذ أهداف أو خطط معينة.
- لا يوجد أنموذج كامل نهائي للنظام يكون على شكل نمط يستخدم في التطبيقات الأخرى المشابهة، أو الرجوع إليه في حالة التعديل والتحديث، ففي حالة تعديل المتطلبات يجب تعديل كافة النماذج في مرحلة المتطلبات والتحليل والتصميم.
- يتم تعريف الخدمات من خلال حالات الاستخدام في مخططات UCDs، وهذا يؤدي إلى عدد كبير من الخدمات التي يمكن أن لا يحتاجها النظام.

6- الأسلوب المقترح Proposed Methodology

تم تطوير أسلوب هجين من أساليب تطوير الوكلاء الأخرى من خلال دمج التقنيات ذات المزايا القوية وإهمال التقنيات الضعيفة، إذ تم تصميم هذا الأسلوب من خلال الاستفادة من مرحلة المتطلبات المبدئية والمتقدمة وجزء من مرحلة التصميم من أسلوب Tropos، والاستفادة من مرحلة البناء وجزء من مرحلة التحليل والتصميم من أسلوب MASD، وتم تعديل هذه المراحل بحذف نقاط الضعف فيها وإضافة بعض التقنيات المستحدثة.

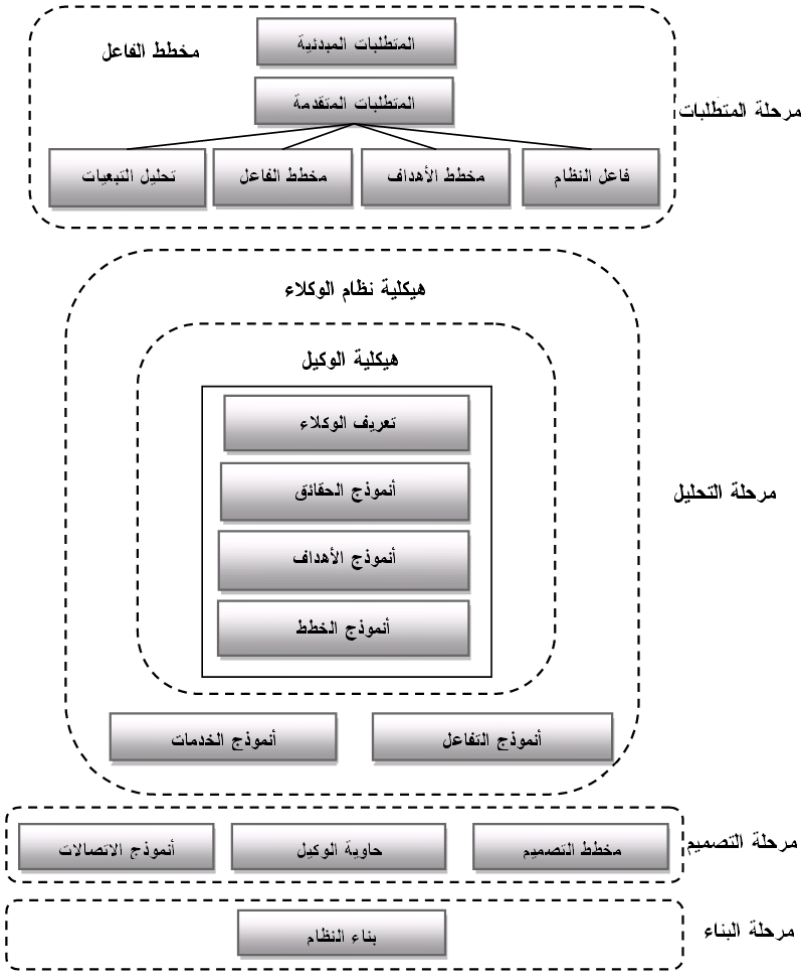
تم اختيار أسلوب Tropos للاستفادة من مرحلة المتطلبات بسبب الاختلاف المهم بين أسلوب Tropos وبين الأساليب الأخرى وهو التركيز على المتطلبات المبدئية والتي يكون فيها الزبون موجوداً ليضع ملاحظاته ليتم توثيقها وتحليلها [23]، والسبب الأخر هو أن أسلوب Tropos يقوم بتحليل الأهداف والمهام والتبعيات في المراحل المتقدمة من تطوير النظام وهذا يقود إلى الوضوح وسهولة تجاوز الأخطاء في المراحل التالية دون الحاجة إلى الرجوع إليها مرة أخرى. تم اختيار أسلوب MASD للاستفادة من بعض التقنيات الموجودة في مرحلة التحليل والتصميم بسبب استخدام الأسلوب لنماذج الحقائق والأهداف والخطط التي تسهل عملية بناء النظام بالإضافة إلى توفر أنموذج التواصل لتسهيل عملية صياغة الرسائل المتبادلة بين الوكلاء. يوضح الشكل (1) عملية التطوير في الأسلوب المقترح.

سوف نستعرض في هذا البحث عملية التطوير في مراحل الأسلوب المقترح مع عرض لمثال تطبيقي وهو "تطبيق التجارة الالكترونية باستخدام الوكلاء البرمجية" لغرض برهنة التطبيق العملي لهذا الأسلوب الجديد.

1-8 مرحلة المتطلبات Requirement Phase

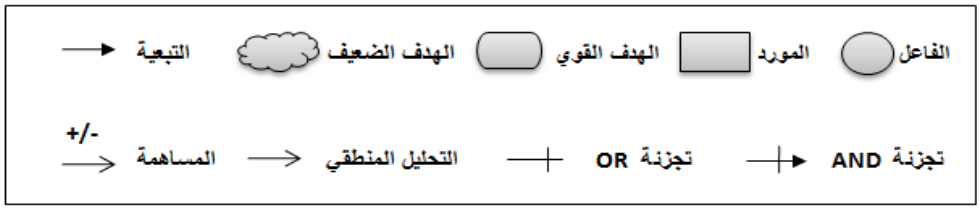
في مرحلة المتطلبات يتم وصف النظام بتمثيل عالي المستوى لتكوين مخطط الفاعل، ويستخدم هذا المخطط لوصف النظام والمكونات والمهام التي يجب أن يؤديها كل مكون داخل النظام، ومفهوم النظام من وجهة نظر المستخدم. يحتوي مخطط الفاعل على **الفاعلين Actors**، وهم الأشخاص أو الكيانات التي تتفاعل مع بعضها البعض أو مع النظام. **الأهداف القوية Hard Goals**، وهي المتطلبات الوظيفية للنظام وتكون على شكل أهداف أساسية أو فرعية. **الأهداف الضعيفة Soft Goals**، وهي تلك الأهداف التي ليس لها تعريف واضح أو معيار لقياس مدى إمكانية استيفائها أو تحقيقها وهي تمثل المتطلبات غير الوظيفية للنظام. **المهام Tasks**، وهي الوسائل اللازمة والفعاليات لتحقيق الأهداف. **الموارد Resources**، وهي الموارد التي يحتاجها الفاعلون لتحقيق

الأهداف. **التبعيات Dependences**، وهي يمكن أن تُفسَر بأن هناك فاعلاً تابعاً Depender وفاعلاً متبوعاً Dependee من خلال وسط التبعية Dependum أو الشيء الذي يطلبه التابع.



الشكل (1). عملية التطوير في الأسلوب المقترح

تحتوي مرحلة المتطلبات على الترميزات الخاصة بجميع المكونات والتي تستخدم في مخطط الفاعل. يوضح الشكل (2) الترميزات المستخدمة في مخطط الفاعل.



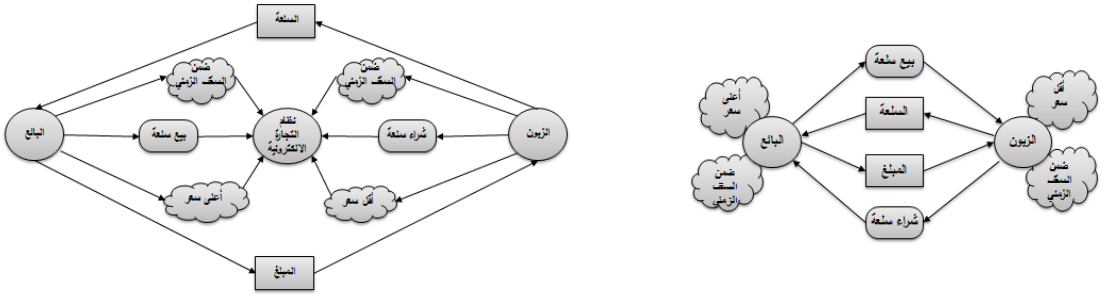
الشكل (2). الترميزات المستخدمة في مخطط الفاعل

تحتوي مرحلة المتطلبات على مرحلتين وهما مرحلة المتطلبات المبدئية والمتقدمة:

1- المتطلبات المبدئية Early Requirements

في مرحلة المتطلبات المبدئية يتم فهم لنطاق النظام بصورة عامة وتمثيله بمخطط بسيط وهو مخطط الفاعل المبدئي، ويتميز هذا المخطط بالبساطة والوضوح للمستخدم. تم وضع المخطط البسيط في هذه المرحلة

وذلك لمشاركة الزبون أو المستخدم النهائي في هذه المرحلة، ولكي تكون المتطلبات واضحة للزبون والمطور وفهم احتياجات الزبون بصورة صحيحة. الشكل (3) في جهة اليمين يوضح مخطط الفاعل المبدئي لتطبيق التجارة الالكترونية.



الشكل (3). مخطط الفاعل المبدئي لتطبيق التجارة الالكترونية

2- المتطلبات المتقدمة Late Requirements

تتكون مرحلة المتطلبات المتقدمة من أربع خطوات وهي إدخال فاعل النظام إلى الأنموذج وتكوين مخطط الأهداف وتكوين مخطط الفاعل النهائي وتحليل التبعيات.

أ- فاعل النظام System Actor

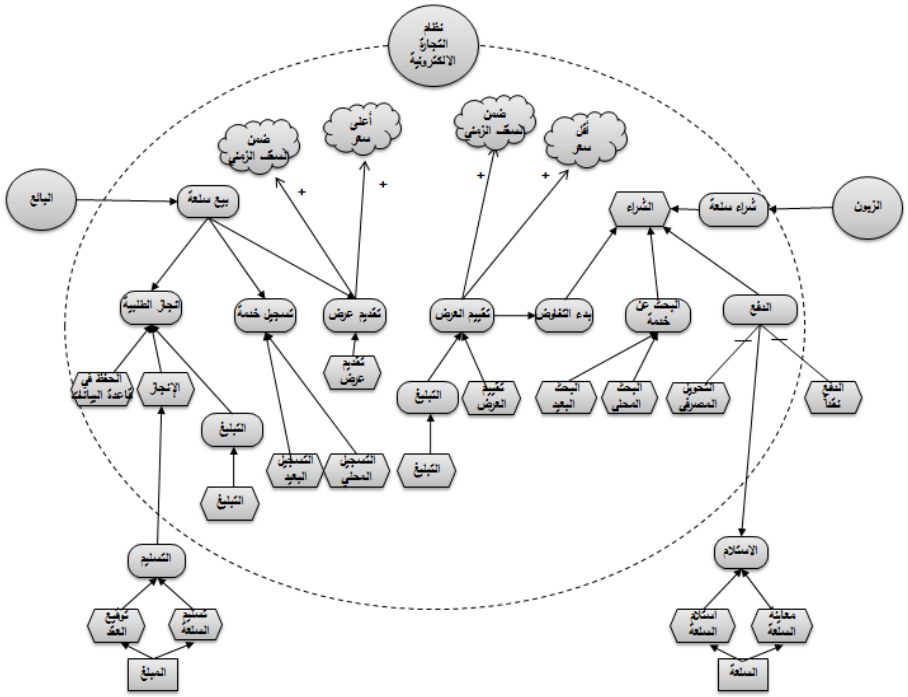
الخطوة الأولى لمرحلة المتطلبات المتقدمة هي إدخال الفاعل الذي يمثل النظام قيد التطوير، ويتم تكرار العمليات السابقة في المتطلبات المبدئية لتتوافق مع الفاعل الجديد، إذ يتم تفويض المهام التي يقوم بها النظام إلى فاعل النظام، وعزل المهام التي لا تنجز داخل النظام. الشكل (3) من جهة اليسار يوضح إدخال فاعل النظام إلى مخطط الفاعل المبدئي لتطبيق التجارة الالكترونية.

ب- مخططات الأهداف Goals Diagrams

في الخطوة الثانية يتم تكوين مخططات الأهداف وتتمحور هذه الخطوة بثلاث مراحل وهي، **تجزئة الأهداف Goal Decomposition**، يتم في هذه المرحلة تجزئة الهدف أو الأهداف إلى أهداف فرعية بطريقة تجزئة AND أو تجزئة OR لتكوين شجرة من الأهداف ولكل فاعل على حدة. **التحليل المنطقي للأهداف Means-end Analysis** يتم في هذه المرحلة التحليل المنطقي للأهداف من خلال تتبع بداية الهدف إلى نهاية الهدف مروراً بجميع الأهداف الفرعية والأهداف الضعيفة والمهام والموارد اللازمة لتحقيق هذا الهدف. **تحليل مساهمة الأهداف Contribution Analysis** يتم في هذه المرحلة تحليل مساهمة الأهداف التي تساهم بشكل ايجابي أو سلبي في تحقيق أهداف أخرى.

ج- مخطط الفاعل النهائي Actor Diagram

يتم في هذه الخطوة تجميع المخططات الخاصة بفاعل النظام والأهداف لتكوين مخطط الفاعل النهائي لمرحلة المتطلبات والذي يحتوي على جميع مكونات النظام، ويتم رسم حدود لتمييز المهام الخاصة بأهداف الفاعلين التي يؤديها النظام. يوضح الشكل (4) مخطط الفاعل النهائي لتطبيق التجارة الالكترونية.



الشكل (4). مخطط الفاعل لتطبيق التجارة الالكترونية

د- تحليل التبعيات Dependency Analysis

يتم في هذه الخطوة تحليل التبعيات بين الفاعلين من خلال الأهداف والمهام والموارد لتحديد الأولويات والشروط السابقة واللاحقة لتسلسل تنفيذ الأهداف والمهام ولتكوين المشغلات التي تقرر أي من الأهداف ومتى سوف يبدأ.

2-8 مرحلة التحليل Analysis Phase

الهدف من مرحلة التحليل هو تحويل متطلبات النظام بشكل واضح لتمثيل النظام بشكل مبدئي وتهيئته لمرحلة التصميم. 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.

1- هيكلية الوكيل Agent Architecture

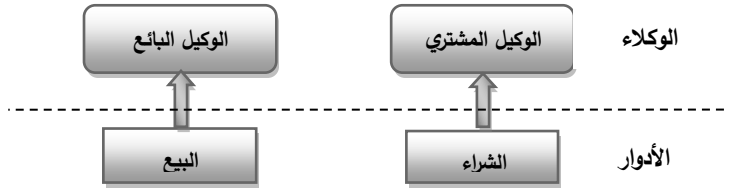
يتم في هذه الخطوة تعريف أنموذج الوكيل الذي يصف الهيكلية الداخلية للوكلاء داخل النظام وكيفية قدرة هؤلاء الوكلاء على توظيف هيكليتهم الداخلية لأداء مهامهم. تستند عملية بناء أنموذج الوكيل على هيكلية BDI. في هذه الهيكلية يمتلك كل وكيل هدف واحد أو أكثر يسعى إلى تحقيقه، وكل هدف يحتوي على مجموعة من الخطط، وكل خطة تتضمن مجموعة من المهام. يمتلك الوكلاء أيضاً المشغلات التي تساعدهم على تحديد الهدف أو الخطة المناسبين ليتم اختياره. تحتوي هذه الخطوة على أربع خطوات فرعية وهي تعريف الوكلاء، تكوين نماذج الحقائق، تكوين نماذج الأهداف، وتكوين نماذج الخطط.

أ- تعريف الوكلاء Agent Identification

يتم في هذه الخطوة تعريف الوكلاء في النظام وتعيين الأدوار اللازمة لهم. يتم الكشف عن الوكلاء من خلال تحليل الفاعلين في مخطط الفاعل، إذ يتم تحليل سلوك الفاعلين في النظام من خلال تتبع مسارات الأهداف في المخطط والكشف عن سلوك فاعل واحد من خلال أهدافه لتمثل دور واحد أو أكثر في النظام، أو الكشف عن

سلوك أكثر من فاعل يقومون بنفس الدور من خلال أهدافهم لتمثل دور واحد في النظام. حيث يمكن بهذه العملية أن يكون كل فاعل على شكل وكيل واحد، أو من خلال دمج عدة فاعلين لتكوين وكيل واحد، وفي هذه الحالة يتم الجمع بين عدة أدوار وتعيينها لوكيل واحد، ثم يتم تعيين الدور أو الأدوار اللازمة للوكيل.

في تطبيق التجارة الالكترونية يتم تحويل كل فاعل إلى وكيل ليتم تعريف وكيلين وهما الوكيل البائع Vender Agent والوكيل المشتري Customer Agent، بالإضافة إلى تعيين الأدوار المستخرجة لكل منهما. يوضح الشكل (5) تعيين الأدوار للوكلاء.



الشكل (5). تعيين الأدوار للوكلاء في تطبيق التجارة الالكترونية

ب- أنموذج الحقائق Beliefs Model

حقائق الوكيل تمثل معلومات الوكيل عن بيئته أو عن نفسه أو غيره من الوكلاء [7]. يتم تعريف حقائق الوكلاء في النظام من خلال تعقب التبعيات الخاصة بكل فاعل واستخراج جميع الشروط السابقة واللاحقة ثم وضعها بجدول ليمثل حقائق الوكيل والتي يجب أن تتلاءم مع هدف أو خطة معينة أو كلاهما، بالإضافة إلى ذلك، الحقائق تخزن معلومات حول الحالة الداخلية للوكلاء. تصنف الحقائق على النحو التالي، **الحقيقة المخزونة Storage Belief**، يستخدم هذا النوع عندما يتم تخزين الحقيقة في قاعدة الحقائق ويمكن للوكيل استخدامها خلال دورة حياته. **الحقيقة المحافظة Maintain Belief**، يستخدم هذا النوع عندما يحافظ الوكيل على قيمة الحقيقة. **حقيقة الإنجاز Achieve Belief**، يستخدم هذا النوع عندما يخزن الوكيل قيمة مطلوبة من الحقيقة، وخلال دورة حياته يحاول أكثر من مرة واحدة التحقق من قيمتها وتشغيل الخطط المناسبة إذا كانت القيمة ليست هي القيمة المطلوبة [6]. ويتم تصنيف الحقائق حسب نوع القيمة إذا كانت ثابتة أو متغيرة.

في تطبيق التجارة الالكترونية ومن خلال تعقب التبعيات الخاصة بفاعلي الزبون والبائع، واستخراج جميع الشروط السابقة واللاحقة، يتم إعداد أنموذج حقائق الوكيل المشتري وأنموذج حقائق الوكيل البائع. يوضح الجدول (1) أنموذج حقائق الوكيل المشتري.

الجدول (1). أنموذج حقائق الوكيل المشتري في تطبيق التجارة الالكترونية

أسم الحقيقة	النوع	الغرض
أسم الزبون	ثابت	للحفظ
أسم البائع	متغير	للحفظ
تم إدخال السلع المطلوب شراؤها	متغير	للحفظ
السلعة معروضة للبيع	متغير	للحفظ
تم العثور على خدمة	متغير	للانجاز
تم إرسال الطلب	متغير	للحفظ
تم إرسال العرض	متغير	للحفظ
قبول العرض	متغير	للحفظ
إنجاز الطلبية	متغير	للانجاز

تم دفع المبلغ	متغير	للحفظ
تم تبليغ المستخدم الحقيقي	متغير	للحفظ
الخدمة	ثابت	للحفظ
السلعة	متغير	للحفظ
تفاصيل المفاوضات	متغير	للحفظ
الوقت	متغير	للحفظ
واجهة الوكيل GUI	متغير	للحفظ

ج- أنموذج الأهداف Goals Model

يمثل الهدف حالة مستهدفة يحاول الوكيل تحقيقها. والهدف يكون سبب تنفيذ إجراءات الوكيل. هناك نوعان من الأهداف: أهداف أساسية وأهداف فرعية، يتم تحقيق الأهداف الفرعية خلال وقت التشغيل، أما الأهداف الأساسية فهي الأهداف الإستراتيجية للنظام [6]. يتم تعريف الأهداف في الأنموذج ولكل وكيل في النظام، ويتم ذلك من خلال تحويل الأهداف الخاصة بالفاعل في أنموذج الفاعل ضمن دوره في هذا الأنموذج وتنقل إلى أنموذج أهدافه، وأيضاً تحويل مهام هذا الدور إلى الخطط الخاصة بهذا الهدف، بالإضافة إلى تحويل الشروط السابقة واللاحقة التي تم تعريفها في مرحلة تحليل التبعيات الخاصة بالهدف إلى أنموذجه.

في تطبيق التجارة الالكترونية يتم تعريف الأهداف الخاصة بالوكيل المشتري من خلال تحويل الأهداف الخاصة بفاعل الزبون في أنموذج الفاعل ضمن دور الشراء وتنقل إلى أنموذج أهدافه. يوضح الجدول (2) أنموذج الأهداف للوكيل المشتري.

الجدول (2). أنموذج الأهداف للوكيل المشتري

أسم الهدف	الشرط السابق	الشرط اللاحق	الخطط
شراء سلعة	• السلعة معروضة للبيع • تم إدخال السلع المطلوب شراؤها	• إنجاز الطلبية	• الشراء
البحث عن خدمة	• تم إدخال السلع المطلوب شراؤها	• تم العثور على خدمة • لم يتم العثور على خدمة	• البحث عن الخدمة
بدء التفاوض	• تم العثور على خدمة	• تم إرسال الطلب	• تقييم العرض
تقييم العرض	• تم إرسال العرض	• قبول العرض • رفض العرض	• تقييم العروض • الرد على العرض
الدفع	• إنجاز الطلبية	• تم دفع المبلغ • لم يتم دفع المبلغ	• التحويل المصرفي
تبليغ المستخدم الحقيقي	• إنجاز الطلبية	• تم تبليغ المستخدم الحقيقي	• تبليغ بالاتفاق والسعر • تبليغ بموعد الاستلام • تبليغ بطريقة الدفع • تبليغ بفشل الشراء

د- أنموذج الخطة Plan Model

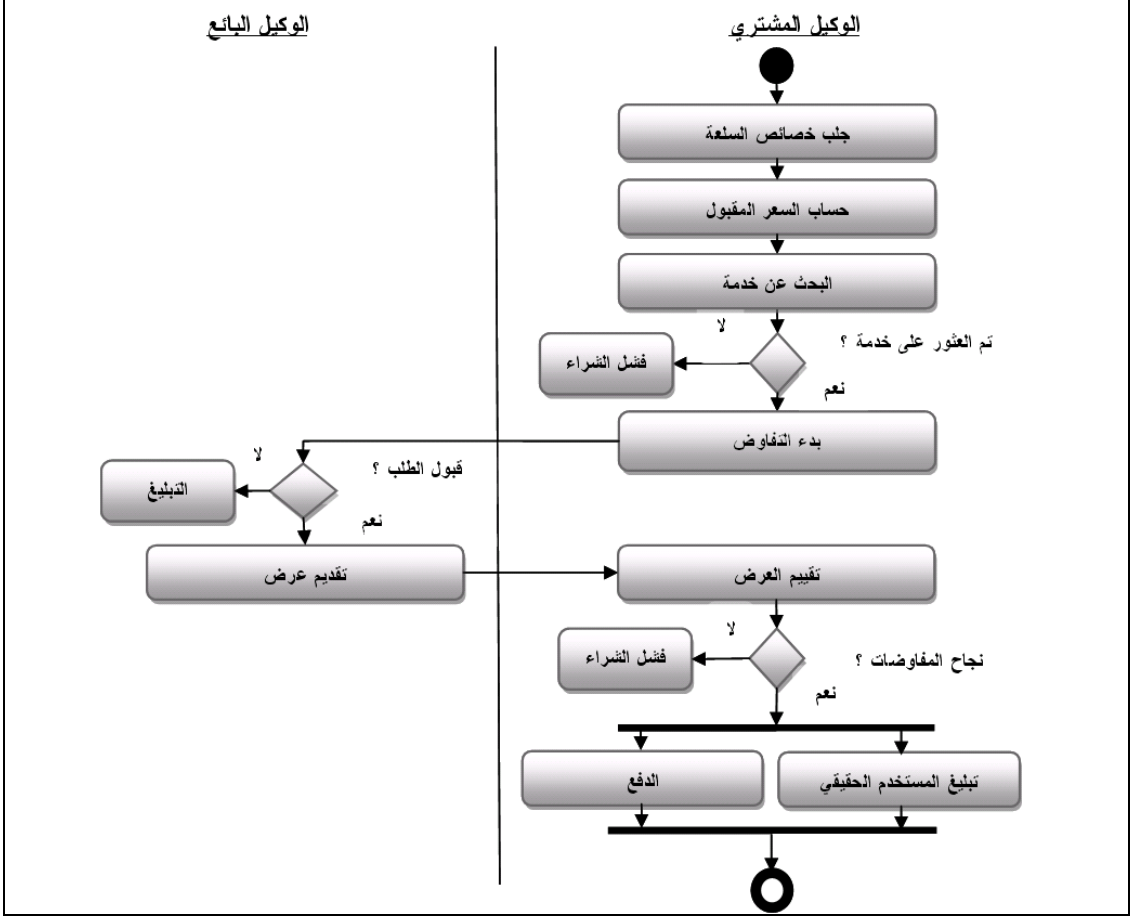
لتنفيذ أهداف الوكيل هناك خطة أو مجموعة من الخطط لكل هدف لغرض تنفيذها من خلال آلية معينة، فالخطة هي وسيلة يمكن من خلالها تحقيق أهداف الوكلاء، يتم تعريف الخطط من خلال تحويل المهام التابعة للأهداف في مخطط الفاعل، أو مباشرة من أنموذج الأهداف.

في تطبيق التجارة الالكترونية ومن خلال تتبع أنموذج أهداف الوكيل المشتري، يتم تكوين نماذج الخطط الخاصة بهذا الوكيل، وبعدها يتم تنقية هذه الخطط من خلال دمج الخطط ذات المهام المتشابهة، أو تجزئة الخطة ذات المهام المختلفة لتكوين خطتين أو أكثر، سوف نستعرض في هذا البحث طريقة عمل أنموذج لخطة واحدة فقط. يوضح الجدول (3) خطة الشراء الخاصة بالوكيل المشتري.

الجدول (3). خطة الشراء للوكيل المشتري

أسم الخطة	الشراء
أسم الهدف	شراء سلعة
الشروط السابقة	تم إدخال السلع المطلوب شراؤها & السلعة معروضة للبيع
الشروط اللاحقة	إنجاز الطلبية
إجراءات نجاح الخطة	تبليغ المستخدم الحقيقي & الدفع
إجراءات فشل الخطة	فشل الشراء
أسم المشغل	هدف شراء سلعة

جسم الخطة

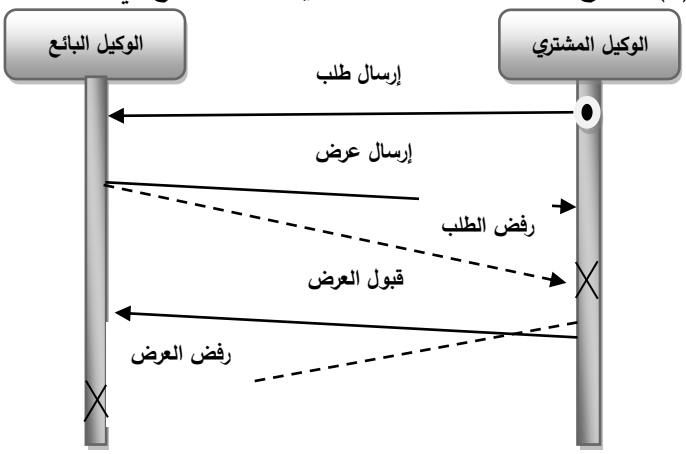


2-كلية نظام الوكلاء MAS Architecture

وهي الجزء الثاني من مرحلة التحليل ويتم فيها تصميم مخطط التفاعل الذي يتم من خلاله الكشف عن جميع التفاعلات بين الوكلاء في النظام. يتم بعد ذلك تصميم أنموذج الخدمات، الذي يتم من خلاله تعريف جميع الخدمات التي يقدمها وكيل معين للوكلاء الآخرين في النظام.

أ- مخطط التفاعل Interaction Diagram

وهو مخطط يصور التفاعل بين جميع الوكلاء في النظام، ويوضح عملية تبادل المعلومات بين الوكلاء مع بعضهم البعض ومع بيئتهم، وهو يصف المحادثات بين الوكلاء على شكل عالي المستوى من التفاعلات ويعد كخطوة أولية لتصميم المراسلات بين الوكلاء. يتم تصميم مخطط التفاعل من خلال الكشف عن التفاعلات الموجودة بين الفاعلين في مخطط الفاعل الناتج من مرحلة المتطلبات عن طريق الاستفادة من التبعيات الموجودة بينهم وتحويلها إلى محادثات. يوضح الشكل (6) أنموذج التفاعل بين الوكيل المشتري والوكيل البائع في تطبيق التجارة الالكترونية.



الشكل (6). أنموذج التفاعل بين الوكيل المشتري والوكيل البائع

ب- أنموذج الخدمات Directory Facilitator Model

يتم في هذه المرحلة إعداد أنموذج دليل الخدمات في النظام، إذ يستخدم دليل الخدمات لعرض الخدمات التي يوفرها بعض الوكلاء في النظام مع المعلومات والوصف الخاص بها والوكيل الموفر للخدمة لكي يستطيع الوكلاء الآخرون تمييز الخدمات التي يعرضها النظام. يتم الكشف عن الخدمات من خلال مخطط الفاعل، إذ يعد الفاعل الذي يكون متبوعاً هو الوكيل موفر الخدمة، والوكلاء التابعين له في تحقيق الهدف الأساسي لهم يعدون الوكلاء المستفيدين من الخدمة.

في تطبيق التجارة الالكترونية تم وضع خدمة (المبيعات) التي يسجلها الوكيل البائع لكي يستطيع الوكلاء المشترون من العثور على الخدمة وتقديم طلب الشراء، بالإضافة إلى ذلك وجود خدمة اختيارية ثانية وهي (المساعدة) التي تستخدم لوصف طريقة البيع والشراء في هذا التطبيق. يوضح الجدول (4) أنموذج خدمات تطبيق التجارة الالكترونية.

الجدول (4). أنموذج خدمات تطبيق التجارة الالكترونية

أسم الخدمة	الوكيل	تاريخ انتهاء الخدمة	الوقت	الكلفة	النطاق
المبيعات	الوكيل البائع	دائماً	دائماً	مجانية	بعيد
المساعدة	الوكيل البائع	دائماً	9 صباحاً 2 ظهراً	مجانية	بعيد

3-8 مرحلة التصميم Design Phase

يتم في هذه المرحلة تمثيل النماذج التي تم بناؤها في مرحلة التحليل بصورة مفصلة، إذ يتم مراجعة هذه النماذج وإضافة بعض التفاصيل وفقاً لمواصفات مرحلة البناء. الهدف الرئيسي من مرحلة التصميم هو الحصول على تصميم الوكيل الهيكلي وتصميم النظام والتي يمكن استخدامها في البناء، بالإضافة إلى الحصول على أنماط يمكن إعادة استخدامها في التطبيقات الأخرى المشابهة أو من خلال مراجعتها لغرض التحديث والصيانة والتعديل من قبل مهندس البرمجيات، والكثير من فوائد التصميم الأخرى. تتمحور هذه المرحلة في ثلاث خطوات: إنشاء مخطط التصميم، تعريف حاوية الوكيل، تعريف أنموذج الاتصالات.

1- مخطط التصميم Design Diagram

يتم في الخطوة الأولى تعريف الهيكلية الشاملة للنظام من خلال تجزئته إلى نظم فرعية وتعريف العلاقات بين هذه النظم التي تكون معتمدة على الموارد والمهام، وهذه النظم الفرعية تكون مترابطة من خلال البيانات والتحكم وغيرها من التبعيات، ويتم تعريف سلوك كل وكيل على حدة في مزيد من التفاصيل، ويتم أيضاً وصف لكيفية عمل مكونات النظام مع بعضهم البعض. بعد تعريف الوكلاء والحقائق والأهداف والخطط في مرحلة التحليل، يمكن إعادة تشكيل مخطط الفاعل الذي تم إعداده في مرحلة المتطلبات لتكوين مخطط التصميم الذي يتم من خلاله وصف الوكلاء في النظام وأهداف وخطط كل واحد منهم بالإضافة إلى عملية التواصل بينهم. الشكل (7) يوضح مخطط تصميم تطبيق التجارة الالكترونية.

2- حاوية الوكيل Agent Container

الخطوة الثانية من مرحلة التصميم هي بناء حاوية الوكيل وهي التي تمثل سلوك الوكيل في النظام. يوجد هدفين أساسيين لحاوية الوكيل، الأول هو تجميع الحقائق والأهداف والخطط الخاصة بوكيل واحد ليكون نظاماً فرعياً في النظام الأساسي، والهدف الثاني هو تجميع الحقائق والأهداف والخطط المتكررة في أكثر من وكيل ووضعها في حزمة واحدة تسمى المقدرة Capability وهي التي يمكن أن يستدعيها جميع الوكلاء بدلاً من تكرار كتابة محتواها، إذ يتم الكشف عن هذه المقدرات من الأدوار التي يلعبها الوكلاء في النظام والتي تم الكشف عنها في مرحلة التحليل. يتم تفصيل نماذج الحقائق والأهداف والخطط التي تم إعدادها في مرحلة التحليل لتكون قريبة لعملية البناء ولتتوافق مع الإطار أو المنصة التي يتم اختيارها لغرض البناء، وفي هذا البحث سيتم إنشاء النماذج لتتوافق مع متطلبات منصة Jadex.

أ- مراجعة نماذج الحقائق Review Beliefs Models

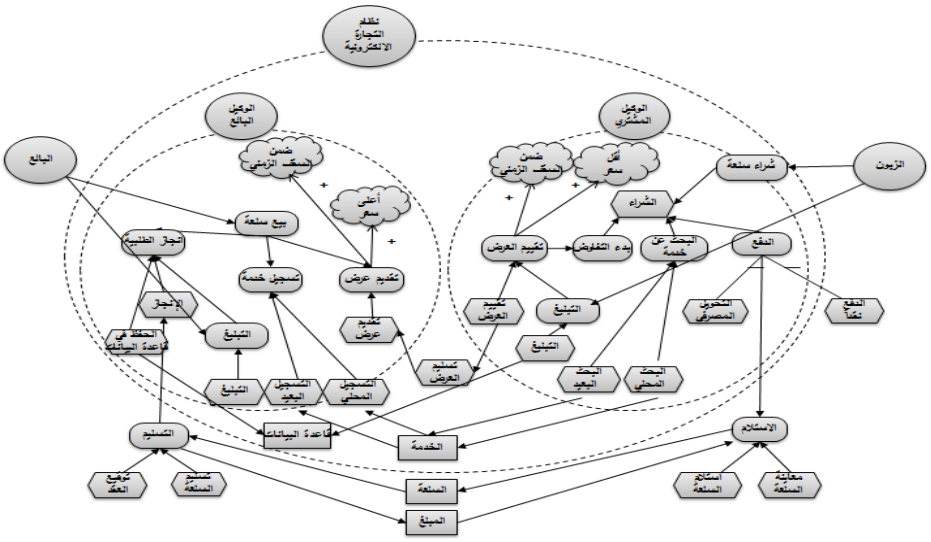
يتم مراجعة نماذج الحقائق في مرحلة التحليل وتحديثها لتتلاءم مع مواصفات التصميم بإضافة حقل الفئة الذي يمثل فئة كائن الحقيقة، والقيمة المبدئية للحقيقة، وصنف الحقيقة الذي يكون على نوعين وهو إما قيمة واحدة One

أو مجموعة قيم Set، والاسم التعريفي Identifier الذي يستخدم في مرحلة البناء. الجدول (5) يوضح الإضافات لأنموذج حقائق الوكيل المشتري.

ب- مراجعة نماذج الأهداف Review Goals Models

يتم في هذه المرحلة مراجعة أنموذج الأهداف الذي تم إعداده في مرحلة التحليل وتحديثه ليتلاءم مع مواصفات التصميم، إذ يتم إضافة حقلي الاسم التعريفي الذي يستخدم في مرحلة البناء، وحقل النوع لتصنيف أهداف الوكيل اعتمادا على متطلبات منصة Jadex.

أن الأهداف في منصة Jadex تصنف إلى أربعة أنواع وهي، هدف الأداء Perform Goal، هدف الانجاز Achieve Goal، هدف الاستعلام Query Goal، والهدف المحافظ Maintain Goal.



الشكل (7). مخطط تصميم تطبيق التجارة الالكترونية

الجدول (5). أنموذج حقائق الوكيل المشتري بعد الإضافات

أسم الحقيقة	النوع	الغرض	الاسم التعريفي	الفئة	القيمة المبدئية	الصنف
أسم الزبون	ثابت	للحفظ	customerName	String	Customer	One
أسم البائع	متغير	للحفظ	vanderName	String	Vender	Set
تم إدخال السلع المطلوب شراؤها	متغير	للحفظ	orderAdded	Boolean	False	One
السلعة معروضة للبيع	متغير	للحفظ	openState	Boolean	False	One
تم العثور على خدمة	متغير	للانجاز	dfFound	Boolean	False	One
تم إرسال الطلب	متغير	للحفظ	requestSent	Boolean	False	One
تم إرسال العرض	متغير	للحفظ	offerSent	Boolean	False	One
قبول العرض	متغير	للحفظ	acceptOffer	Boolean	False	One
إنجاز الطلبية	متغير	للانجاز	doneState	Boolean	False	One
تم دفع المبلغ	متغير	للحفظ	moneyPaid	Boolean	False	One
تم تبليغ المستخدم الحقيقي	متغير	للحفظ	rcNotified	Boolean	False	One
السلع	ثابت	للحفظ	Orders	Order	Null	Set
الخدمة	متغير	للحفظ	dfServiceName	String	مبيعات	One
تفاصيل المفاوضات	متغير	للحفظ	Reports	Report	Null	Set
الوقت	متغير	للحفظ	Time	Long	CurrTime	One
واجهة الوكيل GUI	متغير	للحفظ	Gui	Gui	Constructure	One

هدف الأداء هو نوع من الأهداف الذي يتطلب بعض الإجراءات لتنفيذه، ونتيجة الهدف تعتمد على إجراءات محددة، فعندما لا تكون هناك إجراءات لتنفيذها يفشل الهدف، وعلى عكس ذلك عندما تكون هناك خطة واحدة أو أكثر تتنفذ ينجح الهدف. **هدف الانجاز** هو نوع من الأهداف الذي يستخدم عندما يريد الوكيل تحقيق حالة مستهدفة معينة، وتتمثل هذه الحالة المستهدفة عن طريق شرط، على سبيل المثال عندما يحصل الوكيل على هدف منجز جديد، يبدأ الأنشطة اللازمة لتحقيق الحالة المستهدفة، وعند الوصول إلى الحالة المستهدفة يتم إنجاز الهدف، ولكن إذا لم ينجز إلى حد الآن يتم اختيار الخطط اللازمة لتنفيذها، وهكذا إلى حين الوصول إلى الحالة المستهدفة يتم إلغاء جميع الخطط الخاصة بهذا الهدف. يستخدم **هدف الاستعلام** لاستعلام معلومات حول مسألة محددة، لذلك يستخدم هذا الهدف لاسترداد نتيجة الاستعلام. **الهدف المحافظ** هو الهدف الذي يحافظ على حالة معينة، أي الحفاظ على شرط معين، وعندما لا يتم استيفاء الشرط، يتم استدعاء الخطط اللازمة لإعادة تأسيس الحالة [6].
في تطبيق التجارة الالكترونية يتم إضافة حقلي الاسم التعريفي والنوع لتصنيف أهداف الوكيل في أنموذج الأهداف. الجدول (6) يوضح أنموذج أهداف الوكيل المشتري بعد إضافة الحقول الجديدة.

الجدول (6). أنموذج أهداف الوكيل المشتري بعد إضافة الحقول الجديدة

أسم الهدف	النوع	الاسم التعريفي	الشرط السابق	الشرط اللاحق	الخطط
شراء سلعة	الانجاز	buyGoal	• السلعة معروضة للبيع	• إنجاز الطلبية	• الشراء
البحث عن خدمة	الانجاز	dfSearch	• إدخال السلع المطلوب شراؤها	• تم العثور على خدمة • لم يتم العثور على خدمة	• البحث عن الخدمة
بدء التفاوض	الانجاز	cnpInitiate	• تم العثور على خدمة	• تم إرسال الطلب	• تقييم العرض
تقييم العرض	الاستعلام	cnpEvaluate proposals	• تم إرسال العرض	• قبول العرض • رفض العرض	• تقييم العروض • الرد على العرض
الدفع	الانجاز	payGoal	• إنجاز الطلبية	• تم دفع المبلغ • لم يتم دفع المبلغ	• التحويل المصرفي
تبليغ المستخدم الحقيقي	الانجاز	notifyGoal	• إنجاز الطلبية	• تم تبليغ المستخدم الحقيقي	• تبليغ بالاتفاق والسعر • تبليغ بموعد الاستلام • تبليغ بطريقة الدفع • تبليغ بفشل الشراء

ج- مراجعة نماذج الخطط Review Plans Models

يتم في هذه المرحلة مراجعة نماذج الخطط التي تم إعدادها في مرحلة التحليل وتحديثها لتتلاءم مع مواصفات التصميم، إذ يتم إضافة حقلي الاسم التعريفي الذي يستخدم في مرحلة البناء، وحقل النوع لتصنيف خطط الوكيل اعتماداً على متطلبات منصة Jadex. ذلك أن الخطط تصنف إلى نوعين: خطط الخدمة Service Plans والخطط السلبية Passive Plans. **خطة الخدمة** لديها طبيعة خدمية، نسخة Instance هذه الخطة تتنفذ وتنتظر طلبات الخدمة، وهي تمثل طريقة بسيطة للرد على طلبات الخدمة بأسلوب متسلسل دون الحاجة إلى مزامنة نسخ الخطة المختلفة للخطة ذاتها، ولذلك يمكن لخطة الخدمة تنظيم مهامها في طابور للمعالجة في وقت لاحق حتى عندما تكون مشغولة. يتم تشغيل **الخطة السلبية** فقط عندما يكون لديها مهمة لتحقيقها، وفي هذا النوع من الخطط يجب أن يتم تحديد الأهداف وأحداث المشغلات للسماح للوكيل بمعرفة أنواع الأحداث التي يمكن للخطة أن تتعامل

معها. عندما يستلم الوكيل حدث معين، يجب تحديد خطة أو خطط مرشحة وإنشاء نسخ لها لتنفيذها [6]. الجدول (7) يوضح أنموذج خطة الشراء للوكيل المشتري بعد إضافة الحقول الجديدة.

الجدول (7). أنموذج خطة الشراء للوكيل المشتري بعد إضافة الحقول الجديدة

أسم الخطة	الشراء
الاسم التعريفي	buyPlan
النوع	سلبية
أسم الهدف	شراء سلعة
الشروط السابقة	تم إدخال السلع المطلوب شراؤها & السلعة معروضة للبيع
الشروط اللاحقة	إنجاز الطلبية
إجراءات نجاح الخطة	تبليغ المستخدم الحقيقي & الدفع
إجراءات فشل الخطة	فشل الشراء
أسم المشغل	هدف شراء سلعة

3- أنموذج الاتصالات Communication Model

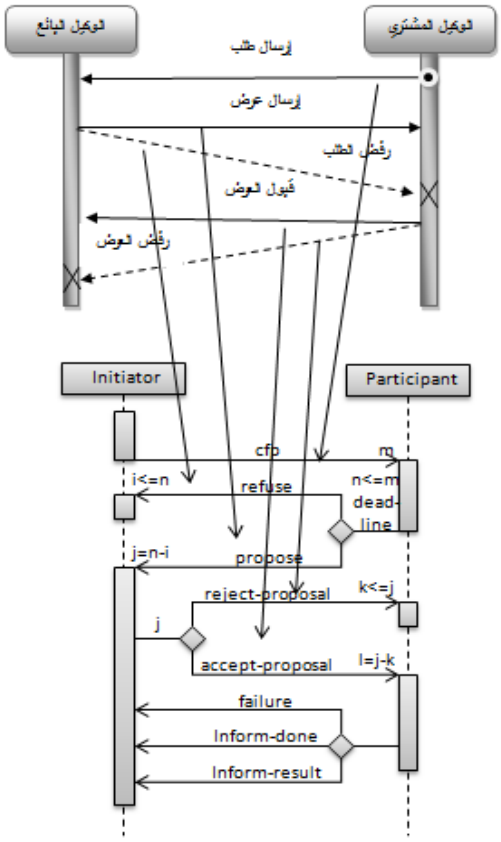
في الخطوة الثالثة من مرحلة التصميم يتم إعداد أنموذج الاتصالات، وهذا الأنموذج يصف بالتفصيل التفاعلات المحتملة بين الوكلاء، وهو يصف المحادثات بين الوكلاء بتفصيل أكثر من أنموذج التفاعل الذي تم إنشاؤه في مرحلة التحليل. يتم إعداد أنموذج الاتصالات من خلال تحويل أنموذج التفاعل الذي تم إعداده في مرحلة التحليل إلى رسائل المحادثة وفقا لأحد بروتوكولات FIPA. ويتم إعداد صيغة الرسائل المتبادلة بين الوكلاء في النظام وتعد الرسائل المتبادلة بين الوكلاء كرسالة FIPA-ACL. في تطبيق التجارة الالكترونية يتم تحويل أنموذج التفاعل إلى بروتوكول التفاوض FIPA Contract Net Protocol [17]، وذلك للاستفادة من خصائص هذا البروتوكول. الشكل (8) يوضح تحويل أنموذج التفاعل إلى بروتوكول التفاوض.

4-8 مرحلة البناء Implementation Phase

مرحلة البناء هي عملية التطوير الفعلية من خلال بدء كتابة الرموز البرمجية للبرنامج. يتم في هذه المرحلة بناء النظام على أساس مواصفات مرحلة التصميم. هناك عدة أطر ومنصات لتطوير نظم الوكلاء، وهذا الأسلوب المقترح يدعم البعض منها مثل JADE [18]، JACK [11]، Jason [5]، Jadex [7]. وجميع المنصات التي تستخدم هيكلية BDI ومقاييس FIPA. في هذا البحث سوف نستعرض مرحلة البناء باستخدام منصة Jadex.

أ- بناء حاوية الوكيل

الخطوة الأولى من عملية البناء هي تحويل حاوية الوكيل التي تم إعدادها في مرحلة التصميم إلى ملف يسمى ملف تعريف الوكيل Agent Definition File (ADF) وهو عبارة عن ملف مكتوب بلغة XML



الشكل (8). تحويل أنموذج التفاعل إلى بروتوكول FIPA Contract Net

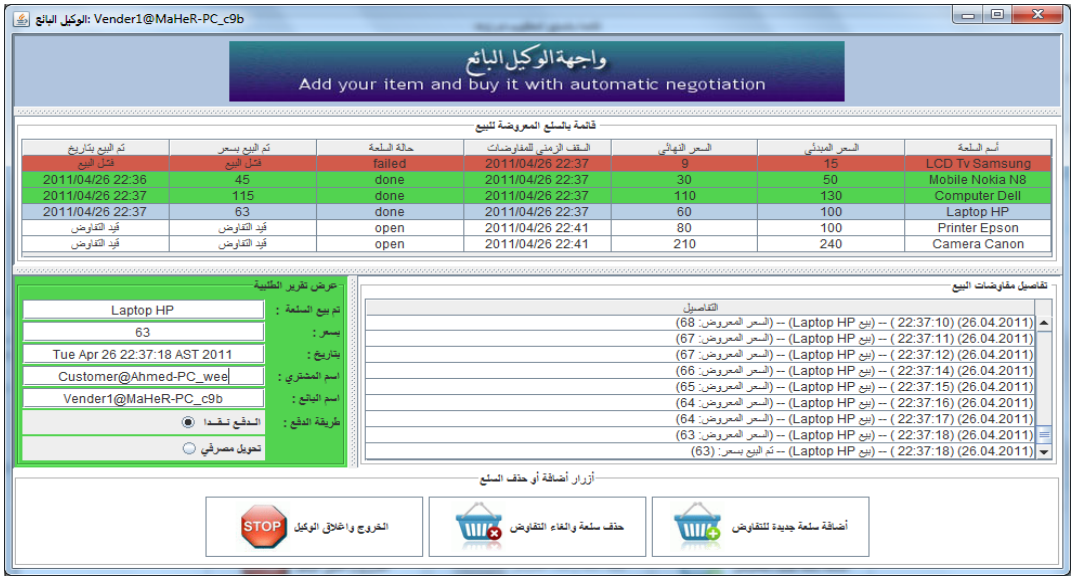
يحتوي على جميع خصائص الوكيل، ويتم تكوين هذا الملف لكل وكيل في النظام. تخضع لغة XML في هذا الملف إلى محددات Jadex Schema المحددة مسبقاً من المنصة. بعد تكوين ملف ADF يتم تحويل الحقائق الموجودة في أنموذج الحقائق الخاصة بالوكيل إلى رموز برمجية بلغة XML في هذا الملف، وبالطريقة نفسها يتم تحويل الأهداف الموجودة في أنموذج الأهداف وتحويل جزء رأس الخطط من نماذج الخطط.

ب- بناء جسم الخطط

بعد عملية بناء ملف ADF، يتم في الخطوة الثانية بناء فئات لغة الجافا من خلال تحويل مخططات الفعالية الموجودة في جزء جسم الخطة في نماذج الخطط إلى الرموز البرمجية بلغة جافا، كل فئة تحفظ في ملف مستقل وبنفس أسم الفئة.

7- تنفيذ تطبيق التجارة الالكترونية

بعد تشغيل منصة Jadex في جهاز الخادم يتم تنفيذ التطبيق وظهور واجهة المستخدم الخاصة بالبائع، يقوم البائع بتسمية الخدمة المراد تسجيلها وإدخال السلع التي يرغب في بيعِها، وفي الحواسيب الأخرى في الشبكة والتي يتوفر فيها منصة Jadex والتطبيق، يقوم الزبون بتنفيذ التطبيق وإدخال أسم الخدمة المراد البحث عنها واسم السلعة التي يرغب في شرائها باختيارها من ضمن السلع المعروضة لديه ويتم ذلك من خلال واجهة المستخدم الخاصة بالزبون. تتم عملية المفاوضات الآلية وسوف تعرض تفاصيل هذه المفاوضات لدى الزبون والبائع. في حالة انجاز الطلبية سوف يتم حفظ تفاصيل الطلبية في قاعدة بيانات لطباعتها كفاتورة بيع وتسليمها للزبون. الشكل (9) يوضح واجهة الوكيل البائع خلال عملية المفاوضات والشكل (10) يوضح واجهة الوكيل المشتري.



الشكل (9). واجهة الوكيل البائع

10- الاستنتاجات والأعمال المستقبلية

يهدف الأسلوب المقترح إلى توفير مجموعة من الطرائق والمبادئ التوجيهية لتسهيل عمل مهندسي البرمجيات في عملية تطوير النظم المتعددة الوكلاء، إذ يغطي الأسلوب المقترح أهم مراحل دورة حياة النظام، بالإضافة إلى أنه يغطي أهم خصائص نظم الوكلاء، وهو يتعامل مع مفهوم الوكيل على مستوى عالٍ من التجريد، ومن الممكن أن يكون قادراً على نمذجة النظم المعقدة والكبيرة نسبياً، وأيضاً توافق الأسلوب مع

الشكل (10). واجهة الوكيل المشتري

مقاييس FIPA وهيكلية BDI، ومن خلال عملية تطوير هذا الأسلوب وتجربته من خلال تطبيق التجارة الالكترونية تم الاستنتاج أن هذا الأسلوب يختلف عن الأساليب الأخرى في الصفات التالية:

- تكامل مراحل التطوير الرئيسية ومن ضمنها المتطلبات المبدئية التي تكون في تواصل مباشر مع الزبون.
- لا توجد إي فجوة بين نماذج التصميم ولغات البرمجة من خلال تفصيل مرحلة البناء التي تستخدم لتحويل نماذج التصميم مباشرة إلى التنفيذ.
- تغطية الأسلوب على النماذج المهمة في تطوير الوكلاء مثل نماذج الأدوار والحقائق والأهداف والخطط ودليل الخدمات والتفاعل.
- إمكانية تحقيق خصائص الوكلاء مثل الاستقلالية والتعاون والتفاوض والتعلم والاستنتاج والتنقل والخصائص الأخرى.
- السهولة في استخدام أحد بروتوكولات التفاوض وعملية التواصل بين الوكلاء.

فيما يتعلق بالأعمال المستقبلية، هناك بعض الأمور التي تحتاج إلى التحقيق فيها مثل مجال نظم الوكلاء التي تكون مستندة على الويب، ومجال الأمنية خاصة في عملية التنقل بين المنصات، ونمذجة الوكيل المنتقل بصورة خاصة، ومرحلة فحص التطبيق، وقضايا إدارة المشاريع مثل المقاييس والجدول الزمني والمخاطر والنوعية.

المصادر

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

[2] Al-Hashel, E., Bala M. and Sharma, D., 2007, "A Comparison of Three Agent-Oriented Software Development Methodologies: ROADMAP, Prometheus, and MaSE" Springer-Verlag Berlin Heidelberg, pp. 909–916.

[3] Basseda, R., Taghiyareh, F., Alinaghi, T., Ghoroghi, C., Moallem, A., 2009, "A Framework for Estimation of Complexity in Agent Oriented Methodologies", IEEE/ACS International Conference on Computer Systems and Applications, aiccsa, pp.645-652.

[4] Bobkowska, A., 2005, "Framework for methodologies of visual modeling language evaluation", Proceedings of the symposia on Meta-informatics, ACM Press, New York.

[5] Bordini, R., Hübner, H., 2005, "Jason", manual, release 0.7, In http://jason.sf.net/.

[6] Braubach, L., Pokahr, A., 2011, "BDI User Guide", In http://jadex-agents.informatik.uni-hamburg.de/xwiki/bin/view/BDI+User+Guide/03+Agent+Specification

[7] Braubach, L., Pokahr, A., Lamersdorf, W., 2004, "Jadex: A Short Overview", Main Conference Net.ObjectDays, Erfurt, Germany, pp.195–207.

[8] Brazier, F., Dunin, B., Treur, J. and Verbrugge, R., 2001, "Modeling Internal Dynamic Behavior of BDI Agents" Kluwer Academic Publishers, The Hong Kong Institute of Education, Dordrecht, Hong Kong, pp. 339-361.

[9] 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.

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

[11] Busetta, P., Ronnquist, R., Hodgson, A. and Lucas, A., 1999, "JACK Intelligent Agents Componenets for Intelligent Agents in Java", updated from AgentLink Newsletter, http://www.agent-software.com.au.

[12] Dam, K., Winikoff, M., 2004, "Comparing Agent-Oriented Methodologies", Giorgini, P., Henderson-Sellers, B., Winikoff, M. (eds.) AOIS 2003. LNCS (LNAI), Springer, Heidelberg, Vol. 3030, pp. 78-93.

[13] Dastani, M., Hulstijn, J., Dignum, F., Meyer, J., 2004, "Issues in Multi-agent System Development", In AAMAS Journal

[14] Finin, T., Labrou, Y., Mayfield, J., 1997, "KQML as an Agent Communication Language" Bradshaw, J. (Ed.) Software Agents. AAAI/MIT Press, pp. 456-463.

[15] FIPA. Foundation for Intelligent Physical Agents, 2002, http://www.fipa.org.

[16] Foundation for Intelligent Physical Agents, 2002, FIPA Communicative Act Library Specification. Document number SC00037J. Geneva, Switzerland. 45 p.

[17] Foundation for Intelligent Physical Agents, 2002, FIPA Contract Net Interaction Protocol Specification. Document number SC00029H. Geneva, Switzerland. 9p.

[18] JADE. Java Agent Development Framework, 1999, http://jade.cselt.it.

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

[20] Jennings, N. R., Sycara, K. and Wooldridge, M., 1998, "A Roadmap of Agent Research and Development", Autonomous Agents and Multi-Agent Systems vol. 1, pp.275-306.

[21] Jia, Y., Huang, C., Cai, H., 2009, "A Comparison of Three Agent-Oriented Software Development Methodologies: MaSE, Gaia, and Tropos", Springer, Lecture Notes in Computer Science, Vol. 4694, pp. 909-916.

[22] Kolp, M., Castro, J., Mylopoulos, J., 2001, "A Social Organization Perspective on Soft¬ware Architectures", In Proceedings of the First International Workshop From Software Requirements to Architectures (STRAW 01) at ICSE 2001, Toronto, Canada, pp. 5-12.

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

[24] Rueda, S., Garcia, A. and Simari, G., 2002, "Argument-based Negotiation among BDI Agents," The Journal of Computer Science and Technology (JCS & T), vol. 2, no. 7, pp.1-8.

[25] Sturm, A., Dori, D., Shehory, O., 2003, "Single-Model Method for Specifying Multi-Agent Systems", The Second International Joint Conference on Autonomous Agents and Multiagent Systems, Melbourne, Australia.

[26] 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, Springer Verlag, Berlin, vol. 1957, pp.207-222.

[27] 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, 3(3).