

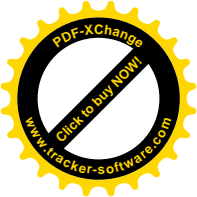
المعاملة الإدارية في "الحكومة الإلكترونية"

الباحث: شادي شماس

ملخص البحث

مع دخول مفهوم الأتمتة الإدارية إلى مجتمعنا، إتجه كثير من المؤسسات والدوائر إلى العمل على أتمتة جميع أعمالها الإدارية، فأحضرت مختصين في عمليات التحليل والتصميم لدراسة متطلبات العمل وإيجاد التصميم المناسب، ومن ثم قام مبرمجون بتحويل هذه المخططات إلى تطبيقات تختصر الكثير من الوقت والجهد في العمل الإداري، ونتخلص عن طريقها من الورقيات المتراكمة على رفوف المكاتب، لذلك كانت الحاجة إلى إيجاد مفهوم جديد للمعاملة، وذلك لتحويلها من شكلها الورقي المعروف الذي يشارك في تسجيله عدد من الأشخاص؛ إلى شكل الكتروني يمتاز بمجموعة من المزايا لمنع التواجد الفيزيائي لطالب المعاملة في المكتب الموافق، وبالتالي تقليل الإزدحام في المكاتب ليصبح بإمكان أي مؤسسة أو شركة أو دائرة تقديم كافة معاملاتها على شبكة الإنترنت كخدمة للعديد من الأطراف.

في هذا البحث أجريت دراسة وافية عن مفهوم المعاملة بشكلها الجديد (النموذج الإلكتروني) المكون من مجموعة من الكيانات المرتبطة بعضها مع بعضها الآخر، والتي تمتلك خصائص متعددة، وبينت مزايا هذا النموذج، و قدمت شرحاً لهذه الكيانات، ووضحت المستويين الأساسيين لهذه الكيانات (مستوى الحزمة-مستوى المعاملة)، وماهي الكيانات التي تتألف منها الحزمة والمعاملة، كما تطرقت لطرق توصيف المعاملات؛ وميزات كل طريقة؛ وكيفية تمثيل كل كيان من خلال استخدام لغة التوصيف القياسية XPD L، وذلك للحصول على ملف خرج يحوي مضمونه كل ما تم إدخاله عن المعاملة، وبالتالي فهو مكافئ للمعاملة.

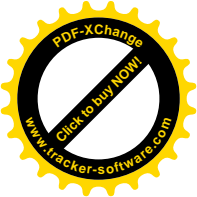


The Administrative Process in “Electronic Government”

ASISTANT: SHADI SHAMMAS

SUMMARY

By the entry of the administrative automation concept to our society, many institutions and departments have automated all their administrative operations. Therefore, they have brought people specialized in design and analysis operations to study work requirements and find the suitable design. Then specialists have changed these designs to applications that save much time and effort in administrative work, and through which we get rid of the cumulate papers on the shelves of offices. So the need has been to find a new concept for dealing. This has been done by changing the dealing from its paper shape to an electronic shape possessing a group of attributes preventing people to be in offices for their dealings. Thereby, congestion in offices will be decreased, and



every institution, firm or department offer all its dealings on the internet as a service for many parts.

In this research, I am presenting an adequate study about dealing concept concerning its new shape (electronic type) which consists of a group of frameworks related to each other and possessing different characteristics. I am also showing the attributes of this type, explaining these frameworks and clarifying the two basic levels of these frameworks (package level, dealing level). I am also referring to the ways of describing these dealings and the attributes of every way, and how to present every framework by using the Standard Description Language XPDL in order to get an output file consisting of everything written about the dealing. In this way this output file will be equivalent to the dealing.

مقدمة البحث :

المعاملة هي مستند يربط بين عدة مكاتب (موجودة في مكان فيزيائي واحد وأحياناً في أماكن مختلفة) من خلال بيانات مطلوبة من هذه المكاتب، ومن هذه الفكرة انطلقت منظمة إدارة تدفق العمل¹ (Workflow Management Coalition) WFMC نحو عملية تحليل لهذه المعاملة من أجل إيجاد تمثيل معياري لبنية المعاملة، بحيث تتم الاستفادة من هذا التمثيل في تسهيل تطبيق عمليات متنوعة كتوصيف المعاملة وإدارتها والتحكم بها والتفاعل معها ، وأن تكون هذه البنية قياسية (Standard) يعتمدها كل المطورين في هذا المجال.

كما أطلقت هذه المنظمة مصطلح تعريف المعاملة (Process Definition) على تمثيل بنية المعاملة، ومن أجل هذا التمثيل تم تأسيس ما يدعى بـ:
(Workflow Process Definition Meta_Data Model)

والذي حول مفهوم المعاملة بشكله المعروف ككلام مكتوب على الورق يشارك في تسجيله عدد من الأشخاص إلى شكل يتألف من مجموعة من الكيانات ترتبط فيما بينها وتمتلك خصائص متعددة.

وكان لا بد من وجود صيغة أو لغة ما للتعبير عن مكونات المعاملة وعلاقتها ببعضها البعض وهذه الصيغة هي ما يسمى بتوصيف المعاملة.

وقد قامت المنظمة Workflow Management Coalition بتقديم صيغة أو لغة بمثابة معيار لتوصيف المعاملات وهي (XML Process Definition XPDL) (Language) [15].

هدف البحث وطريقته:

دراسة المعاملة وفهمها بشكل جيد، ومعرفة مسارها وكل الشروط المتعلقة بإنقالها من مكتب إلى مكتب آخر، ومعرفة نوع المشاركين في كل مرحلة من مراحلها ونوع المعلومات التي يتطلبها كل مشارك والتطبيقات التي تحتاجها. إنني اجري هذا البحث لبيان مدى التطور في تعريف المعاملة، وذلك لتحويلها من الشكل الورقي (الشكل اليدوي) المألوف إلى شكل مؤتمت، أو منظومة مؤتمتة تبدأ بمرحلة طلب

¹ وهي منظمة عالمية لا ربحية تعمل على وضع المعايير الخاصة بمفهوم تدفق العمل

المعاملة من قبل الزبون من خلال الانترنت، وكيفية توصيف المعاملة بشكل رسومي مستخدما بيئة توصيف المعاملات، والتي تمكننا من التعبير عن كل ما يخص المعاملة عن طريق بعض الرسومات، وبعض البيانات المدخلة وذلك من خلال اعتماد لغة قياسية لملف التوصيف.

المناقشة

١- مفهوم المعاملة:

بشكل عام يمكن القول أن المعاملة تتألف من [10],[11],[12]:

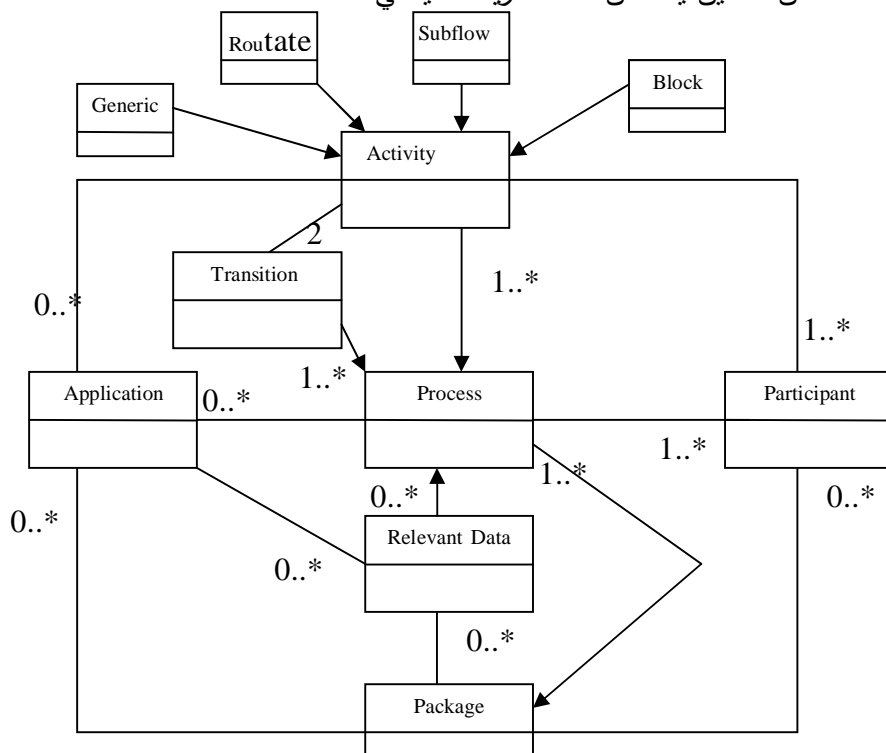
١. مجموعة من المراحل أو الأنشطة (Activities) بالإضافة إلى مجموعة علاقات وشروط تحكم عملية الانتقال فمابينها، ومعلومات خاصة بكل نشاط أو مرحلة.
٢. مجموعة المشاركين (Participants) ويمكن أن يكونوا أشخاص أو آلة أو منظمة أو أي مصدر بيانات يتفاعل مع أي مرحلة من مراحل المعاملة.
٣. مجموعة التطبيقات (Applications) والتي يمكن ربطها مع المعاملة بهدف إنجاز أعمال محددة، مع تحديد البيانات التي يتم تمريرها للتطبيقات كبارامترات دخل والبيانات التي تعطيها كبارامترات خرج.
٤. مجموعة البيانات (Relevant Data) التي تتألف منها المعاملة وهي إما البيانات المطلوبة مع المتفاعلين مع المعاملة، أو البيانات الوسيطة التي تمرر للتطبيقات، أو بيانات تستخدم كشرط للانتقال بين المراحل.
٥. بداية ونهاية المعاملة.

كما أنه في كثير من الأحيان يمكن للمعاملة أن تحتوي على معاملات جزئية أخرى تعمل في إحدى مراحل المعاملة، مع العلم أن أي من هذه المعاملات الجزئية هي معاملة كاملة بحد ذاتها، إلا أن ما تؤديه من عمل يمكن أن يكون جزء من عمل معاملة أخرى.

ويجب التنويه إلى فكرة مهمة ألا وهي مفهوم حزمة تدفق العمل Workflow Package حيث تقابل كل معاملة مستقلة بحد ذاتها بحزمة (يمكن تشبيه مفهوم الحزمة هنا بلغة Java والمعاملات وباقي الكيانات بالصفوف مع فرق أساسي وهو أن الحزمة هنا يمكن أن تحوي أعضاء بيانية بينما ذلك غير ممكن في java)، وهذه الحزمة تحوي:

- المعاملة الأساسية وكل المعاملات الجزئية التي تعمل معها.

- مجموعة من الخواص التي تشترك بها جميع المعاملات المعرفة داخل الحزمة مثل (اسم الكاتب، رقم الإصدار، نوع الكتابة المستخدمة في كتابة الأجزاء النصية...) بحيث تستطيع هذه المعاملات وراثه الخواص من الحزمة الأساسية بدلاً من إعادة تعريفها في كل واحدة على حدا.
 - مجموعة من الكيانات التي تستطيع جميع المعاملات المعرفة داخل الحزمة الوصول إليها في حال احتوت معلومات مشتركة بين هذه المعاملات، فمثلا ليكن لدينا مشارك ما يتفاعل مع أكثر من معاملة ضمن الحزمة، عندها نعرفه ككيان تابع للحزمة بدلا من تعريفه في كل معاملة.
- ويجب أن نعلم انه بوجود مثل هذا التعريف القياسي للمعاملة أصبح من السهل لأي بيئة توصيف للمعاملات أن تستخدم خرجها كدخل لأي محرك مسئول عن إدارة تنفيذ المعاملات طالما أن الاثنين يعتمدان هذا التعريف القياسي.



الشكل (1): مخطط العلاقة بين صفوف الكيانات الأساسية

المصدر: الباحث



٢. الكيانات الخاصة بتعريف المعاملة:

عرف نموذج Meta_Data Model أعلى مستوى من الكيانات الرئيسية المستخدمة في تعريف المعاملات وهي [3] :

Workflow Process Definition.

Workflow Process Activity.

Transition Information.

Workflow Participant Declaration.

Workflow Application Declaration.

Workflow Relevant Data.

Data Repository.

System and Environmental data.

وكما ذكر سابقاً فإنه يوجد مفهومان أساسيان يشتركان في بناء المعاملة وهما الحزمة Package والمعاملة Process، فالحزمة يمكن أن تحتوي على معاملة واحدة أساسية، أو مجموعة معاملات واحدة منها هي المعاملة الأساسية والباقي معاملات جزئية فيها، بالإضافة لذلك تتضمن الحزمة مجموعة من الكيانات والخواص تحوي على معلومات مشتركة بين هذه المعاملات.

ونظراً لهذا التقسيم فإن نموذج Meta_Data Model يضع الكيانات في مستويين أساسيين:

١. مستوى الحزمة (Package Level):

تتألف الحزمة من مجموعة الكيانات التالية [3] :

Workflow Process Definition.

Workflow Participant Declaration.

Workflow Application Declaration.

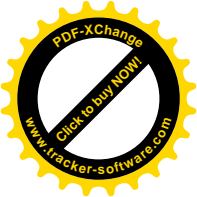
Workflow Relevant Data.

٢. مستوى المعاملة (Process Level):

يمكن القول أن المعاملة [3] تتألف من مجموعة الكيانات التالية :

Workflow Process Activity.

Transition Information.



Workflow Participant Specification

Workflow Application Declaration.

Workflow Relevant Data.

بمقارنة الكيانات في كلا المستويين، يمكن ملاحظة وجود الكثير من الكيانات المشتركة بين المستويين والتي لها المفهوم العام نفسه، ولكن تختلف عن بعضها البعض في مستوى الرؤية [3].

٣. الشرح التفصيلي للكيانات الخاصة بتعريف المعاملة:

أولاً: تعريف المعاملة (Process Definition):

يمثل الكيان الأساسي في تعريف المعاملة، حيث يجمع بين كل الكيانات التي تؤلف المعاملة وتحتوي على معلومات أساسية مشتركة يحتاجها كل كيان من هذه الكيانات ومنها معلومات خاصة بعملية إدارة المعاملة (كتاريخ إنشاء توصيف المعاملة، رقم الإصدار للغة المعاملة،...) ومعلومات تخص تنفيذ المعاملة (مثل أولوية التنفيذ، زمن الانتظار....) [2].

ثانياً: الأنشطة أو المراحل (Workflow Process Activity):

تتألف المعاملة من واحدة أو أكثر من المراحل الجزئية أو الأنشطة Activity التي تمر بها، وبما أن النشاط يعبر عن حدث ما فلا بد من وجود مشارك Participant (سواء كان شخصاً أو منظمة أو مصدر بيانات) يقوم بالتفاعل مع هذا الحدث بحسب طبيعته.

وتوجد عدة أنواع للأنشطة أو المراحل Activities وهي [6],[13]:

١. النشاط العام Generic Activity:

يمثل النشاط يشكله العام، حيث توجد مجموعة من البيانات التي تتم معالجتها، ويمكن تعريف تطبيق أو أكثر لكي يعمل في النشاط [6],[13].

٢. نشاط الطريق Route Activity:

يستخدم هذا النوع من الأنشطة في عمليات فحص الشروط وتوجيه مسار المعاملة، أي لا توجد مهمة حقيقية لهذا النشاط سوى التوجيه، وبالتالي لا داعي لوجود شخص مسئول عن هذا النشاط، كما انه لا يوجد إمكانية لربطه بتطبيق يؤدي مهمة معينة، ولكن على الرغم من ذلك فهو مهم في تحديد مسار المعاملة

لأنه يحقق انتقالات من مرحلة إلى أخرى وفقا لشروط متداخلة من الصعب تحقيقها بدون هذا النشاط [6],[13].

٣. نشاط الكتلة Block Activity:

يقابل هذا النشاط مجموعة من الأنشطة المرتبطة مع بعضها والتي تقوم بإنجاز عمل ما، لذا عند وصول تنفيذ المعاملة إلى هذا النشاط، ينتقل التنفيذ مباشرة إلى هذه المجموعة ليمر على كل نشاط فيها ثم عند الانتهاء من النشاط الأخير يعود التنفيذ إلى هذه الـ Activity من جديد لينتقل إلى المرحلة التي تليها [6],[13].

وبالتالي يمكن عزل كل مجموعة من الأنشطة تقوم مجتمعة بأداء مهمة داخل معاملتنا في مخطط خاص بها، ثم نستعيز عنها بنشاط من النوع Block مرتبط بها ويمثلها في مخطط المعاملة.

٤. نشاط التدفق الجزئي Sub flow Activity:

يقابل هذا النشاط مجموعة من الأمثلة المرتبطة مع بعضها، ولكن الفرق بينها وبين النوع السابق هو أن الأنشطة هنا تشكل مع بعضها معاملة مستقلة بحد ذاتها [6],[13].

بفرض لدينا معاملة أساسية تحتاج في احد مراحلها إلى معاملة أخرى مستقلة (تلعب دور معاملة جزئية بالنسبة للمعاملة الأساسية) فالرابط بين هاتين المعاملتين هو أن الثانية هي نشاط من نوع Sub flow Activity من المعاملة الأولى.

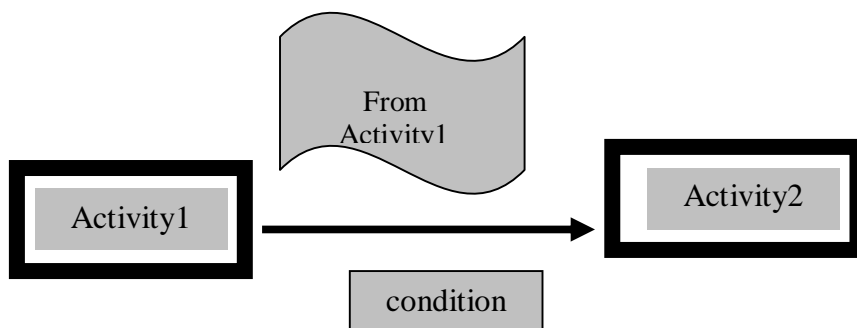
ثالثا: معلومات الانتقال بين الأنشطة (Transition Information):

تتألف المعاملة من مجموعة المراحل أو الأنشطة، وحتى تأخذ المعاملة مفهومها الصحيح لابد من وجود انتقالات بين هذه المراحل لتحديد المسارات التي يمكن للمعاملة أن تسير وفقها، ويمكن للانتقال أن يرتبط بشروط تفرض عدم الانتقال من مرحلة إلى أخرى إلا بعد تحقيقها.

وهذا ضروري في كثير من الحالات، ويعود ذلك إلى طبيعة المعاملة من جهة، ونوع المعلومات التي تحتاجها من جهة أخرى.

يمكن الآن تقديم تعريف جديد للمعاملة على أنها شبكة من الخطوط (الانتقالات Transitions) تربط بين مجموعة من العقد (أنشطة Activities)، ويمثل كل

انتقال بزوج من الأنشطة (From node, To node) تمثل النشاط الذي يخرج منه الانتقال والنشاط الذي يدخل إليه، بالإضافة إلى مجموعة من الشروط التي تحكم عملية الانتقال [9],[5] .



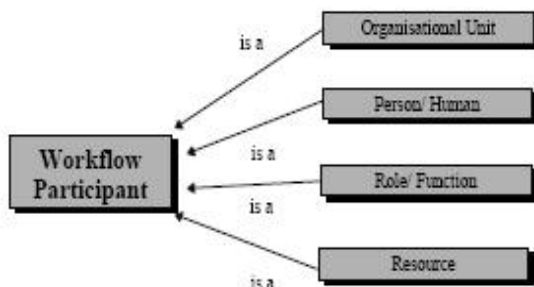
الشكل (1): الانتقال بين الأنشطة

المصدر: الباحث

رابعاً: المشاركون (Workflow Participant Declaration):

يجب أن يوجد من يتفاعل مع كل مرحلة من مراحل المعاملة، إما عن طريق تفعيلها لتبدأ بالعمل أو عن طريق تزويدها بالبيانات أو تعديلها أو غير ذلك. يمكن للمشارك [2] أن يكون:

شخصاً (human) - مصدر بيانات (recourse) - مجموع مصادر (resource set) - وظيفة ما (role) - منظمة (organi zati on) - آلة أو نظام (system) .



الشكل (2): أنواع المشاركين

المصدر: [5]

خامسا: التطبيقات البرمجية (Workflow Application Declarati on):

يتم من خلال هذا الكيان ربط أي مرحلة من مراحل المعاملة (Activity) بتطبيق برمجي موجود مسبقا ليتم الاستفادة منه في إنجاز مهمة ما تحتاجها المعاملة، لتحقيق ذلك لابد من تعريف بيانات (Relevant Data) لتستخدم كبارامترات دخل وخرج لهذا التطبيق [4],[5].

سادسا:بيانات المعاملة (Workflow Relevant Data):

تمثل المعاملات المعرفة في الحزمة أو المعاملة والتي نحتاجها من اجل تمرير البيانات بين الأنشطة أو من اجل الشروط المعرفة عند الانتقال بين نشاطين، أو من اجل تمرير البيانات عبر الـ Subflow Activity إلى المعاملة الجزئية المرتبطة به، ويستطيع المشارك من خلالها التفاعل مع النشاط بإدخال المعلومات أو تعديلها أو مراقبتها [5], [4].

سابعاً:أنواع البيانات(Data Types):

وهي تمثل أنماط المتحولات التي سيتم من خلالها تعريف، وقد تم تقسيمها إلى [2]:

- الأنماط الأساسية (Basic Types):مثل
String, Integer, Boolean, Float.
- الأنماط المصرح عنها (Declared Type).
- الأنماط المعدودة (Enumeration Type).
- المؤشرات الخارجية (External Reference).
- أنماط القوائم (List Type).
- أنماط التسجيلات (Record Type).
- Schema Type.
- Uni on Type.

ثامنا: مخزن مصادر البيانات (Resource Reposi tory):

يشير إلى كل ما يتم تعريفه خارج مستوى المعاملة أو الحزمة، وذلك انطلاقا من حاجتنا لمعطيات ذات بنى معقدة لا تعامل كجزء أساسي في تعريف المعاملات [2].

تاسعا: بيانات البيئة أو النظام (System and Environment Data):

تشير إلى بيانات النظام المستخدم والتي قد نحتاجها أحيانا في مرحلة من مراحل المعاملة، أو في تقديم شروط الانتقال بين المراحل [2] .

٤- طرق التوصيف:

يوجد عدة طرق متنوعة للتوصيف منها:

١. الطريقة النصية: تعتمد هذه الطريقة أسلوبا نصيا صارما نعبر من خلاله عن كيانات المعاملة، إلا أنها تعتبر من الطرق الطويلة والمتعبة إضافة وجود احتمالات عالية لحدوث الأخطاء.
٢. الطريقة الصورية: تعتمد هذه الطريقة على تمثيل المعاملة بلغة صورية، وهنا تنتج مشاكل أخرى بسبب وجود شروط وقواعد صعبة تعتمد على اللغة الصورية في معظم الأحيان.
٣. الطريقة الرسومية: تعتمد هذه الطريقة على تمثيل المعاملة بشكل رسومي، وبهذه الطريقة يستطيع الشخص اخذ فكرة عن المعاملة من نظرة على مخطط المعاملة، وقد استمدت هذه الطريقة سهولتها من الواقع العملي، فلو أننا طلبنا من موظف أو أي شخص عنده معرفة كافية بمسار معاملة أن يشرح لنا هذا المسار فانه بطريقة عفوية سوف يستخدم أسلوبا رسوميا يحوي على مجموعة مكاتب وانتقالات بينها بالشكل التالي:

مكتب ١ ← مكتب ٢ ← مكتب ٣ ← مكتب ٤ ← مكتب ٥ ← مكتب ٦

كما تمتاز هذه الطريقة بسهولة التعديل سواء على مستوى مسار المعاملة أو على مستوى كياناتها، ويجب ألا ننسى هنا أن كل مرحلة من المراحل المرسومة مرتبطة بمجموعة معلومات تخصها (البيانات التي تحتاجها هذه المرحلة، المشارك الذي يتفاعل معها، التطبيق الذي يعمل فيها إن وجد..) بالإضافة إلى معلومات أخرى تخص المعاملة ككل.

٥ - ناتج التوصيف:

يستطيع مهندس المعاملة من خلال الطريقة الرسومية في التوصيف تمثيل المعاملة بكافة معلوماتها على شكل رسومي يكافئ المعاملة، وفي النهاية يكون ناتج مرحلة التوصيف هو ملف يحمل نموذج المعاملة.

إن اعتماد لغة قياسية لملف التوصيف أمر غاية في الأهمية لأنه عندما يكون نموذج المعاملة موصفا بلغة قياسية فإن أي محرك معاملات يدعم هذه اللغة يكون قادرا على فهم نموذج لمعاملة وبالتالي قادرا على إدارة وتنفيذ هذه المعاملة.

٦ - لغة XPDL:

سيتم الآن إلقاء الضوء على بنية اللغة XPDL وذلك يعرض لبعض كيانات المعاملة ممثلة بهذه اللغة حيث يبين الكود البرمجي التالي كيفية تمثيل كل كيان من الكيانات المكونة منها المعاملة وخصائصه والعلاقة بين هذه الكيانات مستفيدين من بعض النماذج المتوفرة والممثلة بهذه اللغة [1] .

يتم تمثيل المشارك (Participant) يحمل الاسم (Part) مثلا والذي من نوع بشري (Human) بالشكل :

```
<Participant id=" process1_Participant1"
Name=" Part" >
<ParticipantType Type=" HUMAN" />
</ Participant>
```

يتم تمثيل التطبيق (Application) يحمل الاسم (App1) وله بارا متر دخل من نمط (Integer) بالشكل:

```
<Application id=" Process1_Application1"
Name=" App1" >
<Formal Parameters>
<Formal Parameter id=" Process1_App1_Formal1"
Index=" 1" Mode=" IN" >
<DataType>
<BasicType Type=" INTEGER" />
```



```
</DataType>  
</ Formal Parameters >  
</ Formal Parameter >  
</ Appl icati on >
```

يتم تمثيل متحول يحمل الاسم (RelevantData1) من نوع (Integer) بالشكل:

```
<DataField id=" Process1_Rel evantData1"  
IsArray=" FALSE" Name=" RelevantData1" >  
<DataType>  
<BasicType Type=" INTEGER" />  
</ DataType >  
</ DataField >
```

يتم تمثيل النشاط أو المرحلة (Activity) تحمل الاسم (Activ1) وتحتوي المعلومات السابقة بالشكل:

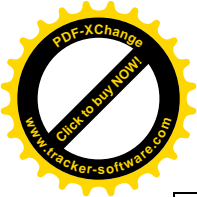
```
<Activity id=" Process1_Activi ty1" Name=" Acti v1" >  
<Implementati on>  
<Tool id=" Process1_APPI ication1"  
Type=" Appl ication" >  
<Actual Parameters>  
<Actual Parameter>  
Process1_Rel evantData1</Actual Parameter>  
<Actual Parameters>  
</ Tool >  
</ Implementati on >  
<Performer> process1_Partici pant1</ Performer >  
<StartMode>  
<Automatic/>  
</ StartMode >
```



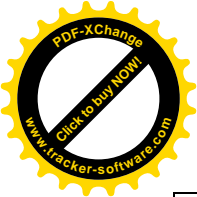
```
<FinishMode>
<Automatic/>
</ FinishMode >
<ExtendedAttributes>
<ExtendedAttribute Name=" ParticipantID"
Value=" process1_Participant1" />
< ExtendedAttribute Name=" Xoffset"
Value=" 320" />
ExtendedAttribute Name=" Yoffset"
Value=" 60" />
</Activity >
```

يتم تمثيل المعاملة التي تحوي النشاط السابق مع كل ما يحويه من معلومات
بالشكل:

```
<WorkflowProcess AccessLevel=" PUBLIC"
id=" Examp_Process1" Name=" Process1"
ProcessHeader DurationUnit=" D"
<Created> 2004-06-10- 15:15:35 </Created>
</ProcessHeader>
<RedefinableHeader PublicationStatus=" UNDER_TEST" />
<DataField id=" Process1_RelevantData1"
IsArray=" FALSE" Name=" RelevantData1" >
<DataType>
<BasicType Type=" INTEGER" />
</ DataType >
</ DataField >
<Participants>
<Participant id=" process1_Participant1"
Name=" Part" >
<ParticipantType Type=" HUMAN" />
</ Participant>
</ Participants>
< Applications >
```



```
<Application id=" Process1_Application1"
Name=" App1" >
<Formal Parameters>
<Formal Parameter id=" Process1_App1_Formal1"
Index=" 1" Mode=" IN" >
<DataType>
<BasicType Type=" INTEGER" />
</DataType>
</ Formal Parameters >
</ Formal Parameter >
</ Application >
</ Applications >
<Activities >
<Activity id=" Process1_Activity1" Name=" Activ1" >
<Implementation>
<Tool id=" Process1_Application1"
Type=" Application" >
<Actual Parameters>
<Actual Parameter>
Process1_Rel evantData1</Actual Parameter>
</ Actual Parameters >
</ Tool >
</ Implementation >
<Performer> process1_Participant1</ Performer >
<StartMode>
<Automatic/>
</ StartMode >
<FinishMode>
<Automatic/>
</ FinishMode >
<ExendedAttributes>
<ExendedAttribute Name=" ParticipantID"
Value=" process1_Participant1" />
```

```
< ExtendedAttribute Name=" Xoffset"
Value=" 320" />
ExtendedAttribute Name=" Yoffset"
Value=" 60" />
</ExtendedAttributes>
</Activity >
</Activities >
</ExtendedAttributes>
<ExtendedAttribute Name=" StartOfWorkflow"
value=" Process1_participant1; Process1_Activity1
120; 60; NOROUTING" />
<ExtendedAttribute Name=" EndOfWorkflow" Value="
Process1_participant1; Process1_Activity1
570; 60; NOROUTING" />
<ExtendedAttribute Name=" ParticipantVisual Order"
value=" Process1_participant1" />
</ExtendedAttributes>
</WorkflowProcess>
```

٧--تطبيق عملي:

سنحاول توضيح الأفكار السابقة من خلال تطبيق مثال عملي يتم من خلاله إدارة المعاملة ضمن جامعة البعث، فلتحويل المعاملة من شكلها الورقي التقليدي إلى معاملة إلكترونية على الويب للطلب والمتابعة يجب القيام بالخطوات التالية:

1. تحليل المعاملة: يتم فيها دراسة المعاملة وفهمها بشكل جيد ومعرفة مسارها وكل الشروط المتعلقة بإنقالها من مكتب إلى مكتب آخر، ومعرفة نوع المشاركين في كل مرحلة من مراحلها، ونوع المعلومات التي يتطلبها كل مشارك والتطبيقات التي تحتاجها.
2. توصيف المعاملة: بعد أن نتضح كل النقاط الخاصة بالمعاملة يقوم مهندس المعاملة بتوصيفها بشكل رسومي مستخدماً بيئة توصيف المعاملات، والتي تمكنه التعبير عن كل ما يخص المعاملة عن طريق بعض الرسومات (مسار المعاملة متضمناً المكاتب التي تمر عليها) وبعض البيانات المدخلة (المعلومات التي يحتاجها كل مكتب، المشاركون

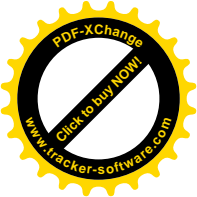
الذين يتفاعلون مع كل مكتب، شروط الإنتقال بين المكاتب بإستخدام لغة مثل (VBSCRIPT).

بعد الانتهاء من تمثيل كل المعلومات السابقة، تعطي بيئة التوصيف ملف XML وفق هيكلية XPDL القياسية كملف خرج يحوي مضمونه كل ماتم إدخاله عن المعاملة، وبالتالي فهو مكافئ للمعاملة.

٣. تحميل ملف التوصيف إلى محرك المعاملات (إدارة العمل): بعد أن أصبح لدينا ملف توصيف يقابل المعاملة، يتم تحميل هذا الملف إلى محرك المعاملات حيث أنه يستطيع أن يفهم ويفسر هذا الملف، وأن يكون قادراً على بناء أغراض مكافئة لهذا التوصيف. كما أنه يتمتع بالقدرة على تنظيم وإدارة عملية نقل المعاملة بين المشاركين، وذلك وفقاً للإنتقالات المحددة للمعاملة في مرحلة التوصيف، كما أنه يقوم بتخزين كل ما يتعلق بالمعاملة في قاعدة البيانات.

٤. التفاعل مع نظام تخاطب من خلال صفحات الويب: يستطيع المدير من خلال نظام التخاطب القيام بجميع عمليات الإدارة والتحكم بالمعاملات، وذلك ابتداءً من تحميل معاملة جديدة (أي تحميل ملف التوصيف XPDL إلى محرك المعاملات)، ومن ثم تفعيل هذه المعاملة، ليتم بعد ذلك تهيئة المصادر الخاصة بها من ربط الموظفين وربط التطبيقات مع المعاملة.

وبإتمام ذلك تصبح المعاملة متاحة على الموقع حيث يستطيع أي شخص طلب نسخة منها من موقع الجامعة، كما يستطيع أيضاً مراقبة هذه النسخة بعد فترة من طلبها لمعرفة أين وصلت وذلك إما عن طريق جهاز الكمبيوتر أو عن طريق الهاتف الخليوي. كما يستطيع أيضاً موظف الجامعة الذي يعمل في إحدى مراحل هذه المعاملة متابعة عمله حيث تظهر في الواجهة الخاصة به جميع النسخ التي تقف عنده حالياً. إن كل العمليات التي تم ذكرها في هذه الخطوة يتم تحقيقها من خلال الواجهات الديناميكية للموقع وذلك بناءً على إرتباط متين بين محرك المعاملات وبين نظام التخاطب (تطبيق الويب).



ويجب التتويه إلى أن مهندس المعاملة هو المسؤول عن إنجاز جميع الوظائف في النظام، فهو لقب إصطلاحي يدل على الشخص الذي يتفاعل مع النظام ويقوم بإنشاء المعاملات، والوظائف الأساسية التي يقوم مهندس المعاملة هي:

١. إنشاء أو حذف أو تعديل حزمة
(Create, Remove, Update Package)
٢. إنشاء أو حذف أو تعديل خواص حزمة
(Create, Remove, Update Package Attribute)
٣. إنشاء أو حذف أو تعديل معاملة
(Create, Remove, Update Process)
٤. إنشاء أو حذف أو تعديل خواص معاملة
(Create, Remove, Update Process Attribute)
٥. إنشاء أو حذف أو تعديل مرحلة أو نشاط
(Create, Remove, Update Activity)
٦. إنشاء أو حذف أو تعديل خواص النشاط
(Create, Remove, Update Activity Attribute)
٧. إنشاء أو حذف أو تعديل إنتقال
(Create, Remove, Update Transition)
٨. إنشاء أو حذف أو تعديل خواص إنتقال
(Create, Remove, Update Transition Attribute)
٩. إنشاء أو حذف أو تعديل مشارك
(Create, Remove, Update Participant)
١٠. إنشاء أو حذف أو تعديل خواص مشارك
(Create, Remove, Update Transition Attribute)
١١. إنشاء أو حذف أو تعديل تطبيق
(Create, Remove, Update Application)
١٢. إنشاء أو حذف أو تعديل خواص تطبيق
(Create, Remove, Update Application Attribute)
١٣. إنشاء أو حذف أو تعديل البيانات

(Create, Remove, Update Relevant)

١٤. إنشاء أو حذف أو تعديل خواص البيانات

(Create, Remove, Update Relevant Attribute)

أما الخطوات الأساسية لإنشاء المعاملة في بيئة التوصيف فهي تكون بالشكل التالي:

١. في البداية يجب إنشاء الحزمة التي ستحتوي المعاملة في داخلها، ففي حال إحتاجت المعاملة إلى معاملة جزئية، يتم تغليف العاملتين داخل الحزمة، ومن الضروري عند إنشاء حزمة إعطاء رقم ID فريد يميزها عن غيرها.
٢. إدخال المعلومات المتعلقة بخواص الحزمة أو تعريف مشاركين أو تطبيقات أو بيانات على مستوى الحزمة، وعادةً يجري تنفيذ هذه الخطوة عند وجود معلومات مشتركة بين أكثر من معاملة داخل الحزمة وذلك لتجنب تكرارها في كل معاملة.
٣. تعريف المعاملة داخل الحزمة وإعطاؤها ID كمعرف لهذه المعاملة لتمييزها، وتحديد نوعية المعاملة، وهي أساسية أم جزئية وذلك بإختيار القيمة (Public) للمعاملة الأساسية أو (Private) للمعاملة الجزئية.
٤. إدخال المعلومات المتعلقة بخواص المعاملة في حال لم يتم إدخالها في مستوى الحزمة.
٥. تعريف الأنشطة التي تتألف منها المعاملة حيث لكل نشاط ID يميزه، كما يجب تحديد المشارك المسؤول عن التفاعل مع هذا النشاط من مجموعة مشاركي المعاملة، كما ويمكن ربط هذا النشاط مع أحد التطبيقات المعرفة داخل الحزمة، وإضافة البيانات التي يحتاجها هذا النشاط، ويجب هنا التمييز بين أنواع الأنشطة التالية:

Block-SubFlow- Route-Generi c

٦. تعريف الإنتقالات بين الأنشطة داخل المعاملة والتي تساهم في رسم مخطط

المعاملة والذي يحتاج بدوره إلى العناصر التالية:

- حدث بداية ونهاية.
- مجموعة الأنشطة.

• مجموعة الإنتقالات.

٧. تعريف المشاركين في هذه المعاملة ولكل منهم ID يميزه، كما أن المعاملة تستطيع الاستفادة من المشاركين المعرفين على مستوى الحزمة.
 ٨. تعريف التطبيقات التي قد تحتاجها المعاملة إما داخل المعاملة، أو على مستوى الحزمة لتستطيع أكثر من معاملة الوصول إليها ويتم بعدها ربط التطبيق بالنشاط.
 ٩. تعريف البيانات التي تستخدم ك
 - بيانات يتفاعل معها المستخدم عند نشاط ما.
 - شروط إنتقال بين الأنشطة.
 - بارامترات للتطبيق أو المعاملات الجزئية.مع تحديد نوع البيانات، سواء أكان بسيطاً أم مركباً.
 ١٠. إجراء عملية حفظ لكل المعلومات السابقة، وبالتالي الحصول على ملف XPDL كخرج يكافئ المعاملة.
- أما محرك المعاملات فهو غرض يحقق واجهة محرك المعاملات وتتحصر وظائفه في:
١. تفسير نموذج المعاملة أي بمعنى آخر يجب على محرك المعاملات أن يفهم ملف توصيف المعاملة، وأن يكون قادراً على بناء أغراض مكافئة لهذا التوصيف.
 ٢. القدرة على التحكم بنسخ المعاملات من حيث الإنشاء والتفعيل والإيقاف المؤقت والإنهاء.
 ٣. القدرة على تنظيم وإدارة عملية نقل المعاملة بين المشاركين وذلك وفقاً للإنتقالات المحددة للمعاملة في مرحلة التوصيف. والقدرة على مراقبة أزمدة تنفيذ الأنشطة وإتخاذ قرار بشأن الأنشطة التي وصل زمن تنفيذها إلى الزمن الحرج.
 ٤. القدرة على تفسير البيانات التي تخص كل نشاط وتميرير هذه البيانات إلى المشارك المناسب.
 ٥. القدرة على التفاعل مع التطبيقات الخارجية المسؤولة عن إنجاز جزء من المعاملة.
 ٦. تقديم خدمة تسجيل الدخول والخروج للموظفين ضمن الوزارة.
 ٧. تقديم خدمات المتابعة والمراقبة بمستوياتها المختلفة، وذلك يتطلب تسجيل جميع الأحداث التي تجري في النظام.

نتائج البحث

إن النتائج التي تم التوصل إليها في هذا البحث تتلخص في النقاط التالية:

١. يعتبر نظام إدارة تدفق العمل الأساس الذي يساعد على تحويل المعاملات من شكل يدوي إلى شكل جديد، فهو منظومة مؤتمتة تبدأ بمرحلة المعاملة من قبل طالبها من خلال الإنترنت لتمر بجميع الموظفين المسؤولين عن عملية إتخاذ القرار بشأنها.
٢. تحتاج أي معاملة لدراسة وتحليل حتى نستطيع تحديد جميع معطياتها وخصائصها.
٣. إن الطريقة الرسومية في التوصيف هي الأسهل والأبسط.
٤. من المفيد أن يكون لنتائج التوصيف (ملف التوصيف) بنية قياسية ك XPDL
٥. عرفت WFMC المعاملة بأنها مجموعة من العناصر تدعى كيانات ترتبط فيما بينها، وتمتلك خصائص متعددة، وتتكامل بعضها مع بعض الآخر لتكون المعاملة بشكلها النهائي.
٦. يوجد مفهوم هام، وهو مفهوم حزمة تدفق العمل Workflow Package، حيث أن كل معاملة مستقلة تقابل بحد ذاتها بحزمة تحوي المعاملة الأساسية، وكل المعاملات الجزئية التي تعمل معها، وتحوي مجموعة من الخواص التي تشترك بها جميع المعاملات داخل الحزمة، ومجموعة من الكيانات التي جميع المعاملات المعرفة داخل الحزمة تستطيع الوصول إليها وذلك عندما تحتوي على معلومات مشتركة فمابينها.
٧. يوجد مفهومان أساسيان يشتركان في بناء المعاملة، وهما الحزمة package والمعاملة process، وذلك حسب نموذج meta_data model.
٨. توجد كيانات مشتركة بين المستويين المشتركان في بناء المعاملة ولها مفهوم عام نفسه، ولكنها تختلف بعضها عن بعض الآخر في مستوى الرؤية.
٩. إن الطريقة الرسومية في التوصيف يستطيع من خلالها مهندس المعاملة تمثيل المعاملة بكافة أشكالها ومعلوماتها على شكل رسومي يكافئ المعاملة، وفي النتيجة نحصل على ملف يحمل نموذج المعاملة.



- [1] Wil M.P. van der Aalst, Patterns and XPDL: A Critical Evaluation of the XML Process Management Eindhoven University of Definition Language, Department of Technology Technology, The Netherlands,web) . w.m.p.v.d.aalst@tm.tue.nl site(
[2] Workflow Management Coalition, 2008-Workflow Process Definition Interface-- XML Process Definition Language. Document Number WFMC-TC-1025, Version 2.1a,USA,217p.
[3] Workflow Management Coalition, 2002-Workflow Process Definition Interface-- XML Process Definition Language. Document Number WFMC-TC-1025, Version 1.0,USA,87p.
[4] Work Group 1,The Workflow Management Coalition Specification,1999- Workflow Management Coalition Interface 1: Process Definition Interchange Process Model. The Workflow Management Coalition, Document Number WFMC TC-1016-P Official P. 1.0 release,USA,
[5] Work Group 1,The Workflow Management Coalition Specification,1999- Workflow Management Coalition Interface 1: Process Definition Interchange Process Model. The Workflow Management Coalition, Document Number WFMC TC-1016-P Version 1.1(Official release),USA,100P.
[6] Workflow Management Coalition,1999- Workflow Management Coalition Terminology & Glossary. Winchester, Document Number WFMC-TC-1011, United Kingdom,65P.
[7] Object Management Group,1999- Workflow Process Definition Request For Proposal. Framingham Corporate Center, OMG Document Draft1, Framingham,11P.
[8] Work Group 1, Workflow Management Coalition,1999- Workflow Management Coalition Interface 1: Process Definition Interchange Q&A and Examples. Workflow Management Coalition, Document Number WfMC TC-1016-X,45P.
[9] Work Group 1,The Workflow Management Coalition Specification,1998- Workflow Management Coalition Interface 1: Process Definition Interchange Process Model. The Workflow Management Coalition, Document Number WFMC TC-1016-P Document Status - 7.04 (Official release),USA,100P.
[10] <http://jawe.ow2.org/doc/1.4/Tutorial/ch01.html>.
[11] <http://jawe.ow2.org/doc/1.4/Tutorial/ch02.html>.
[12] <http://jawe.ow2.org/doc/1.4/Tutorial/index.html>.
[13] <http://jawe.ow2.org/doc/1.4/Tutorial/ch03.html>.
[14] <http://www.together.at/together/prod/twe/index.html>.
[15] <http://www.wfmc.org>.