مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (36) العدد (1) 2014
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (36) No. (1) 2014

نمذجة خدمات التجارة الإلكترونية بالإعتماد على شبكات بتري الملونة

الدكتور جعفر الخير*
الدكتور إبراهيم الشامي**
ماهر وقاف***

(تاريخ الإيداع 17 / 1 / 2013. قُبِل للنشر في 24/ 2 / 2014)

□ ملخّص □

دراسة وتحليل أداء خدمات التجارة الالكترونية عن طريق النمذجة هام للتثبت من اعتماد صحتها وإزالة الحالات غير المرغوبة منها، إذ أن ذلك يمكّن المصممين من استعراض واختبار سلوك الأنظمة قبل تطبيقها وبالتالي فإن تقديم هذه الخدمات بالطريقة المثلى يمكن من تلبية احتياجات العملاء حيث يلعب الزمن دوراً هاماً.

نقدم في هذه الورقة حلولاً لبعض مشاكل نمذجة خدمات التجارة الالكترونية بالاعتماد على شبكات بتري الملونة، والتي كان من الصعب تحقيقها بالاعتماد على شبكات بتري غير الملونة، والفرضية الأساسية للعمل المطروح هي أنه بإمكاننا من خلال هذه النمذجة تحليل كفاءة أداء خدمات التجارة الالكترونية، وبالتالي تصحيح الأخطاء فيها إن وجدت ومن ثم تطوير هذه الخدمات نحو الأفضل بما يساهم في تطوير أداء خدمات نظم التجارة الالكترونية في القطر العربي السوري.

وقد تمت مقارنة طريقة النمذجة المقترحة مع الطرق الأخرى التي استخدمت لنمذجة خدمات التجارة الالكترونية. وقد تم انجاز ما أشرنا إليه من خلال تمثيل النظام ومحاكاته باستخدام المحاكي (CPN نسخة 3.0.2 كانون الثاني 2011).

الكلمات المفتاحية : شبكات بتري، التجارة الالكترونية، النمذجة والمحاكاة والتحليل، شبكات بتري الملونة.

* أستاذ مساعد - قسم هندسة الحاسبات والتحكم الآلي - كلية الهندسة الميكانيكية والكهربائية - جامعة تشرين - اللاذقية - سورية.
** أستاذ مساعد - قسم هندسة التحكم الآلي والحاسبات - كلية الهندسة الميكانيكية والكهربائية - جامعة البعث - حمص - سورية.
*** طالب دراسات عليا - ماجستير - قسم هندسة الحاسبات والتحكم الآلي - كلية الهندسة الميكانيكية والكهربائية - جامعة تشرين - اللاذقية - سورية.

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (36) العدد (1) 2014
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (36) No. (1) 2014

Using Colored Petri Nets for Modeling E-Commerce Services

Dr.Jaafar Alkheer[*]
Dr.Ibrahem Alshami[**]
Maher Wakkaf[***]

(Received 17 / 1 / 2013. Accepted 24 / 2 / 2014)

□ ABSTRACT □

Studying and analyzing E-commerce services are important to know their reliability and removing the undesirable cases in order to enable designers to browse and test the systems behavior before application. Thus, introducing these services in the best way contributes to providing all the clients' needs, with which time plays a major role. We present, throughout this paper, solutions to some of e-commerce services modeling problems depending on colored Petri net, especially those which cannot be solved depending on Petri nets. The basic thesis for the proposed idea is that we are able, through this modeling, to analyze the quality of the performance of e-commerce services, thus correcting the mistakes, if there are any. Developing these services contribute to developing e-commerce services performance in the Syrian Arab Republic. The proposed modeling method is being compared with other methods which have been used to simulate e-commerce services. All the above-mentioned can be done through representing and modeling the system, using colored Petri nets simulator (CPN version 3.0.2, January 2011).

Keywords: Petri nets, electronic commerce, modeling, analysis, simulation, Colored Petri nets.

[*] **Associate Professor, Department of Computer and Automation Engineering, Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia, Syria.**
[**] **Associate Professor, Department of Computer and Automation Engineering, Faculty of Mechanical and Electrical Engineering, Albaath University, Homs, Syria.**
[***] **Postgraduate Student, Department of Computer and Automation Engineering, Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia, Syria.**

مقدمة:

مع انتشار التقنيات الحديثة بشكل كبير وتطور الانترنت ليشمل جميع مناحي الحياة تقريباً ومنها الحياة التجارية ظهر ما يعرف بالتجارة الالكترونية التي لاقت رواجاً هائلاً وسريعاً بين جميع فئات الناس وفي مختلف المناطق، إذ اعتبرت حلاً رائعاً بالنسبة لرجال الأعمال والصناعيين والتجار ومندوبي الشركات لأنها وفرت الكثير من الوقت والمال والجهد وأصبح بإمكانهم عرض منتجاتهم بتكلفة أقل، والوصول إلى أكبر عدد من الزبائن المحتملين بوقت أقصر وتكلفة أقل والتواصل معهم بشكل آني [1].

وبهدف التوجيه الأفضل للزبون والمنتج، والسيطرة على العمليات التجارية بشكل أفضل كان لابد من تمثيل وتحليل خدمات التجارة الالكترونية على الحاسوب [2]، الأمر الذي يساعد أيضاً في تفادي الأخطاء وتصحيحها ووضع العلاقات المناسبة والمرضية لكل من يستخدم التجارة الالكترونية [3] .

هناك العديد من الطرق لنمذجة خدمات التجارة الالكترونية [11] مع وجود العديد من السلبيات والإيجابيات في كل منها وفي بحثنا اخترنا شبكات بتري [5][4] التي تسمح بتحليل عميق للأنظمة بالإضافة إلى أنها من الأكثر شعبية أكاديمياً وتعمل على تحليل الأداء بشكل أكثر وثوقية وأكثر سهولة [6] ويعتبر محاكي شبكات بتري الملونة CPN Coloured Petri Nets [8] لغة نمذجة للأنظمة المتزامنة وهو يجمع بين شبكات بتري ولغة برمجية تتميز بتحرير وفحص النماذج و تعطي نمذجة آلية وتفاعلية، إذ أن برنامج شبكات بتري الملونة يحقق إمكانية تمثيل الشبكة ومحاكاتها ونمذجتها وفق جدول زمني يبين خطوات العمل مع إعطاء تقارير حول النمذجة [9] كما ويعمل على حل جميع مشكلات النمذجة المتعلقة بخدمات التجارة الالكترونية المطروحة في بحثنا والتي يصعب حلها باستخدام شبكات بتري غير الملونة [10].

أهمية البحث وأهدافه:

إن انفتاح الأسواق الإلكترونية وارتفاع حدة المنافسة أدى إلى زيادة وعي المنظمات وشركات التجارة الإلكترونية، فأصبح من الضروري الاهتمام بالخدمات وتسليمها بالشكل المخطط له والذي يكون عادة مبنياً على آراء العملاء واحتياجاتهم، فالمسؤولية تنحصر في تصميم طريقة لتسليم الخدمة للعميل بالطريقة التي يرغبها وضمن المواصفات التي تحقق رضاه. وعليه يتوجب اتخاذ جملة من التدابير والإجراءات التي تساعده على ذلك: كتصنيف الزبائن وتحديد الأولويات ودراسة مدى توافق الخدمة من حيث الزمان والمكان. وهذا يمكن الشركة من تقديم خدمات ذات جودة عالية تحقق لها مجموعة من المزايا تنعكس على الإنتاجية.

طرائق البحث ومواده:

اعتمدنا في هذا البحث طريقة النمذجة والمحاكاة الحاسوبية باستخدام شبكات بتري الملونة وذلك لحل مشكلات النمذجة باستخدام شبكات بتري غير الملونة والتي تخص عمليات محددة وهامة في التجارة الإلكترونية لذا تم إتباع الخطوات الآتية مع كل مشكلة:

1) عرض عملية التجارة الإلكترونية وسبب المشكلة في نمذجتها باستخدام شبكات بتري غير الملونة.
2) الهدف من هذه العملية الإلكترونية.
3) اقتراح الحل المناسب لها باستخدام شبكات بتري الملونة.

4) وضع تطبيق كمثال على العملية الإلكترونية باستخدام محاكي شبكات بتري الملونة Coloured Petri Nets(CPN).

ويتم في هذا البحث مقارنة الطريقة المقترحة مع طرق النمذجة الأخرى التي قامت بنمذجة عمليات التجارة الالكترونية.

النتائج والمناقشة:

التجارة الإلكترونية شكل من أشكال الصفقات التجارية التي يتصل أطرافها (البائع والمشتري) بعضهما البعض عبر شبكة المعلومات الدولية سواء على المستوى المحلي أو الدولي ولكن لها العديد من المخاطر حيث تمت صياغة النصوص القانونية على أساس التعامل الورقي والوثائق وضرورة التوقيع بوسائل كالختم والبصمة، ولهذا فإن الانتقال من مرحلة التعامل الورقي إلى الإلكتروني دون تحليل وتمثيل ونمذجة المعاملات الإلكترونية لحماية الحقوق المتولدة من هذه التجارة وإثباتها سوف يؤدي إلى فراغ قانوني يهدد حقوق المتعاملين ويقلل فرص الاستفادة من استخدام التطور التكنولوجي في وسائل الاتصال.

استخدام شبكات بتري الملونة من أفضل الطرق لنمذجة خدمات التجارة الإلكترونية مع وجود العديد من المشكلات التي تعيق النمذجة بهذه الطريقة وسيتم التركيز لحل بعض هذه المشكلات والتي تناسب بحثنا.

مثال عملي على التجارة الالكترونية:

في نظام الشبكة العالمية (W3C) فإن خدمات التجارة الالكترونية هي تطبيق برمجي معتمد على URI الذي يشكل الواجهة والبناء القادر على التعريف وهو موصوف ومكتشف بواسطة XML ويدعم التفاعلات المباشرة مع التطبيقات البرمجية الأخرى باستخدام رسائل XML عبر الانترنيت والتي هي عبارة عن لغة مقدمة بواسطة ائتلاف W3C من اجل تبادل المستندات والوثائق. ولنأخذ مثالاً عملياً على عملية شراء بسيطة متمثلة بثلاثة مشاركين هم الزبون والتاجر والمنتج كما في الشكل (1) وعلى فرض أن السلع المباعة هي سلع معلومات (مجلات، كتب..) سيكون سيناريو تنفيذ الصفقة كالتالي:

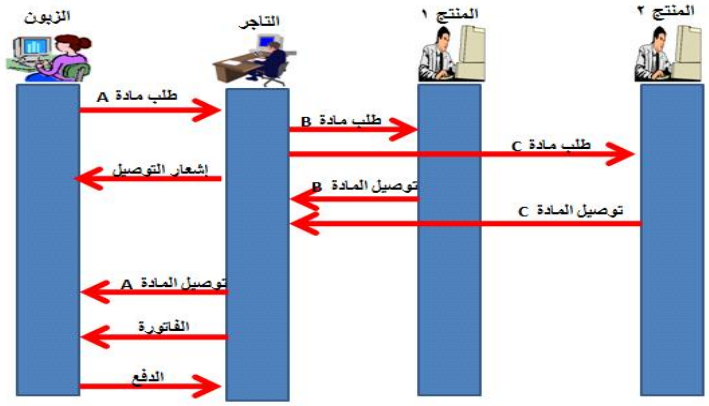
1- يرسل الزبون طلب المادة A للتاجر.

2-يستقبل التاجر الطلب ويرسل للمنتجين طلبات للحصول على مكونات المادة A ولتكن B,C.

3- يرسل التاجر إشعار بالتوصيل للزبون.

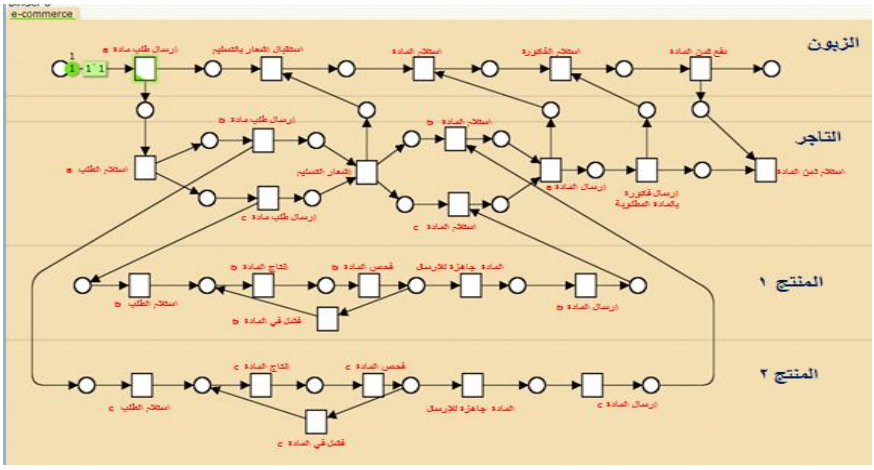
4-يستقبل التاجر مكونات المادة A ويرسلها إلى الزبون ويرسل فاتورة الشراء.

5-يستلم الزبون الفاتورة ويقوم بالدفع.



الشكل (1) عملية الكترونية متمثلة بصفقة شراء

ولتمثيل النظام باستخدام شبكات بتري PN نأخذ بعين الاعتبار أن حواجز شبكة بتري transitions تمثل عمليات التجارة الإلكترونية بينما الحلقات الدائرية لشبكات بتري places تمثل حالات التجارة الإلكترونية أما الأسهم arcs بين الحلقات الدائرية والحواجز تستعمل لتحديد العلاقات السببية وأخيراً طلقات شبكات بتري Token تستعمل لتحاكي الفعاليات الحالية والديناميكية للنظام كما في الشكل (2).



الشكل (2) تمثيل عملية الشراء باستخدام شبكات بتري PN

ملاحظة في الشكل (2) استخدمنا محاكي بتري الملونة لتمثيل الشبكة فقط ولم نستخدم الأدوات والشروط والتوابع والطلقات الملونة والتعاريف والتصاريح ليبقى النموذج شبكة بتري فقط PN وليس شبكة بتري الملونة CPN.

ولتوسيع نموذج الشراء السابق ليشمل عمليات الدفع الإلكتروني (مشاركة البنك) مع ضمان حماية المال للزبون وضمان توصيل مؤكد للسلع مع الأخذ بعين الاعتبار حالات الفشل والأخطاء المتوقعة والمخاطر من عمليات الاحتيال وحالات الإلغاء المتوقعة من قبل كل طرف من المشاركين فإننا غير قادرين باستعمال شبكات بتري PN لتمثيل النموذج وهذا بسبب الحاجة لمرشحات تضمن تحديد شروط معينة للرسائل المتبادلة والحاجة لاستخدم معرفات للكائنات والرسائل لضمان عدم ضياع البيانات وتفادي حالات الانتظار غير المبرر والحاجة لإجراء عملية واحدة على مجموعة من الكائنات أو كائن محدد ضمن مجموعة بالإضافة لتنظيم وترتيب طلبات الزبائن بحسب الأولوية وتنفيذ العمليات بحسب شروط محددة وإنشاء قواعد بيانات مشتركة مركبة وإدارتها وتحديد الوصول إليها مع الأخذ بعين الاعتبار مفهوم التزامن، وسنقوم بتوضيح طرق الحل باستخدام المحاكي CPN لشبكات بتري الملونة [12].

حل مشكلات نمذجة خدمات التجارة الالكترونية بالاعتماد على شبكات بتري الملونة:

إن خدمات التجارة الالكترونية هي في الحالة العامة عبارة عن خدمات مركبة من عدة خدمات بسيطة، فالخدمة الواحدة إما أن تؤدي إلى خدمة أخرى تليها أو أن تعتمد على خدمة سابقة وعلى سبيل المثال ليكن لدينا شركة تجارية تعمل على بيع الكتب: هناك خدمات خارجية كالدفع وتحويل المال بالإضافة إلى النقل والشحن أو الاستيراد والتصدير والاتفاقيات التجارية مع شركات منافسة على الشبكة، أي أن تركيب خدمتين أو أكثر يؤدي إلى توليد خدمة جديدة تعمل على تفعيل وتطوير السلوك الفردي لكل منهما والسلوك التعاوني الجديد بهدف تسويق خدمة جديدة. ومن الصعب تحليل وتمثيل عمليات التجارة الالكترونية باستخدام شبكات بتري PN بسبب العديد من المشكلات التي تواجهنا من حيث التطبيق والتي سنقوم بحلها أدناه باستخدام شبكات بتري الملونة CPN وهي لغة نمذجة للأحداث المنفصلة تدمج بين شبكات بتري PN ولغة برمجة رسومية عالية المستوى ML.

نموذج CPN للنظام يصف حالة النظام والأحداث التي تسبب تغير في حالة النظام، حيث عن طريق النمذجة يصبح من الممكن تحقيق سيناريوهات مختلفة واستكشاف سلوك النظام وغالبا يكون هدف النمذجة هو تنقيح وتصحيح والتحقق من النظام المصمم، ومن الممكن مراقبة الأحداث وتأثير الخطوات الفردية مباشرة على التمثيل التخطيطي لنموذج CPN الذي يستعمل من اجل فحص وتحليل الأداء.

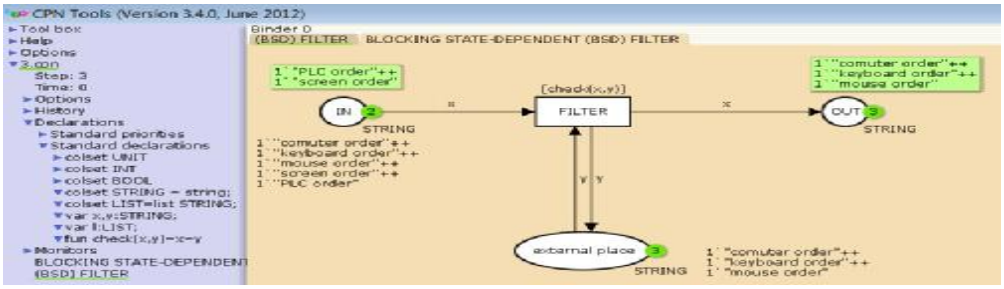
في هذا الفصل سنقوم بحل /14/ مشكلة من المشاكل الأكثر أهمية والتي تواجهنا أثناء القيام بتحليل وتمثيل عمليات التجارة الإلكترونية باستخدام شبكات بتري:

1- ترشيح الطلبات غير المرغوبة بالاعتماد على بيانات خارجية:

عرض المشكلة: غير مسموح في شبكات بتري (غير الملونة) بتخزين المعلومات في حلقة خارجية وبالتالي هناك مشكلة عند تحديد دخول الكائنات بالاعتماد على معطيات محددة في الشبكة، وعلى سبيل المثال في احد أقسام الشركة لا تقبل الطلبات إلا من قبل موظفي الشركة فقط والمسجلة أسمائهم في قائمة خاصة بالقسم.

الهدف: الهدف هو منع مرور البيانات وذلك بعد مقارنتها مع بيانات خارجية، في بعض الحالات فإن تحليل البيانات لا يعتمد فقط على خواص البيانات ذاتها وقيمتها بل يمكن أن تعتمد على بيانات خارجية مخزنة.

الحل: لحل هذه المشكلة نستعمل مرشح Blocking State-Dependent (BSD) Filter الذي يستقبل البيانات من حلقة الدخل ويفحصها مع مجموعة بيانات خارجية وبحسب شروط الترشيح يمرر بيانات إلى حلقة الخرج، ويتم هذا بوصل حاجز المرشح مع بيانات مخزنة بحلقة خارجية تساعد بعملية ترشيح البيانات، ملاحظة: يجب دائما وجود بيانات في الحلقة الخارجية أي طلقة واحدة على الأقل في الحلقة تمثل قاعدة البيانات المشتركة، وفي الشكل (3) حارس حاجز المرشح يتضمن تابع لفحص العنصرين x , y الذي يفحص عنصر بيانات الدخل x وعنصر بيانات المجموعة الخارجية y لمطابقة خواص البيانات إذا كانت متوافقة، والحلقة الخارجية تخزن مجموعة من قيم كل البيانات المارة إلى حلقة الخرج والتابع check يفحص العنصرين (x , y) حيث عنصر البيانات x محتوى في القائمة الموجودة في الحلقة الخارجية والتي تحدد شروط الترشيح.



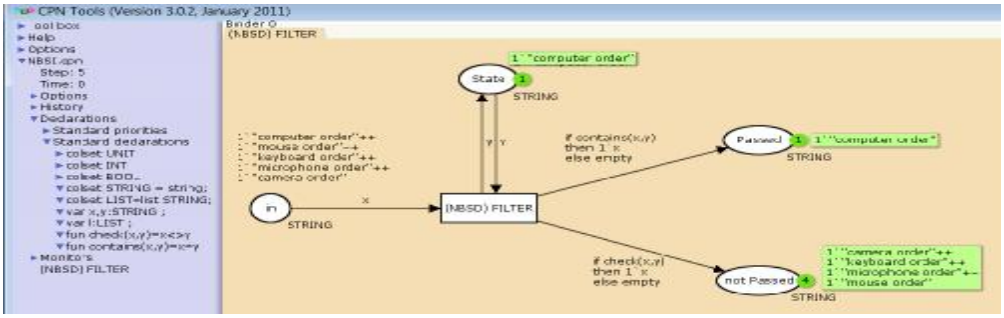
الشكل (3) حل مشكلة ترشيح الطلبات غير المرغوبة بالاعتماد على بيانات خارجية باستخدام شبكات بتري الملونة

2- ترشيح الطلبات غير المقبولة بالاعتماد على بيانات خارجية مع تفادي تراكم البيانات بالدخل:

عرض المشكلة: إن عنصر البيانات الذي لا يمتلك خاصية معينة تحددها مجموعة بيانات خارجية والذي لا يستطيع المرور يتراكم في حلقة الدخل، هذا الحل لا يسمح للبيانات غير المناسبة بالاستعمال في مكان آخر في المعالجة بحجبها في حلقة دخل المرشح. وعلى سبيل المثال معالجة طلبات التأمين حيث كل الطلبات القادمة يجب أن تحلل وتعطي النتيجة فوراً أو أن تعالج بالتفصيل إن كان فيها مشكلة.

الهدف: العمل على ترشيح البيانات (طلبات الزبائن) التي لا تتوافق مع مجموعة خواص محددة مرفقة وذلك بمقارنتها بمجموعة بيانات خارجية مع تفادي تراكم البيانات غير الخاضعة لهذه الخواص في دخل المرشح.

الحل: نستخدم مرشح Non-Blocking State-Dependent (NBSD) الذي يستقبل البيانات من حلقة الدخل ويفحص حالة مجموعة البيانات الخارجية ويمرر عناصر البيانات التي تحقق الشروط إلى خرج محدد بينما بقية البيانات تمرر إلى خرج آخر. وفي الشكل (4) نصل حاجز المرشح مع حلقتي خرج، حلقة لتمرير البيانات المتوافقة إليها والحلقة الأخرى لعناصر البيانات غير المتوافقة، الأسهم التي تصل حاجز المرشح مع حلقات الخرج تكون مرتبطة بشروط الترشيح وبمعنى آخر عناصر البيانات تمر عبر المرشح بحسب الشرط المحدد إلى الخرج المحدد في الشكل فقط عناصر البيانات المخزنة ضمن الحلقة الخارجية تمر إلى الخرج المحدد، هذا الحل يضمن أن كل البيانات القادمة تعالج مع تفادي تراكم البيانات في حلقة دخل المرشح.



الشكل (4) حل مشكلة تراكم الطلبات الغير مرغوبة في دخل المرشح الذي يعتمد على بيانات خارجية

3- ضياع وفقد طلبات الزبائن بعد مرورها في النظام:

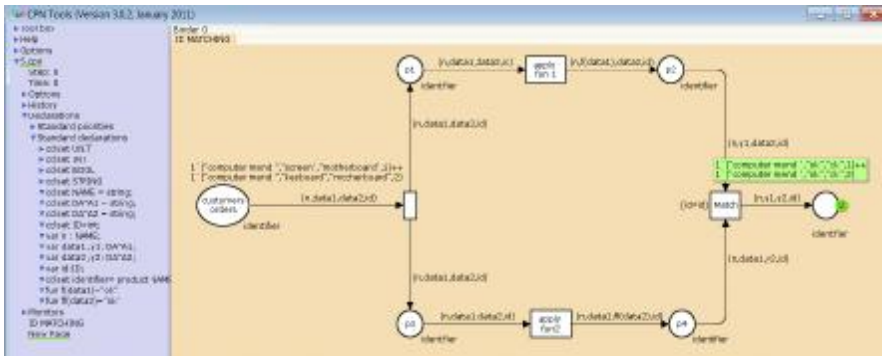
عرض المشكلة: إن إجراء عدد من العمليات على بعض الطلبات يفقد التعريف الكامل للطلب الأصلي وقد تكون القيمة الناتجة بواسطة العمليات المتتابعة منسجمة مع الكائن الأصلي وهذا سلوك غير مرغوب به يعطي نتائج

غير معروفة، وعلى سبيل المثال كأن يطلب زبون إصلاح حاسب شخصي فلا بد من الاحتفاظ بمعرفات الطلب الرئيسي لحساب الأعمال المقدمة وتقدير النتائج.

الهدف: تزويد الكائنات المتشابهة بمعلومات خاصة لتمييزها، قد تحتوي الحلقة العديد من الكائنات من نفس النوع، وفي بعض الحالات من الضروري مقارنة القيمة الأصلية للكائن قبل وبعد التعديل (المعالجة) لمعرفة النتيجة، قيمة الكائن قد تتغير وقد يخسر التعريف الكامل وبالتالي يستحيل مطابقة أي قيمة معدلة مع القيمة الأصلية.

الحل: لحل هذه المشكلة نربط كل كائن مع معرف أو أكثر خاص به، توفر المعرفات تجعل من الممكن تمييز الكائنات من نفس النوع حتى ولو امتلكوا القيمة نفسها، لون الطلقة يرتبط مع نوع الكائن لتمييزه وتستبدل بنوع مجموعة أي الكائن مع قيمته مع المعرف الخاص به، في محاكي شبكات بتري الملونة نستخدم ID تعريفاً عشوائياً للكائن (INT, STRING, ...) ففي حالة integers، المعرف قد يكون 3,2,1... الخ ويمكن أن يستعمل نوع مركب، حيث لكل كائن معرف فريد خاص به، كأن نستبدل المتغير data نوع سلسلة T بالمجموعة (n,data1,data2,id) حيث id نوع المعرف ID وبالتالي نشير للكائن بمعرف مرتبط بقيمة الكائن.

وفي الشكل (5) نلاحظ تطابق كائنين مع اختلاف المعرف وإذا تغيَّرت قيمة الكائن فمن الممكن تمييز الكائن عن طريق المعرف. يطبق هذا النمط عند الحاجة للعودة للكائن الأصلي بشكل دقيق بين مجموعة من الكائنات المتشابهة بالرجوع إلى المعرف بدلا من القيمة الفعلية للكائن.



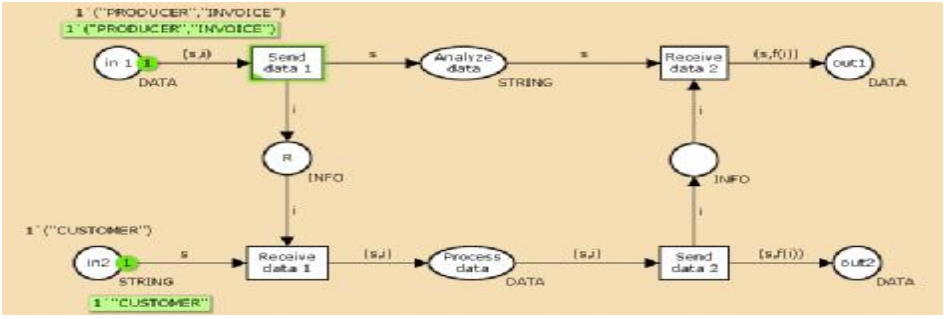
الشكل (5) حل مشكلة ضياع التعريف الكامل للطلب باستخدام محاكي شبكات بتري الملونة

4- الانتظار غير المبرر للعمليات المترابطة:

عرض المشكلة: عمليتان تعملان بشكل مستقل، تنتج العملية 1 بيانات وترسلها إلى العملية 2، تستقبل العملية 2 بيانات العملية 1 وتنتج بيانات بالاعتماد عليها وتعود وترسلها إلى 1، ولتفادي الانتظار غير الضروري كلا العمليتين يجب أن تكون قابلة للتبادل كإرسال واستقبال البيانات بنفس الوقت. وعلى سبيل المثال الزبون يرسل طلب سلعة محددة من التاجر (هنا تتم مقاطعة عمليات التاجر) الذي يقوم بإرسال الفاتورة للزبون وبعد دفع الفاتورة يقوم التاجر بإرسال المادة المطلوبة.

الهدف: السماح بنقل البيانات من مكان لآخر بينما يتجنب منع استكمال إرسال بيانات المرسل باستعمال الناقل غير المتزامن. ففي البيئة الموزعة العديد من العمليات تعمل بشكل مستقل عن بعضها وقد تقوم عملية بمقاطعة أخرى كأن ترسل البيانات إليها، بالرغم من تفاعل العمليات مع بعضها البعض فإن كل منها يعمل بشكل مستقل.

الحل: لكي ترسل البيانات من حلقة إلى أخرى نستخدم ناقلاً غير متزامن placeholder الذي يخزن البيانات الواصلة من المرسل حتى يلتقطها المستقبل. لإنجاز النقل غير المتزامن للبيانات من عملية إلى أخرى في الشكل (6) نستخدم الحلقة R بين حاجز الإرسال والاستقبال للعملية، صيغة البيانات في هذه الحلقة يجب أن تتضمن معرفات معلومة لكل من المرسل والمستقبل، المستقبل سيستخدم معرف البيانات كمرجع عند استقبال البيانات من الحلقة، الحاجز إرسال send يضع البيانات في placeholder بصيغة tuple (s , i) بينما الحاجز استقبال receive يستعمل نموذج ID لاستخراج قيمة عنصر البيانات من placeholder. العملية الأولى ترسل عنصر البيانات إلى العملية الثانية عبر حلقة R. وبعد إرسال البيانات تستمر العملية الأولى بتحضير البيانات للناقل التالي والعملية الثانية تحلل البيانات القادمة وتتابع عملها.



الشكل (6) حل مشكلة الانتظار للعمليات عند مقاطعة بعضها البعض

5- معالجة الكائنات المجمعة:

عرض المشكلة: عندما تكون الحلقات كلها من نفس النمط T ولنفترض أننا بحاجة لأحد العمليات التالية:

حساب عدد العناصر في المجموعة (كحساب عدد المواد الموجودة في المخزن) - إجراء عملية واحدة على كل الكائنات كزيادة السعر مثلاً - اختبار احتواء المجموعة الكاملة لوجود عنصر معين والذي يمتلك خاصية محددة.

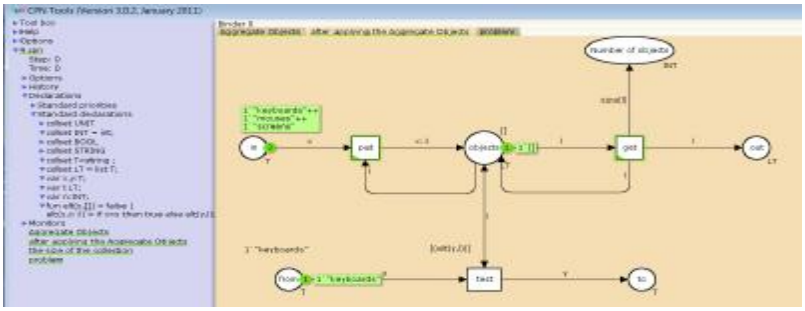
لا يمكن تطبيق أي من هذه العمليات مباشرة يمكننا فقط أن نفحص طلقة واحدة (Token) وهذا عن طريق تعريف تعليمة السهم (inscriptions) أو حارس الحاجز (guards) مما يجعل العمليات صعبة جدا أو مستحيلة في بعض الحالات.

الهدف: السماح لمعالجة كائن معلومات ضمن مجموعة، في العديد من الحالات من الطبيعي تمثيل كائن المعلومات (طلب – رسالة ...) ككيان مفرد وبمعنى آخر يوجد مطابقة بين الكائنات في النظام الحقيقي والطلقات في النموذج، في بعض الأوقات من الضروري تجميع الكائنات AGGREGATE OBJECTS في طلقة واحدة لتفعيل مجموعة من الكائنات التي نستطيع استرجاعها ككيان مفرد.

الحل: لمعالجة مجموعة من الكائنات ككيان entity مفرد، نجمع كل الكائنات في طلقة مفردة من نوع مجموعة وفي الشكل (7) الحلقة objects من النوع T التي تحتوي الكائنات المتعددة يستبدل بنوع مجموعة List Type LT (قائمة-مجموعة) وذلك لنعامل الكائن كمجموعة وفي هذا المثال نختار قائمة نمط المجموعة color LT list T = الأسهم بين الحاجز put والحلقة objects مستبدلة بقوس ثنائي باتجاهين مع التعابير، التعبير يزود كائن للمجموعة يتضمن تابع الإضافة add(x , l) الذي يضيف الكائن x من النوع T إلى القائمة 1، وبالرجوع إلى القائمة 1 كل الكائنات فيها تعنون بشكل آلي، التابع add() يمثل في شبكات بتري الملونة عن طريق المشغل

operator المعطى بالشكل (::) وفي مثالنا نكتب x::l لكي تنجز إضافة الكائن x إلى القائمة l وتنهج سلوك الصفوف LIFO،

بتقديم نمط المجموعة يصبح من الممكن الرجوع لمجموعة من الكائنات ككيان مفرد وأداء عمليات على مجموعة من الكائنات ضمن قائمة كحذف هذه المجموعة أو تطبيق فعل مشترك كزيادة السعر أو ... ، نلاحظ أنه دائما يوجد طلقة واحدة في الحلقة objects تمثل كل الكائنات. حجم القائمة للحلقة Number of objects، يستخدم التابع size(l) لتحديد حجم المجموعة المعرفة في CPN Tools، وبطريقة مشابهة من الممكن تعديل كل الكائنات بفعل وحيد (كزيادة السعر عشرة بالمائة) ولإزالة كل الطلقات (ببساطة نرجع للطلقة مع القيمة الفارغة).



الشكل (7) حل مشكلة إجراء عملية على كائن ضمن مجموعة

6- اختبار احتواء مجموعة لكائن محدد:

عرض المشكلة: نفترض أننا بحاجة لفحص الحلقة Object أنها لا تحمل طلقة محددة القيمة أو الخاصية وهذا غير ممكن وذلك بسبب أن محاكي شبكات بتري يقوم فقط باختبار الطلقات الموجودة في الحلقة بواسطة ربطها مع الحاجز وإطلاقها على التوالي، وعلى سبيل المثال الموظف المسئول عن بيع الماسحات الضوئية عند عدم وجود طلبات له (عدم احتواء المجموعة للطلبات الخاصة به) ينتقل ليقوم ببيع مواد تابعة لقسم آخر.

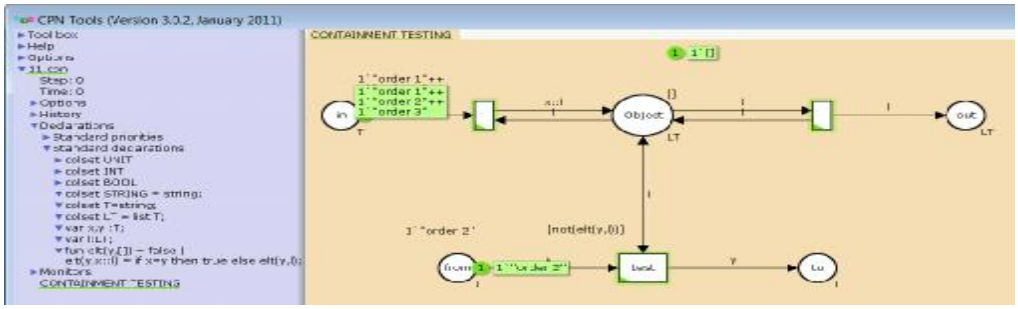
الهدف: هو السماح باختبار وجود كائنات محددة وغير المتوفرة في موقع ما (CONTAINMENT TESTING). في بعض الحالات من الضروري فحص عدد الطلقات أو ألوان الطلقات الموجودة في الحلقة، وبالتالي من الضروري وجود حاجز يتفعل عند وجود حلقة تتضمن طلقة بخواص معينة، على سبيل المثال إجراء تصحيح للبيانات الخاطئة إن وجدت في قواعد البيانات أو اختناق في عملية ما، على أي حال محاكي شبكات بتري لا يستطيع فحص عدم وجود طلقات في حلقة معينة.

الحل: نقوم بجمع الكائنات في حلقة ونصل هذه الحلقة إلى حاجز خارجي الذي سيفحص حالة المجموعة عن طريق سهم الفحص، سهم الفحص سهم مزدوج الذي يستهلك محتوى الحلقة ويرجع المحتويات الغير معدلة للحلقة.

وفي الشكل (8) الكائنات من النمط T مجمعة في قائمة LT=list T القائمة الحالية للكائنات 1 تفحص بواسطة التابع elt() المرفق بحاجز الحارس Test، هذا الحارس يعرف فيما إذا كانت Y عنصر من المجموعة 1، ونعرف التابع elt ضمن التصاريح DECLARATIONS لبرنامج CPN بالشكل:

```
fun elt(y,[]) = false |  elt(y,x::l) = if x=y then true else elt(y,l);
```

هذه الطريقة يمكن أن تستخدم لاختبار وجود كائنات ذات خاصية معينة ضمن الحلقة، إذا كانت الكائنات لا تحقق شرط محدد معرف في الحلقة قد يكون من الضروري حذفها ولفعل هذا يمكن أن نجمع مع مرشح NBSI أو NBSD. وفي الشكل (8) يتم اختبار الحلقة from إذا كانت تحتوي كائنات ذات خاصية معينة باستخدام التابع elt()، عند تطابق الكائنات بين الحلقة from والحلقة Object يكون الحاجز test فعال.



الشكل(8) حل مشكلة اختبار احتواء أو عدم احتواء مجموعة لكائن ما

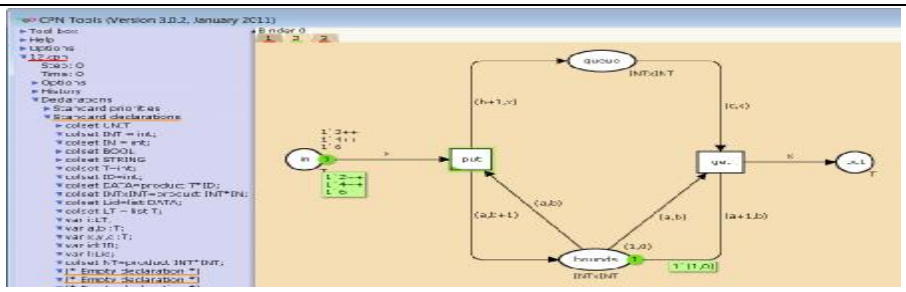
7- **التعامل مع صفوف الانتظار** Queue system:

نظرية خطوط الانتظار "Queueing theory": تقنية رياضية تختص في تحليل خطوط الانتظار كوصول الزبائن إلى البنك /وصول البيانات إلى الحاسب/ وصول طلبات دخل خرج إلى الأنظمة وفي بحثنا سنستخدمها في عمليات خدمات التجارة الالكترونية لتحديد طريقة وصول الزبائن - الأولوية لطلبات الخدمة - عدد وترتيب المخدات في النظام.

عرض المشكلة: نفرض مجموعة من الكائنات بشكل صف انتظار وانه من الضروري إما إضافة كائن للمجموعة أو اختيار كائن لحذفه من الصف وهذا يعتمد على خصائص الكائن مثل (العمر – الوزن....) أو الموقع في الصف (FIFO,LIFO) أو الطابع الزمني. وعلى سبيل المثال متجر يعالج طلبات الزبائن ولتقليل زمن الانتظار فإن الطلب الذي يصل أولا يخدم أولا، وبالاعتماد على الزمن والاضطرارية يتم تخديم الطلبات.

الهدف: هو معالجة الكائنات المصطفة كطابور، في العديد من الأنظمة يوجد عدد متغير من الكائنات تقف في طابور بين خطوتين من العملية وبهذه الطريقة نفترض أن الطابور غير محدود، الكائنات من الضروري أن تكون متوضعة في طابور ومسترجعة طبقا لسياسة محددة، وباستبدال سياسة مع غيرها يصبح من الممكن الحصول على الطلب المرغوب من الكائنات الموجودة.

الحل: نفرض أن العناصر في الصف تتحرك في نظام محدد تماما حيث أن كل كائن في الطابور مميز ككيان منفصل، الكائنات يجب أن تدمج مع رقم يشير إلى موقعها في الطابور، في كل طابور من الضروري متابعة الكائنات عند بداية ونهاية الطابور لكي يسهل إضافة وحذف الكائنات. متغيران يميزان حدود صف الانتظار (المقدمة والذيل) هذه المتغيرات من الضروري أن تجدد عندما العنصر الأول يدخل في الطابور والعنصر الأخير يسترجع منه على التوالي، الحد الأعلى يزداد عندما عنصر جديد يضاف للطابور، الحد الأدنى يزداد عندما عنصر يحذف من الطابور، بواسطة استخدام الحد الأدنى والأعلى يمكن الرجوع إلى العنصر الأول والأخير في الطابور على التوالي، وفي الشكل (9) كل كائن يزاوج مع عدد، المتغيران A,B تتابع العناصر في الطابور وبمعنى آخر المتغير A يخزن موقع العنصر الأول في الطابور بينما المتغير B يشير إلى العنصر الأخير المضاف، عندما عنصر جديد يضاف إلى الطابور القيمة B تزداد بمقدار واحد وهكذا المؤشر ينتقل إلى العنصر الأخير في الطابور. عند الضرورة لحذف عنصر من الطابور، العنصر مع الموقع A (العنصر الأول من الطابور) يزود إلى الحاجز get، وبعد استرجاع العنصر يزداد المتغير A لكي يشير إلى بداية جديدة للطابور.



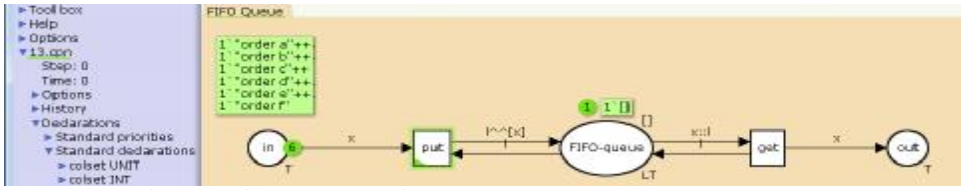
الشكل (9) **حل مشكلة الطلبات المصطفة كخطوط انتظار**

8- **تطبيق صفوف الانتظار باستعمال نمط الواصل أولا يعالج أولا** first-in-first-out queue:

عرض المشكلة: تتراكم طلبات الزبائن عند حلقة الدخل على شكل صفوف انتظار ولتقليل زمن انتظار الزبائن والدقة في العمل وحسن سير الأعمال يجب معالجة الطلبات الواصلة أولا.

الهدف: هو معالجة صفوف انتظار الكائنات في نظام محدد وضمان أن الكائن الواصل أولا يعالج أولا.

الحل: نفترض أن العناصر في الصف تتحرك في طلب الوصول باستعمال صفوف الانتظار first-in-first-out queue، حيث العناصر الجديدة تكون مضافة لنهاية القائمة، والعناصر المصطفة تكون مخزنة والعناصر القديمة محذوفة من رأس القائمة. وفي الشكل (10) كائنات الطابور من النوع T مجمعة في مجموعة من نمط قائمة color LT = list T، الحاجز put يضيف كائن X لنهاية القائمة 1 مع مساعدة تابع السلسلة [x] ^^l، بعد إضافة الكائن x، القائمة المحدثة تعود كدخل للحاجز put، الحاجز get يحذف العنصر الأول من القائمة x::l ويعود ويضع القائمة المحدثة في الحلقة queue وبالتالي الكائنات التي تصل أولا تعالج أولا.



الشكل (10) **معالجة طلبات صفوف الانتظار وفقاً لنمط الواصل أولاً يخدم أولاً**

9- **تحديد الأولوية في صفوف الانتظار** PRIORITY QUEUE:

عرض المشكلة: نفرض مجموعة من الكائنات بشكل صف انتظار وأنه من الضروري إما إضافة كائن للمجموعة أو اختيار كائن لحذفه من الصف وهذا يعتمد على خواص الكائن التي تحدد الأولويات لانجاز المهمة. على سبيل المثال طلبات التجار والشركات المصطفة في الطابور يجب أن تعالج قبل طلبات الزبائن.

الهدف: هو اختيار كائنات في الطابور بحسب الأولوية. عند وجود عدد متغير من الكائنات المصطفة مجمعة في مجموعة ومصطفة بين خطوتين في العملية، يبدأ الطلب بإضافة كائن أو حذف كائن من المجموعة بالاعتماد على الأولوية المرتبطة مع الكائن.

الحل: للسماح بمعالجة الكائنات بحسب الأولوية المعرفة بواسطة خواص محددة للكائن نستخدم صفوف الأولوية Priority Queue التي تنظم دخول الكائنات في الصف، حيث يعالج أولاً الكائن ذو الخاصية الأعلى،

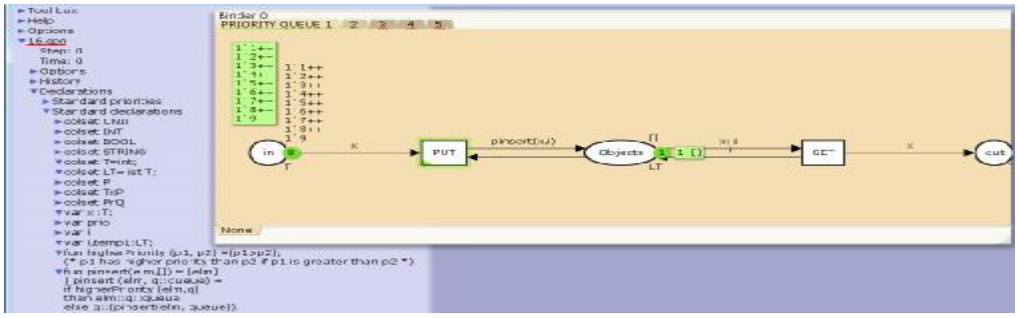
أولوية كائن معتمدة على قيمة الكائن وفي الشكل (11) مجموعة من الكائنات مخزنة في الحلقة Objects، إما خالية أو مشغولة، حيث أن الحلقة Objects من النوع LT وهو نوع القائمة التي تجمع الكائنات من النمط T، "color LT = list T" وفي هذا النموذج الكائنات مصنفة في الطلب بشكل تصاعدي، بمعنى آخر يسحب الكائن ذو الأولوية الأعلى من الصف أولاً. ومن أجل كائنين نوع INT، الكائن ذو الأولوية الأعلى يكون ذو القيمة الأعظم، ويعرف تابع الأولوية higherPriority في محاكي شبكات بتري الملونة بالشكل:

```
fun higherPriority (p1, p2) =(p1>p2);
(* p1 has higher priority than p2 if p1 is greater than p2 *)
```

تخزن الكائنات في الحلقة Objects بشكل تصاعدي Ascending order وبمعنى آخر العنصر الأول يمتلك القيمة الأعلى بينما العنصر الأخير يمتلك القيمة الأصغر، يحدد الحاجز Put الأولوية للكائن الجديد قبل إدخاله insert في الصف باستخدام التابع ()pinsert، مع ضمان بقاء صف الانتظار مصنف بشكل صحيح. نعرف التابع ()pinsert بالشكل:

```
fun pinsert(elm,[]) = [elm]
  | pinsert (elm, q::queue) =
      if higherPriority (elm,q)
      then elm::q::queue
  else q::(pinsert(elm, queue));
```

الحاجز GET يحذف الكائن ذو الأولوية الأعلى (القيمة الأكبر من الطابور) من خلال أخذ العنصر الأول من المجموعة x::l .



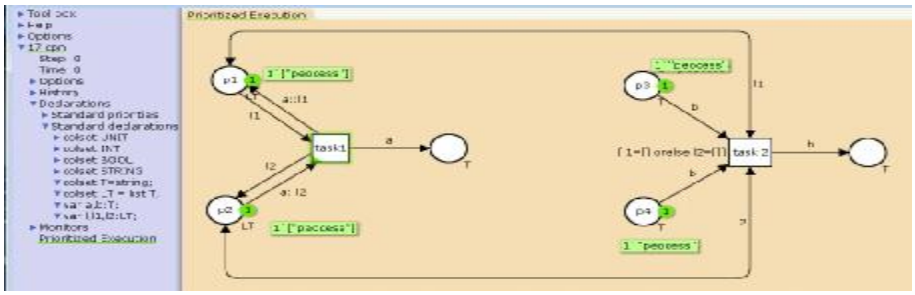
الشكل (11) **معالجة طلبات صفوف الانتظار بحسب الأولوية**

10- **تنفيذ العملية الأفضل** PRIORITIZED EXECUTION:

عرض المشكلة : ليكن لدينا حاجزين A,B يتفعلان عند وجود طلقات في حلقات الدخل ولنفرض أنه وجد طلقات في حلقات دخل الحاجز B ونريد للضرورة أن يتم إطلاق الحاجز A قبل الحاجز B، نستطيع من خلال محاكي شبكات بتري فحص الحلقات إذا كانت تحتوي طلقات ولكننا لا نستطيع فحصها أنها لا تملك طلقات. وعلى سبيل المثال لابد من التحقق من صحة بطاقة الائتمان وتوفر الرصيد للشراء قبل إرسال نتيجة الطلب(السلع).

الهدف : هو تنسيق تنفيذ طلب بمهمتين عندما يكون كلاهما فعالاً، أحدهم يجب أن ينفذ قبل الآخر، تنفيذ الطلب من مهمتين يجب أن ينظم بحسب الأولويات المرتبطة مع هذه المهام، مهمة العملية ذات الأولوية الأعلى تحتاج لأن تنفذ قبل مهمة العملية ذات الأولوية الأدنى.

الحل : لكي ننسق تنفيذ طلب من مهمتين حيث مهمة واحدة لها الأولوية على الأخرى، نختبر غياب الطلقات في حلقة الدخل للمهمة ذات الأولوية الأعلى باستخدام حارس الحاجز التالي [[]=l2 orelse []=l1] وذلك بعد تنفيذ المهمة ذات الأولوية الأدنى. لتحقيق ذلك نجمع الطلقات في الحلقة التي يجب أن تفحص في مجموعة كما هو موصوف في نمط جمع الكائنات Aggregate Objects ونصل سهم لهذه الحلقة من اجل فحص عدد الكائنات في المجموعة وفي الشكل (12) النوع T للحلقات p2, p1 مستبدل مع مجموعة من النوع LT متوافق مع نمط جمع الكائنات، في هذه الحالة نوع القائمة list T= color LT، الطلقات الداخلية للحلقات p2, p1 يجب أن توضع في القائمة الفارغة [] إذا هذه الحلقة لا تحتوي أي كائن، ماعدا ذلك القائمة يجب أن تملأ بالعناصر المطابقة، تمرر المهمة ذات الأولوية الأعلى الفعالة العدد المطلوب من الطلقات من الحلقة (1) وترجع بقية القائمة للحلقة.



الشكل (12) تنفيذ المهمة ذات الأفضلية الأعلى

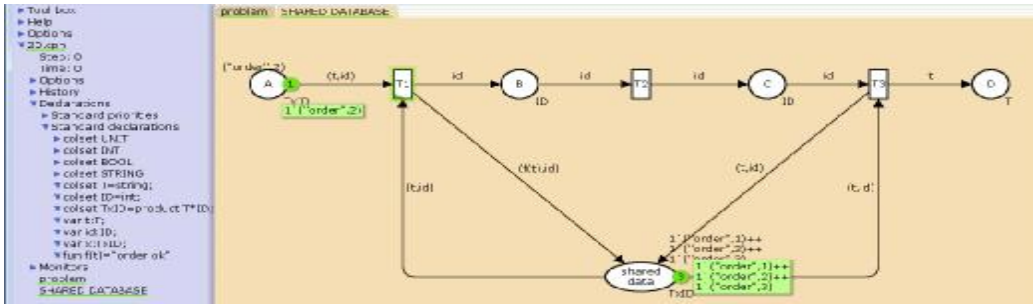
11- الحاجة لوجود قاعدة بيانات المشتركة SHARED DATABASE:

عرض المشكلة: ضمن سلسلة من العلميات لابد من تمرير بعض الخطوات من بداية السلسة إلى نهايتها من دون تأثير على العمليات الداخلية في السلسلة، عند تمرير البيانات مع معرفها من A إلى D وذلك عبر الحواجز T1, T2, T3 مع العلم أن T2 لا يستعملها ولكنه يمررها فإن هذا يزيد من مدة نقل البيانات ويؤثر على الأداء. على سبيل المثال شركتان كل منهما لها عدد من المشاريع الداخلية وأعضاء الشركة الواحدة غير قادرين للوصول إلى المشاريع الداخلية للشركة الأخرى وهي خاصة بموظفي الشركة الواحدة.

الهدف: العمل على تفعيل مخزن مركزي من البيانات المشتركة بواسطة مهمات متعددة مع دعم المستويات المختلفة من منظور البيانات (محلي – مجموعة – عالمي).

الحل: نستعمل حلقة تمثل قاعدة بيانات مشتركة تزود الوصول للحواجز الموصولة معها تكون بمثابة مخزن مركزي للبيانات، لكي نحصل على رؤية شاملة للبيانات في النموذج، كل الحواجز في الشبكة يجب أن توصل لقاعدة البيانات المشتركة، ولكي نحدد رؤية البيانات لمجموعة محددة من الحواجز يجب وصل قاعدة البيانات المشتركة فقط لهذه الحواجز المسموح لها الوصول للبيانات، الرؤية المحلية ممكن الحصول عليها من وصل حاجز فقط لقاعدة البيانات وهكذا عدد الحواجز الموصولة مع قاعدة البيانات المشتركة تحدد رؤية البيانات في النموذج. صيغة البيانات في قاعدة البيانات يجب أن تسمح لكل البيانات المخزنة في قاعدة البيانات لأن تكون مسترجعة بواسطة معرفات مميزة، بدلا من مرور عنصر البيانات عبر سلسلة كاملة من الحواجز التي لا تملك الوصول لقاعدة البيانات المشتركة، يكفي مرور فقط المعرفات المطابقة مع عناصر البيانات، المستوى المرغوب لرؤية البيانات محدد بواسطة عدد الحواجز التي تصل لقواعد البيانات، الحواجز المعرفة موصولة مع قاعدة البيانات المشتركة بواسطة أسهم مزدوجة، السهم الموجه للحاجز يزود قيمة حالية لعنصر البيانات للحاجز المطلوب.

إن تطبيق هذه الطريقة يمكن أن يستعمل لتشارك البيانات بين مجموعة من الحواجز، حيث كل حاجز يُمكّن من الوصول إلى قاعدة البيانات عند الحاجة، تعريف مستوى رؤية البيانات واحد من القرارات الأكثر أهمية، مصمم النموذج يحتاج لأخذ خطوات مبكرة للتطوير لاختيار درجة رؤية البيانات التي تساعد منع وصول البيانات غير المخولة، تقريبا في كل نموذج نحتاج لوجود واحدة من قواعد البيانات العامة التي تخزن المعلومات الدائمة (غالبا غير متغيرة) والعديد من قواعد البيانات المحلية.



الشكل (13) **إضافة قاعدة بيانات مشتركة لسلسلة من العمليات**

12- **إدارة قواعد البيانات** DATABASE MANAGEMENT:

عرض المشكلة: شركتان مستقلتان بحاجة لاسترجاع البيانات من قاعدة البيانات المشتركة للقراءة فقط وبغرض التعديل على التوالي، وطبقا لنمط قاعدة البيانات المشتركة أنه من الضروري وصل حاجز للقراءة فقط وتعديل حلقة قاعدة البيانات المشتركة الذي يجعل البيانات مخزنة في قاعدة البيانات المشتركة والمرئية بين هذه الحواجز، والمشكلة في العلاقة التي يجب أن نستخدمها لأغراض استرجاع البيانات للقراءة والتعديل على التوالي. على سبيل المثال قواعد البيانات المتعلقة بالكتب والمجلات والمقالات والإعلانات المنشورة على الانترنيت أون لاين.

الهدف: هو تحديد وصول البيانات المخزنة في قاعدة بيانات مشتركة من اجل القراءة فقط وأغراض التعديل.

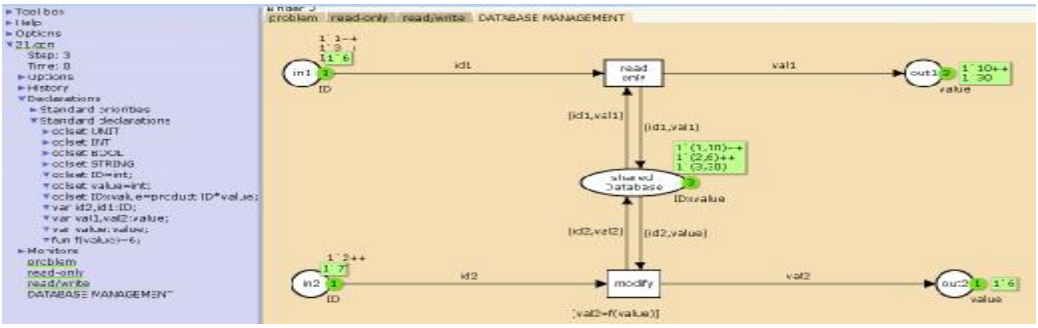
نمط قاعدة البيانات المشتركة يزود حلاً من أجل بيانات مركزية مشتركة بين العديد من الحواجز ويسمح بدعم مستويات مختلفة من رؤية البيانات، عادة قاعدة البيانات المشتركة تستعمل إما مزود بيانات ساكن الذي يتضمن بيانات من أجل أغراض القراءة فقط ويمنع التعديل أو مخزن ديناميكي حيث البيانات من اجل أغراض القراءة والكتابة وللتميز بين النمطين من حيث قواعد البيانات فإن واجهة الوصول يجب أن تكون واضحة تماما.

الحل: كل البيانات المخزنة في قواعد البيانات المشتركة يجب أن تمتلك معرف من أجل أغراض الرجوع، لاسترجاع عنصر البيانات من قاعدة البيانات المشتركة نستخدم معرفات مرتبطة مع عنصر البيانات المخزن كمرجع، من اجل أغراض القراءة فقط قيمة عنصر البيانات مسترجعة وغير معدلة، من أجل أغراض التعديل قيمة عنصر البيانات مسترجعة ويمكن أن تعدل القيمة في قاعدة البيانات.

وفي الشكل (14) حلقة قاعدة البيانات المشتركة موصولة مع الحاجز Modify وتزود قيمة عنصر البيانات مع المعرف المطلوب بالشكل (id1, value). ملاحظة معرف البيانات يعمل كمفتاح لاسترجاع قيمة عنصر البيانات من قاعدة البيانات باستعمال نمط ID Matching، حارس الحاجز Modify يتضمن التابع f(value) الذي يعرف قيمة جديدة معدلة لعنصر البيانات، بعد تقييم الحارس فإن القيمة الجديدة val1 تعود وتوضع في قاعدة البيانات

المشتركة، يزود الحاجز معرف id من أجل الوصول لقيمة عنصر البيانات val1 المخزنة في قواعد البيانات المشتركة من أجل أغراض القراءة فقط (هذه القيمة المسترجعة غير معدلة).

هذا النمط لا يعالج المشاكل التي تظهر خلال الوصول الآني للبيانات من اجل التعديل بواسطة الحواجز المتعددة.



الشكل (14) تطبيق الحل من أجل أغراض القراءة والكتابة

13- استبدال البيانات من قواعد البيانات المركزية بمخزن محلي باستخدام مدير النسخ:

عرض المشكلة: قواعد بيانات مركزية فيها بيانات مخزنة متعلقة بالموظفين، الأقسام المختلفة: قسم المالية - خدمات السكن - مركز التعليم، نحتاج لدخول هذه الأقسام بالوقت نفسه، ونتيجة لقواعد الوصول الخاصة للبيانات لا نستطيع الوصول لبيانات قسم واحد حتى يعطى عن طريق قسم آخر وهذا يؤدي إلى أزمنة انتظار طويلة بالإضافة إلى أن الوصول للبيانات قد يكون محدوداً من أجل كل الأقسام إذا كانت الشبكة الواصلة بين الأقسام وقواعد البيانات غير متاحة مؤقتا وهذا يتوجب جعل البيانات المخزنة على قاعدة بيانات مركزية متاحة محلياً. على سبيل المثال بعض الموظفين الجوالين يحتاجون للدخول إلى بيانات الشركة والإدخال أو التغير وذلك عن طريق أجهزة محمولة وهذا يحتاج برامج موثوقة ومخدم متحكم به عن بعد بوصلة لاسلكية.

الهدف: هو جعل بيانات مخزنة في قواعد البيانات المشتركة متاحة من أجل مواقع أخرى للاستعمال المحلي. في العديد من الحالات البيانات المخزنة في قواعد بيانات مركزية من الضروري الوصول إليها بالوقت نفسه من أماكن مختلفة والتي تكون غالبا مستقلة عن بعضها البعض، كالمواقع التي من الضروري أن تكون قابلة للعمل عندما يكون الاتصال مع قواعد البيانات غير متاح، البيانات المخزنة في قواعد البيانات المركزية يجب أن تبقى محدثة ومتوافقة مع النسخ المختلفة منها والتي هي قيد الاستعمال.

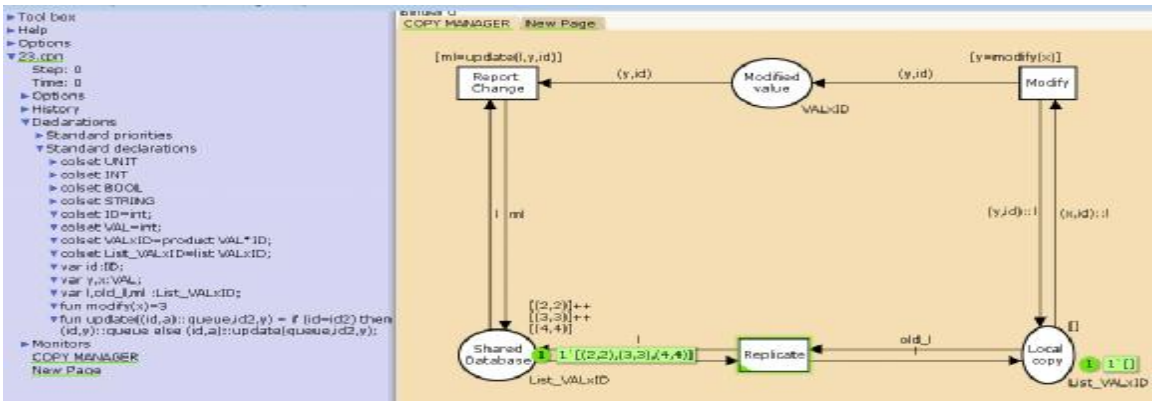
الحل: لجعل البيانات المخزنة في قواعد البيانات المشتركة متاحة لكل المواقع نستعمل copy manager، الذي يكرر البيانات من قواعد البيانات المركزية في مخزن محلي. ولتكرار البيانات المخزنة في قواعد البيانات نقوم بجعل البيانات المخزنة في قواعد البيانات المشتركة مجمعة في مجموعة كنمط Aggregate Objects،

وفي الشكل (15) نضيف حلقة جديدة Local copy حيث تخزن نسخة محلية متاحة من بيانات قاعدة البيانات المركزية، حيث تكون حلقات قواعد البيانات المشتركة والنسخة المحلية هي نفسها حيث قاعدة البيانات المشتركة موصولة مع قواعد البيانات المحلية المنشئة حديثا بواسطة الحاجز Replicate، بعد أن تخزن البيانات في قواعد بيانات محلية Local copy تعدل بواسطة الحاجز Modify، ويتم تحديث القيمة المخزنة في قواعد البيانات المركزية عن طريق الحاجز Report change مع كل حدث جديد بالتوافق مع قواعد البيانات المحلية المحدثة،

حيث يأخذ الحاجز Report Change مجموعة البيانات الكاملة من قواعد البيانات المشتركة ويستبدل قيمة عنصر البيانات مع المعرف id المحدد باستخدام التابع التالي:

```
fun update((id,a)::queue,id2,y) = if (id=id2) then:
(id,y)::queue else (id,a)::update(queue,id2,y)
```

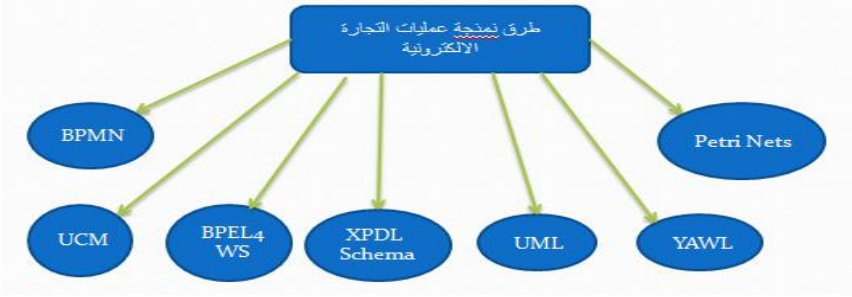
الحلقة Local copy يجب أن تكون متزامنة بانتظام مع حلقة قواعد البيانات المشتركة بواسطة تنفيذ الحاجز Replicate وهذا يضمن أن البيانات في النسخة المحلية Local copy تتزامن مع البيانات المخزنة في قاعدة البيانات المشتركة. هذه الطريقة يمكن أن تطبق لإدارة التناسق بين البيانات المخزنة في حلقة ما وتلك المخزنة في حلقة أخرى بواسطة تكرار البيانات.



الشكل (15) جعل قاعدة البيانات المشتركة متاحة من اجل مواقع أخرى للاستعمال المحلي وبشكل متزامن

مقارنة الطرق التي قامت بنمذجة عمليات التجارة الالكترونية:

إن الأهداف من النمذجة هي: توجيه أفضل للزبون – السيطرة على العمليات بشكل أفضل – حسن توثيق المؤسسات – تحسين الاتصالات بين الأقسام المختلفة – تفادي الأخطاء، والشكل (16) يبين طرق نمذجة عمليات التجارة الإلكترونية مع معاني الاختصارات المعتمدة للطرق، والشكل (17) يجمل المقارنات بين تلك الطرق.



الشكل (16) الطرق التي قامت بنمذجة عمليات التجارة الالكترونية

BPEL4WS: Business Process Execution Language for Web Services by OASIS . (OASIS: Organization for the Advancement of Structured Information Standards)
XPDL: XML Process Definition Language by WfMC . (WfMC: Workflow Management Coalition) ,
YAWL :Yet Another Workflow Language .
UCM : use case maps.
BPMN : Business Process Modelling Notation.
UML :Unified Modeling Language.

نوضح في الشكل (17) مقارنة بين الطرق التي قامت بنمذجة عمليات التجارة الالكترونية وفقاً لمجموعة من النماذج الأساسية المستعملة في نمذجة العمليات التجارية بحسب أنظمة إدارة جريان العمل WFMS [11].

Pattern	CPN	YAWL	UML	BPEL4WS	BPMN	UCM	XPDL
1- Sequence	+	+	+	+	+	+	+
2-Parallel Split	+	+	+	+	+	+	+
3-Synchronization	+	+	+	+	+	+	+
4-Exclusive Choice	+	+	+	+	+	+	+
5-Simple Merge	+	+	+	+	+	+	+
6-Multi Choice	+	+	+	+	+	+	+
7-Synchronizing Merge	+	+	-	+	+	+	+
8-Multi Merge	+	+	+	-	+	+	-
9-Discriminator	+	+	+	-	-	+	-
10-Arbitrary Cycles	+	+	+	-	+	+	+
11-Implicit Termination	+	-	+	+	+	+	+
12-MI without Synchronization	+	+	+	+	+	+	+
13-MI with a Priori Design Time Knowledge	+	+	+	+	+	+	+
14-MI with a Priori Runtime Knowledge	+	+	+	-	+	+	-
15-MI without a Priori Runtime Knowledge	+	+	-	-	-	-	-
16-Deferred Choice	+	+	+	+	+	+	-
17-Interleaved Parallel Routing	+	+	-	-	-	+	-
18-Milestone	+	+	-	-	-	-	-
19-Cancel Activity	+	+	+	+	+	+	-
20-Cancel Case	+	+	+	+	+	+	-

الشكل (17) **جدول مقارنة بين الطرق التي قامت بنمذجة عمليات التجارة الالكترونية**

نصف النماذج المستخدمة في جدول المقارنة أعلاه:

النماذج (5,4,3,2,1) تصف التركيب الأكثر أهمية الذي يمكن من النمذجة تسلسلي، تفرعي، شروط التوجيه.

النماذج (9,8,7,6) تصف البناء المتزامن والمتفرع المتقدم حيث رقم محدد من الفروع يمكن أن يكون مختاراً مع انضمام إلى الطرق المختلفة.

النموذج (10) يصف البناء من اجل الحلقات.

النموذج (11) يصف البناء من اجل الإنهاء بشكل ضمني حيث العمليات تنتهي عندما لا يوجد أي فعل.

النماذج (15,14,13,12) تشير إلى البناء الذي يسمح للحالات المتعددة للنشاطات بأن ينفذ.

النماذج (18,17,16) يهتم بمواصفات السلوك المعتمد على الحالة.

النماذج (20,19) تشير إلى إلغاء الفعاليات.

وبالتالي نجد أن شبكات بتري الملونة CPN في الشكل (17) تدعم جميع النماذج المحددة للمقارنة وهذا يبين تفوقها على الطرق الأخرى التي قامت بنمذجة عمليات التجارة الإلكترونية.

الاستنتاجات والتوصيات:

من خلال هذا البحث نبين فوائد استعمال محاكي شبكات بتري الملونة CPN والذي يعرض بيئة متكاملة من أجل بناء النظام وتحليله وإظهار خواصه. بالمقارنة مع العمليات الجبرية نجد أن شبكات بتري الملونة تزود تمثيل واضح لكل من الحالات والأحداث والنمذجة التي تبنى بشكل متزامن وحقيقي وتضع أرضية من أجل وصف ميثاق التحكم والتزامن الذي يكون متكاملاً مع وصف معالجة البيانات. ومن خلال مقارنة الطرق التي قامت بنمذجة عمليات التجارة الإلكترونية تبين تفوق شبكات بتري الملونة وذلك للأسباب التالية:

1- من اجل أي حالة يمكن الوصول إليها يوجد سلسلة من الطلقات تقود العملية من الحالة البدائية (i) إلى الحالة النهائية (o).

2- الحالة النهائية للنظام (o) هي الحالة الوحيدة القابلة للوصول من الحالة البدائية بوجود طلقة واحدة Token في حلقة الخرج.

3- لا يوجد حالات ميتة (حالات توقف النظام غير المرغوبة).

4- تسمح بتحليل عميق للأنظمة.

5- العديد من أنظمة جريان العمل تستعمل شبكات بتري.

6- تعمل على تحليل الأداء بشكل أكثر وثوقية وأكثر سهولة وتتميز بتحرير وفحص النماذج.

7- تحقق إمكانية تمثيل الشبكة ومحاكاتها ونمذجتها وفق جدول زمني يبين خطوات العمل مع إعطاء تقارير حول النمذجة.

8- نموذج CPN للنظام يصف حالة النظام والأحداث المنجزة التي تسبب تغيرات الحالة، حيث عن طريق النمذجة يصبح من الممكن تحقيق سيناريوهات مختلفة واستكشاف سلوك النظام.

9- أدوات CPN مناسبة للتحرير والنمذجة وتحليل فضاء الحالة وتحليل الأداء. المستخدم يعمل مباشرة على التخطيط التمثيلي GUI لأدوات CPN التي تعتمد على تقنيات التفاعل كأدوات لوحات الألوان والقوائم.

10-النمذجة يمكن أن تستعمل لعدد محدد من طرق تنفيذ النموذج المحلل لتحديد الحالة المرغوبة، وهذا يجعل النمذجة مناسبة من أجل فحص الأخطاء والحصول على الثقة المتزايدة في استقامة النظام.

المراجع:

[1] 2011. < http://ar.wikipedia.org/wiki >.

[2] FENSEL,D ; BUSSLER,C. *The Web Service Modeling Framework WSMF* . Electronic Commerce Research and Applications, USA, 2002, 113–137.

[3] OPEZ-SANZ,M.L; ACUNA,C.J; CUESTA,C.E; MARCOS,E. *Modelling of Service-Oriented Architectures with UML*. Electronic Notes in Theoretical Computer Science 194 , 2008 , 23–37.

[4] TADAO,M, *Petri Nets: Properties, Analysis and Applications*. IEEE VOL 77, NO.4, 1989, 541.

[5] JENSEN,K. *History of Petri Nets*, Aarhus University, 2010, access time 6 Dec.2010. http://cs.au.dk/cpnets/intro/.

[6] KOUNEV,S ; BUCHMANN,A. *SimQPN—A tool and methodology for analyzing queueing Petri net models by means of simulation*. Department of Computer Science, Darmstadt University of Technology, Germany, Performance Evaluation 63, 2006, 364–394 .

[7] VALERO,V; CAMBRONERO,M.E; DIAZ,G; MACIA,H. *A Petri net approach for the design and analysis of Web Services Choreographies*. The Journal of Logic and Algebraic Programming 78, Spain, 2009, 359–380 .

[8] JENSEN,K .*An Introduction to the Theoretica Aspects of Coloured Petri Nets* . Lecture Notes in Computer Science vol. 803, Springer-Verlag ,1994, 230-272.

[9] GHARBI,N; DUTHEILLET,C; LOUALALEN,M. *Colored stochastic Petri nets for modeling and analysis of multiclass retrial systems*. Mathematical and Computer Modelling 49, 2009, 1436-1488.

[10] N.A. MULYAR; W.M.P. van der Aalst. *PATTERNS IN COLORED PETRI NETS*. Department of Technology Management, Eindhoven University of Technology, 2005.

[11] KOSKIMIES,M. *Composing Services in SOA: Workflow Design, Usage and Patterns*. Department of Computer Science, University of Helsinki, 2006.

[12] CPN Tools, version 3.0.2, January 2011.