

Design a Software Engineering CASE Tool (ERToOracle10g) to transform a graphical Entity Relationship model (ER model) to Oracle10g

Nada N. Saleem

Saad Ahmed Th.

Nada_N_S@uomosul.edu.iq

College of Computer Sciences and Mathematics, University of Mosul, Iraq

Received on: 10/7/2011

Accepted on: 16/8/2011

ABSTRACT

CASE Tools play a significant role in the field of software engineering development processes where they are utilized in system modeling and analysis, in system design and in generating automatic software codes based on system design models.

In this research the (ERToOracle10g) tool has been designed and implemented as a CASE Tool. Hence it is used to construct a data modeling using ER Model, then transforming the graphical model to (Oracle 10g) SQL instructions and executing it automatically inside an (Oracle 10g) server. XML has been used to transform graphical ER model to SQL instructions. The graphical ER model includes symbols and shapes that are difficult to analyze or to directly generate SQL instructions. The graphical model is transformed to descriptive text using XML which is easy to analyze and transform to SQL instructions.

Keyword: CASE Tools, ERToOracle10g, Entity Relational model, XML

بناء أداة هندسة البرمجيات بمساعدة الحاسوب (ERToOracle10g) CASE Tool

لتحويل نموذج العلاقات الكيانية (ER Model) إلى شفرة أوراكل 10g

سعد أحمد ذياب

ندى نعمت سليم

كلية علوم الحاسوب والرياضيات، جامعة الموصل

تاريخ قبول البحث: 2011/08/16

تاريخ استلام البحث: 2011/07/10

المخلص

تلعب أدوات هندسة البرمجيات بمساعدة الحاسوب (CASE tool) دورا كبيرا في عمليات تطوير البرمجيات إذ يتم استخدامها على نطاق واسع في نمذجة الأنظمة وتحليلها وكذلك في تصميم الأنظمة وتوليد الشفرات البرمجية تلقائيا بالاعتماد على نماذج تصميم الأنظمة.

تم في هذا البحث تصميم وتمثيل الأداة ERToOracle10g بوصفها أداة من أدوات هندسة البرمجيات بمساعدة الحاسوب إذ تم استخدام هذه الأداة لنمذجة البيانات بالاعتماد على نموذج العلاقات الكيانية (ER Model) ومن ثم تحويل النموذج الرسومي إلى تعليمات SQL الخاصة بـ (Oracle 10g) وتنفيذها بأسلوب آلي داخل مزود خدمة Oracle 10g. تم الاعتماد على لغة التوصيف الموسعة (XML) في عملية تحويل نموذج العلاقات الكيانية الرسومي إلى تعليمات SQL، إذ يضم نموذج العلاقات الكيانية مجموعة من الأشكال والرموز والتي من الصعب تحليلها وتوليد تعليمات SQL بشكل مباشر ولذلك تم تحويل النموذج الرسومي إلى أوصاف على شكل نصوص بالاعتماد على لغة (XML) والتي يسهل تحليلها وتحويلها إلى تعليمات SQL.

الكلمات المفتاحية: أداة هندسة البرمجيات, ERToOracle10g, نموذج العلاقات الكيانية, وسوم لغة التوصيف الموسعة

1- مقدمة:

إن أي أسلوب منظم ومنضبط وقابل للقياس لعمليات تطوير البرمجيات وصيانتها يعد تطبيقاً للهندسة على البرمجيات ودراسة هذه الأساليب وتطبيقها يعد هندسة برمجيات [1]. إن أدوات هندسة البرمجيات بمساعدة

الحاسوب¹ (CASE tools) تغطي مجموعة واسعة لأنواع مختلفة من البرامج التي يتم استخدامها لدعم أنشطة معالجة البرمجيات مثل تحليل المتطلبات، تصميم نماذج النظام وتصحيح الأخطاء والاختبار، وتشمل أيضا مولد الشفرات البرمجية (Code Generator) الذي يولد الشفرات تلقائيا من نموذج النظام وبعض التوجيهات العملية لمهندسي البرمجيات [2]. إن القدرة على توليد الشفرة المصدر (Source Code) تلقائيا من بيئة أدوات هندسة البرمجيات بمساعدة الحاسوب تقلل من الجهد المطلوب لتكوين الشفرة المصدر بصورة يدوية كما أنها تعمل على تسريع عملية تنفيذ البرمجيات [3]، إن أدوات هندسة البرمجيات بمساعدة الحاسوب تساعد مهندس البرمجيات في إنتاج منتجات ذات جودة عالية كما أن توفير الأتمتة يتيح لمستخدم الأدوات إنتاج المزيد من منتجات العمل المخصصة التي لا يستطيع إنتاجها بصورة سهلة أو بالتجربة العملية من دون دعم أدوات هندسة البرمجيات [4]، إن التعقيدات المتزايدة في مشاريع بناء البرمجيات جعلت من استخدام الأدوات الداعمة مسألة حتمية في بناء تلك المشاريع [5] ، كما يوجد مدى واسع من أدوات هندسة البرمجيات بمساعدة الحاسوب التي تدعم مختلف الطرائق ومختلف مراحل دورة حياة تطوير البرمجيات [6-7-8].

قامت العديد من الشركات بإنتاج أدوات هندسة البرمجيات بمساعدة الحاسوب والتي تتلاءم مع أسلوب نموذج العلاقات الكيانية (Entity Relationship Model) والذي قدمه Peter Chen عام 1976 ولا يزال الأسلوب الأكثر انتشارا والذي يستخدم على نطاق واسع في تمثيل نظم قواعد البيانات، حيث يصور هذا الأسلوب متطلبات المعلومات لمؤسسة ما على شكل مجموعة من الكيانات وكل كيان يضم مجموعة من الصفات والتي تشترك بعلاقات مع كيانات أخرى. الإصدارات الحديثة للنموذج تضمنت تحسينات وتعزيزات للرموز المستخدمة في تمثيل الكيانات والعلاقات، وعلى الرغم من أن هذه الرموز غير موحدة بصورة تامة لكن معرفة رموز إصدار معين يجعل من السهل تخمين معنى الرموز المشابهة في الإصدارات الأخرى. إن عملية نمذجة البيانات هي جزء لا يتجزأ من عملية تصميم وتطوير نظام البيانات كما توفر الطرائق والوسائل لوصف متطلبات المعلومات في العالم الحقيقي بطريقة مفهومة لأصحاب المصلحة في المؤسسة ، وتمكن نمذجة البيانات مصمم قواعد البيانات على أن يأخذ هذه البيانات وينفذها كنظام قاعدة بيانات حاسوبي لدعم أعمال المؤسسة. [9]

2- الأعمال السابقة

هناك العديد من أدوات هندسة البرمجيات المعتمدة على نموذج العلاقات الكيانية (ER Model) والتي يتم استخدامها بشكل تفاعلي من قبل مصممي قواعد البيانات لإنشاء مخطط أنموذج العلاقات الكيانية في تطبيقات قواعد البيانات الخاصة بهم، تقوم هذه الأدوات بتكوين المخططات رسوميا ومن ثم تحويلها تلقائيا إلى لغة محددة من لغات نظم إدارة قواعد البيانات العلائقية (Relational Database Management system) ومن هذه الأدوات:

- POWER AMC : وهي منتج من SYBASE.
- Designer 2000 : وهي منتج من ORACLE.
- Windev : وهي منتج من PC-SOFT.

¹ () هي اختصار لـ (Computer Aided Software Engineering tools)

على الرغم من أن هذه الأدوات تقوم بتمثيل نموذج العلاقات الكيانية ولكنها تحتوي على عدة عيوب تتلخص فيما يلي:

1- المستوى المفاهيمي (Conceptual level) والمستوى الفيزيائي (Physical level) غير مفصولين بعناية.

2- التحقق البنوي للمخطط المفاهيمي لم يؤخذ بنظر الاعتبار قبل توليد شفرة SQL المناظرة للمخطط المفاهيمي.

3- تدعم ثلاثة أنواع من القيود وهي (Primary Key, Foreign Key, Not null) ولا توجد إمكانية للتحقق من صحة القيود، ولا توجد إمكانية كشف حالة عدم الانسجام (Conflict) بين القيود عند استخدام أكثر من قيد. [10]

3- لغة التوصيف الموسعة (XML) ونمذجة البيانات:

تم تطوير لغة التوصيف الموسعة XML كصيغة لتمثيل البيانات لاسيما في صفحات الويب، وقيمة عملها الأساسي هو توفير معلومات حول معنى البيانات بنفس الطريقة التي توفرها HTML مما أدى إلى اعتمادها على نطاق واسع بوصفها صيغة لنقل البيانات بين التطبيقات والمؤسسات وتطوير مجموعة متنوعة من الأدوات لتوليد بيانات XML ومعالجة البيانات بتنسيق XML، وقد أدى نجاح XML في هذه الأدوار إلى استخدامها كتسويق لتخزين البيانات. ومهما كان للغة التوصيف (XML) من نقاط قوة ونقاط ضعف أخرى، فأنها لا تزال الحاجة إلى فهم صحيح لمتطلبات البيانات. [11]

4- وصف أداة نمذجة البيانات المقترحة ERToOracle10g CASE Tool

تعتمد الأداة التي تم اقتراحها في هذا البحث على مفاهيم نموذج العلاقات الكيانية إذ أن بالإمكان تمثيل جميع مكونات نموذج العلاقات الكيانية والتي تمثل المستوى المفاهيمي (Conceptual Level) بالاعتماد على متطلبات المعلومات للنظام ثم بناء وتمثيل قواعد البيانات. تشمل الأداة المقترحة الأجزاء التالية:

1-4 الكيانات (Entity Types)

تعد الكيانات العنصر الأساسي في نموذج العلاقات الكيانية لأنها تمثل الجداول بالمستوى المنطقي (Logical Level) عند تصميم قاعدة البيانات، يتم من خلال الأداة إعطاء الاسم المناسب لكل كيان وتحريك الكيان على مساحة العمل بصورة عالية المرنة لكي يتسنى وضع كل كيان بالقرب من الكيانات التي يرتبط معها بعلاقات معينة حسب متطلبات النظام، اسم الكيان سيمثل اسم الجدول في مرحلة توليد الشفرات البرمجية.

2-4 الصفات (Attributes)

بعد تكوين الكيان يتم إعطاؤه مجموعة من الصفات الخاصة به والتي يتم تحديدها في مرحلة تحليل النظام، تدعم الأداة تكوين الصفات للكيانات وإعطاء كل كيان المميزات التي تصف الكيان وتتضمن (اسم الصفة - نوع بيانات الصفة - تحديد الصفة بوصفها مفتاحاً رئيسياً للكيان - طول بيانات الصفة). يتم الاستفادة من (اسم الصفة) في تمثيل الكيان أي في مرحلة التمثيل وكذلك في مرحلة توليد الشفرات البرمجية، لأنها تمثل الحقول

للجداول في قاعدة البيانات بالمستوى المنطقي، أما (نوع بيانات الصفة- تحديد الصفة كمفتاح رئيسي للكيان - طول البيانات) فيتم الاستفادة منها في مرحلة توليد الشفرات البرمجية.

3-4 العلاقات (Relationships)

تدعم الأداة ربط الكيانات مع بعضها البعض من خلال مجموعة من العلاقات والتي تكون على ثلاثة أنواع وهي: (علاقة One To One ، علاقة One To Many ، علاقة Many To Many)، وتدعم الأداة أيضا تحديد نوع العلاقة بالإضافة إلى تحديد أنواع التواجدات لكل كيان والتي تكون على نوعين (تواجد اختياري (Optional: Zero or more occurrence) و(تواجد إجباري (Mandatory: One or more occurrence) والتي تكون ذات فائدة من ناحية التمثيل ومن ناحية توليد الشفرات البرمجية، بعد إضافة أنواع التواجدات على أنواع العلاقات أصبح هنالك ثمانية أنواع من العلاقات مع أنواع التواجدات وكما موضح بالجدول (1):

الجدول (1). أنواع العلاقات والتواجدات وتمثيلها الرسومي في الأداة (ERToOracle10g)

نوع العلاقة	نوع التواجد من جهتي العلاقة	التمثيل الرسومي
One To One	إجباري (Mandatory)	
One To One	اختياري (Optional)	
One To One	إجباري / اختياري	
One To Many	إجباري (Mandatory)	
One To Many	إجباري / اختياري	
Many To Many	إجباري (Mandatory)	
Many To Many	اختياري (Optional)	
Many To Many	إجباري / اختياري	

4-4 العمومية والخصوصية (Generalization/Specialization)

تدعم الأداة تمثيل الكيانات التي تمتلك صفات عامة والتي تشترك معها كيانات أخرى بنفس الصفات بالإضافة إلى صفات خاصة بها والتي تسمى (Supertype/ Subtype) إذ يتم تكوين الكيان الأعم وإعطائه صفاته الخاصة به والتي تعد صفات عامة بالنسبة للكيانات المشتقة التي تمتلك صفات خاصة بها بالإضافة إلى جميع صفات الكيان الأعم.

5-4 محددات التمثيل

لا تسمح ERToOracle10g بتكوين كيائين يحملان نفس الاسم كما لا تسمح بتكوين صفتين تحملان نفس الاسم ضمن نفس الكيان وهذه من محددات تمثيل نموذج العلاقات الكيائية.

6-4 تحويل نموذج العلاقات الكيائية إلى لغة التوصيف الموسعة (XML)

بعد أن يتم تمثيل نموذج العلاقات الكيائية (ER Model) رسومياً باستخدام الأداة ERToOracle10g التي تم تصميمها في هذا البحث تتم التهيئة لتحويل هذا التمثيل إلى شفرة SQL الخاصة بأوراكل 10g وتتلخص عملية التهيئة بمرحلتين، يتم في المرحلة الأولى تحويل جميع مكونات نموذج العلاقات الكيائية من شكلها الرسومي إلى وصف متكامل على شكل نصوص بلغة التوصيف الموسعة (XML)، إذ يتم وصف كل كيان وصفا دقيقا من ناحية (اسم الكيان- صفات الكيان-الصفة التي تعمل مفتاحاً رئيسياً للكيان- نوع بيانات الصفات لكل كيان-طول البيانات لكل صفة-نوع علاقة الكيان مع الكيانات الأخرى). ومن فوائد عملية تحويل التمثيل الرسومي إلى XML هي توليد الشفرة البرمجية بصورة متكاملة خالية من التعارضات والتكرارات والتي تجعل من السهل في المستقبل تطبيق عمليات الهندسة العكسية (Reverse Engineering) على قواعد البيانات وتحويلها إلى نموذج العلاقات الكيائية. والجدول (2) يوضح وسوم لغة التوصيف الموسعة الناتجة عن تحويل نموذج العلاقات الكيائية باستخدام الأداة المصممة (ERToOracle10g):

الجدول (2). وسوم XML وما يماثلها في التمثيل الرسومي

وسوم XML الناتجة عن عملية التحويل	ما يماثلها في التمثيل الرسومي
<EntityName>	تمثل أسماء الكيانات في التمثيل
<AttributeName>	تمثل أسماء الصفات في التمثيل
<AttributeType>	تمثل أنواع بيانات الصفات
<Length>	تمثل أطوال بيانات الصفات
<PrimaryKey>	تمثل الصفات التي تعمل كمفتاح رئيسي
<Reference>	تمثل الكيان الذي يضم المفتاح الرئيسي
<ReferenceTo>	تمثل الكيان الذي يضم المفتاح الثانوي
<ReferenceName>	تمثل اسم العلاقة التي تربط الكيانات
<RelationshipType>	تمثل أنواع العلاقات في التمثيل
<Cardinality>	تمثل أنواع التواجدات في التمثيل

7-4 تحويل لغة التوصيف الموسعة (XML) إلى شفرة Oracle 10g

بعد أن تم في المرحلة الأولى تحويل نموذج العلاقات الكيائية الممثلة بشكلها الرسومي إلى نصوص وسمية بلغة التوصيف الموسعة (XML) يتم في المرحلة الثانية تحويل النصوص من لغة (XML) إلى تعليمات (Oracle 10g) حيث تحتوي الأداة (ERToOracle10g) المصممة في البحث على مفسر نصوص XML والذي يقوم بقراءة وسوم XML وتحويلها إلى شفره SQL الخاصة بـ Oracle 10g وكما يلي:

الخطوة الأولى: قراءة اسم الكيان من الوسوم <EntityName> وجعله اسم للجدول بعد كتابة إيعاز تكوين

الجدول: Create Table <EntityName>

الخطوة الثانية : قراءة أسماء الصفات من الوسم <AttributeName> وأنواع الصفات من الوسم <AttributeType> وطول بيانات الصفات من الوسم <Length> التابعة للوسم <

Create Entity Name > ووضعها بعد إيعاز تكوين الجدول بالشكل التالي: <AttributeName> <AttributeType> <Length> , <EntityName> Table (<AttributeName> <AttributeType> <Length> ,);

الخطوة الثالثة : تحديد المفتاح الرئيسي للجدول من الصفة ذات الوسم <PrimaryKey> وإضافتها إلى نهاية إيعاز تكوين الجدول وكالاتي :

```
CONSTRAINT<EntityName>_<AttributeName>_pk
PRIMARY KEY (<AttributeName>)
```

الخطوة الرابعة : تحديد المفتاح الأجنبي للجدول الذي يضم الوسم <Reference> والذي يمثل الكيان الذي يضم المفتاح الأساسي والوسم <ReferenceTo> الذي يضم المفتاح الثانوي باستخدام إيعاز تعديل الجدول وكالاتي:

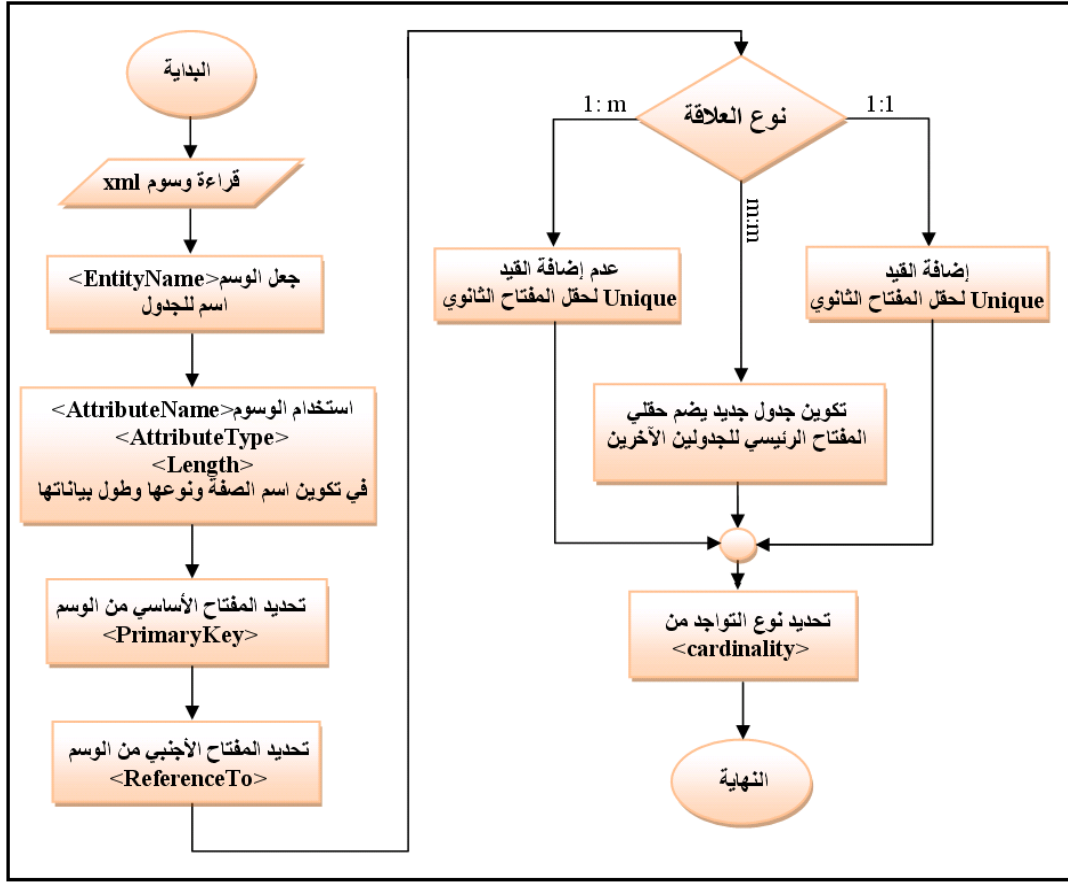
```
ALTER TABLE <EntityName> CONSTRAINT <EntityName>
_<AttributeName>_fk FOREIGN KEY (<AttributeName>) REFERENCES
<EntityName> (<AttributeName>);
```

الخطوة الخامسة : تحديد نوع العلاقة بين الجداول من الوسم <RelationshipType> وإضافة القيود إلى المفتاح الرئيسي والثانوي لكل كيان تربطهما علاقة وإضافة القيود الخاصة بكل علاقة بالاعتماد على الشروط التالية:

- إذا كانت العلاقة من نوع One To One فيتم إضافة القيد (Unique) على حقل المفتاح الثانوي،(وذلك لتحقيق العلاقة حيث لا يسمح القيد (Unique) بتكرار قيمة أكثر من مره في حقل المفتاح الثانوي).
- إذا كانت العلاقة من نوع One To Many لا نضيف القيد (Unique) على حقل المفتاح الثانوي،(وذلك لتحقيق العلاقة والتي تسمح بأكثر من تواجد في حقل المفتاح الثانوي).
- إذا كانت العلاقة من نوع Many To Many فيتم تكوين جدول جديد باستخدام إيعاز Create table واسمه يتكون من اسم العلاقة الذي يؤخذ من الوسم <ReferenceName> الذي يمثل اسم العلاقة بين الجدولين الذين تربطهما العلاقة واخذ حقلي المفتاح الأساسي من كلا الجدولين لتكوين الجدول الجديد وإضافة قيد المفتاح الأجنبي إلى كلا الحقلين في الجدول الجديد باستخدام إيعاز . Alter Table

الخطوة السادسة : تحديد نوع التواجد من الوسم <Cardinality> لكل كيان والتي تكون على نوعين تواجد اختياري وتواجد إجباري وفي حالة كان التواجد من نوع إجباري فيتم إضافة القيد NotNull على حقل المفتاح الثانوي للكيان، أما إذا كان التواجد من نوع اختياري فلا تتم إضافة اي قيد إلى حقل المفتاح الثانوي، (أي أن العلاقة تسمح بوجود قيم فارغة للتواجدات).

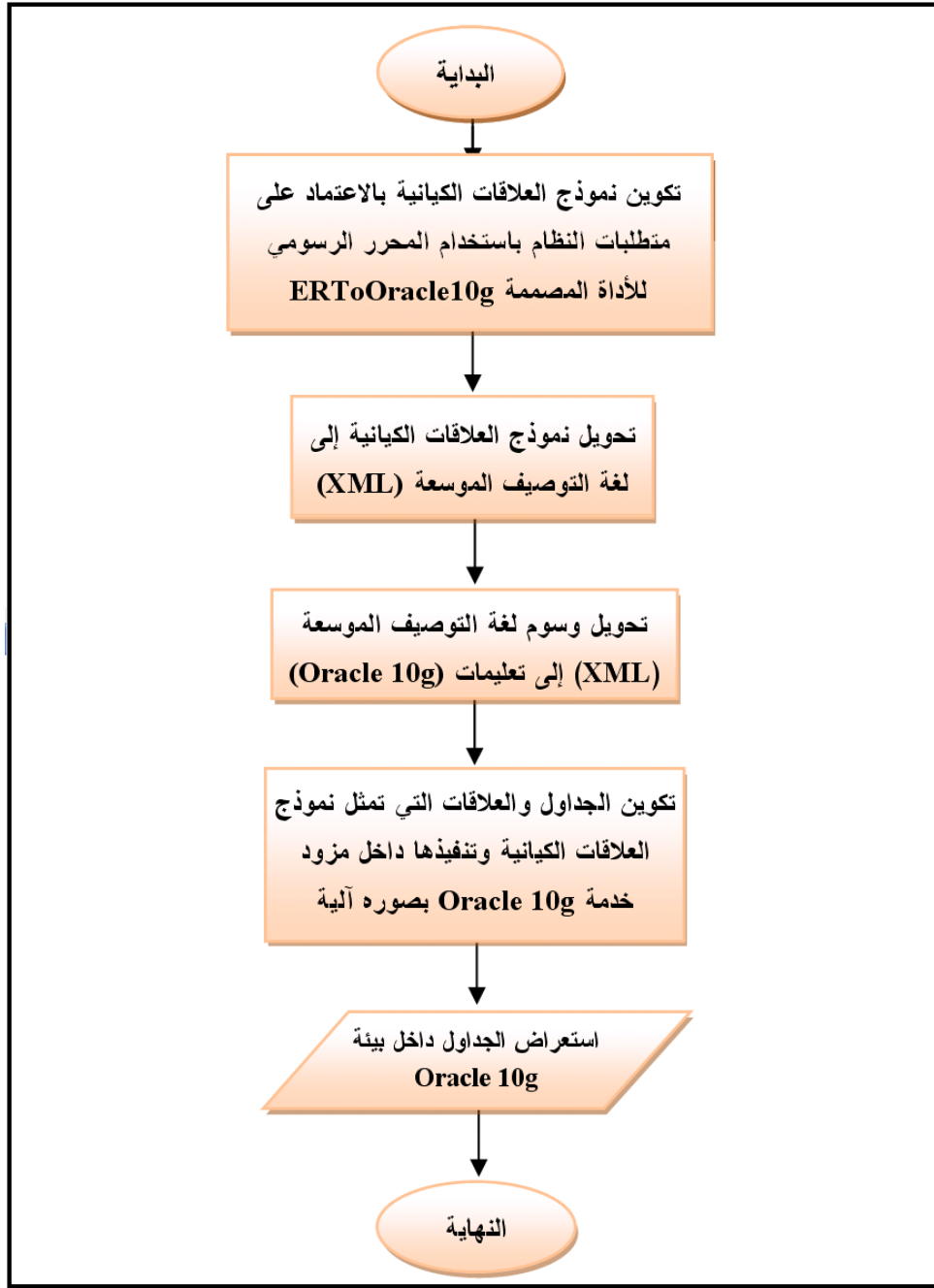
والشكل (1). يوضح المخطط الانسيابي لخوارزمية تحويل وسوم XML إلى شفرة SQL الخاصة بـ Oracle 10g:



الشكل (1). المخطط الانسيابي لخوارزمية تحويل وسوم xml إلى شفرة SQL (Oracle 10g).

8-4 خوارزمية الأداة المصممة ERToOracle10g

مما سبق شرحه عن وصف للأداة ووظائفها يمكن تلخيص الشكل العام للخوارزمية التي تم بناؤها للأداة المصممة في الشكل (2) الذي يوضح المخطط الانسيابي للأداة بشكلها العام:

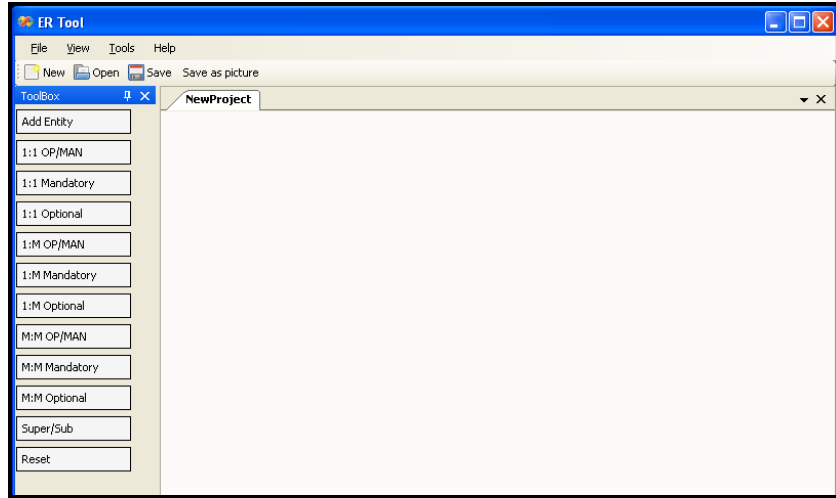


الشكل (2). المخطط الانسيابي للأداة المصممة ERTToOracle10g

9-4 المحرر الرسومي للأداة المصممة ERTToOracle10g

تم تصميم الأداة باستخدام لغة (C#) ضمن Visual Studio 2008.NET حيث يوضح الشكل (3) المحرر الرسومي الذي تم تصميمه لرسم وبناء نموذج العلاقات الكيانية والذي يتكون من مجموعة من القوائم والأدوات التي تسهل للمستخدم عملية التمثيل لنموذج العلاقات الكيانية. القوائم تتكون من (File)، (View)، (Tools) حيث تدعم قائمة (File) عمليات فتح ملف عمل موجود، تكوين ملف عمل جديد، حفظ ملف وغلق ملف، تدعم قائمة (View) عرض مساحة العمل بالأسلوب الذي يناسب المستخدم، تدعم قائمة (Tools) خيارات توليد تعليمات Oracle 10g. تحتوي الأداة على مجموعة من الخيارات الأخرى التي تظهر على

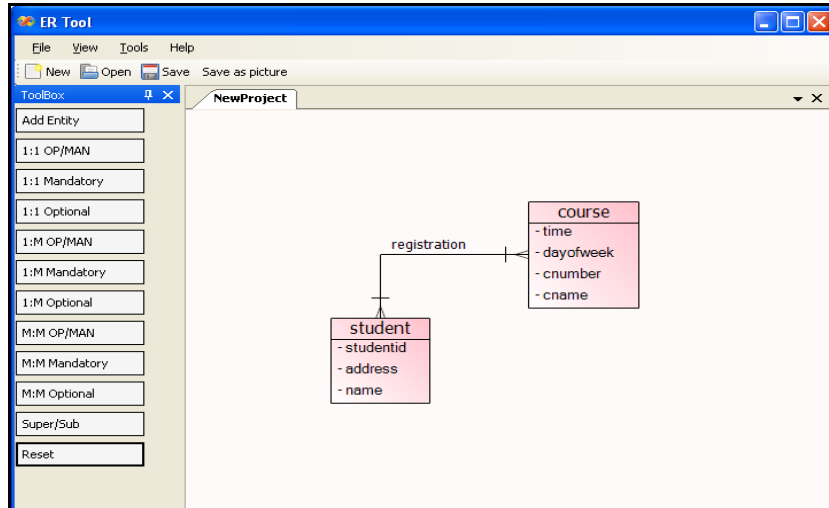
شكل قائمة عمودية يتم من خلالها تكوين الكيانات وإضافة الصفات للكيانات ورسم العلاقات والتواجدات بين الكيانات.



الشكل (3). المحرر الرسومي للأداة المصممة ERToOracle10g

5- نتائج استخدام الأداة (ERToOracle10g) في تمثيل نموذج ER وتوليد الشفرة البرمجية

لغرض توضيح عمل الأداة سيتم الاعتماد على مثال مبسط وهو علاقة الطالب مع المواد الدراسية التي يريد التسجيل من أجل دراستها وهنا سيكون للطالب (Student) مجموعة من الصفات الخاصة به وللمواد الدراسية (Courses) مجموعة من الصفات الخاصة بها وستكون العلاقة المقترحة بينهما هي علاقة Many To Many أي أن بإمكان طالب أو أكثر دراسة مادة دراسية واحدة أو أكثر وسيكون التمثيل للكيانات والعلاقات باستخدام المحرر الرسومي للأداة المصممة والشكل (4) يوضح التمثيل الرسومي للعلاقة :



الشكل (4). التمثيل الرسومي للكيانات والعلاقات باستخدام الأداة المصممة (ERToOracle10g)

وعند تنفيذ اختيار مولد الشفرات الموجود في قائمة Tools سيقوم بتحويل الرسم في الشكل (4) إلى أوصاف بلغة التوصيفات الموسعة XML والموضح في الشكل (5):

```

<?xml version="1.0" standalone="yes"?>
<Field>
<EntityName> student </EntityName>
<AttributeType>number</FieldType>
<AttributeName>studentid</FieldName>
<Length>4</Length>
<PrimaryKey>Primary</PrimaryKey>
</Field>
<Field>
<EntityName> student </EntityName>
<AttributeType>varchar2</FieldType>
<AttributeName>Address</FieldName>
<Length>20</Length>
<PrimaryKey>Not</PrimaryKey>
</Field>
<Field>
<EntityName> student </EntityName>
<AttributeType>varchar2</FieldType>
<AttributeName>Name</FieldName>
<Length>40</Length>
<PrimaryKey>Not</PrimaryKey>
</Field>
<Field>
<EntityName> course</EntityName>
<AttributeType>number</FieldType>
<AttributeName>cnumber</FieldName>
<Length>4</Length>
<PrimaryKey> Primary </PrimaryKey>
</Field>
<Field>
<EntityName> course</EntityName>
<AttributeType>varchar2</FieldType>
<AttributeName>cname</FieldName>
<Length>40</Length>
<PrimaryKey> Not </PrimaryKey>
</Field>
<Field>
<EntityName> course</EntityName>
<AttributeType>varchar2</FieldType>
<AttributeName>dayofweek</FieldName>
<Length>10</Length>
<PrimaryKey> Not </PrimaryKey>
</Field>
<Field>
<EntityName> course</EntityName>
<AttributeType>number</FieldType>
<AttributeName>time</FieldName>
<Length>4</Length>
<PrimaryKey> Not </PrimaryKey>
</Field>
<References>
<Reference>student</Reference>
<ReferenceTo>course</ReferenceTo>
<ReferenceName>registration</ReferenceName>
<RelationshipType>ManyToMany</RelationshipType>
<Cardinality>Mandatory</Cardinality>
</References>

```

الشكل (5). وسوم لغة التوصيف الموسعة (XML) الناتجة عن تحويل التمثيل الرسومي

ومن ثم يعمل مفسر وسوم XML إلى تحويل الوسوم إلى الشفرة البرمجية SQL الخاصة بـ Oracle 10g والتي تتكون من التعليمات الآتية:

```

Create table student (studentid number(4),address varchar2 (20),name
varchar2(40),CONSTRAINT student_studentid_pk PRIMARY KEY (studentid));
Create table course (cnumber number(4),cname varchar2(40),dayofweek varchar2(10),time
number(4),CONSTRAINT course_cnumber_pk PRIMARY KEY(cnumber));
Create table registration (studentid number(4) not null, cnumber number(4) not null);
ALTER TABLE registration ADD CONSTRAINT registration_studentid_fk FOREIGN KEY
(studentid) REFERENCES student (studentid);
ALTER TABLE registration ADD CONSTRAINT registration_cnumber_fk FOREIGN KEY
(cnumber) REFERENCES course (cnumber);

```

الشكل (6). تعليمات SQL الخاصة بـ Oracle 10g الناتجة عن الأداة (ERToOracle10g)

6- مميزات الأداة المصممة (ERToOracle10g)

تحتوي الأداة (ERToOracle10g) التي تم بناؤها في هذا البحث على ميزات عديدة من أهمها:

- 1- الشفرة البرمجية التي يتم توليدها والخاصة بـ Oracle 10g يتم إرسالها عن طريق Oracle Connector والذي تدعمه Visual Studio.Net 2008 لكي تُنفذ داخل مزود خدمة Oracle 10g إذا كان الحاسوب يحتوي على مزود الخدمة وفي حالة عدم وجود مزود الخدمة على الحاسوب يتم توليد الشفرة البرمجية على ملف نصي وبإستطاعة مستخدم الأداة أن يأخذ نسخة من هذا الملف النصي وتنفيذ الشفرة الناتجة على حاسوب آخر يحتوي على مزود خدمة Oracle 10g.
- 2- المرونة العالية في التمثيل إذ أن بالإمكان تحريك جميع الكيانات والعلاقات التي ترتبط بها على مساحة العمل الرسومية من دون حدوث أي تعارضات أو مشاكل في التمثيل.
- 3- إمكانية تخزين مساحة العمل كصوره من نوع (BMP, JPEG, GIF, PNG) والتي تكون ذات فائدة كبيرة في توثيق التصميم على شكل صور.

7- الاستنتاجات

- تم في هذا البحث استنتاج ما يأتي:
- إن أدوات هندسة البرمجيات بمساعدة الحاسوب مهمة جدا في عملية تطوير البرمجيات إذ أنها تعطي نتائج دقيقة وسريعة في نفس الوقت.
 - توليد الشفرات البرمجية تلقائيا يقلل الجهد المبذول من قبل المستخدم في تكوين الشفرات البرمجية يدويا والتي تستغرق وقتاً كبيراً بالإضافة إلى إمكانية حدوث الأخطاء من قبل المستخدم التي من غير الممكن حدوثها عند استخدام أدوات هندسة البرمجيات بمساعدة الحاسوب.
 - لغة التوصيفات الموسعة (XML) مهمة جدا عند بناء أي أداة تحتاج إلى التمثيل الرسومي إذ تعمل على تحويل الأوصاف الرسومية التي لا يمكن التعبير عنها برمجيا إلى أوصاف على شكل نصوص يمكن التعبير عنها برمجيا بعد تحليل تلك النصوص.
 - تتميز الأداة ERToOracle10g بأسلوب تمثيل أنموذج العلاقات الكيانية عن طريق بناء واجهة صعبة البرمجة للمبرمج ولكنها تجعل من السهل لمستخدم الأداة تمثيل أنموذج العلاقات الكيانية ومن ثم تحويل هذا الأنموذج إلى تعليمات SQL الخاصة بأوراكل 10g.

8- التوصيات

- من أهم التوصيات لتطوير عمل الأداة هي :
- توليد تعليمات SQL للغات أخرى مثل (My sql و sql server) لما لها من أهمية في مجال تصميم قواعد البيانات.
 - تنفيذ عمليات الهندسة العكسية (Reverse Engineering) لتحويل تعليمات SQL إلى نموذج العلاقات الكيانية.

المصادر

- [1] Pressman, Roger S., (2010) "Software engineering: a practitioner's approach", McGraw-Hill series in computer science, Boston.
- [2] Ian Sommerville, (2006) "Software Engineering" ,China Machine Press, China.
- [3] T.M. McGinnity, L.P. Maguire , "A CASE-tool oriented approach for embedded systems design " , Elsevier Science, Microprocessors and Microsystems 24 (2001) p.p 493–499.
- [4] Pressman, Roger S., (2001) "Software engineering: a practitioner's approach" , McGraw-Hill series in computer science, Boston.
- [5] Ivan Garcia-Magarino, Rubén Fuentes-Fernandez, Jorge J. Gomez-Sanz," A framework for the definition of metamodels for Computer-Aided Software Engineering tools",Elsevier Science , Information and Software Technology 52 (2010), p.p 422–435.
- [6] Qingping Lin, Chor Ping Low, Jim Mee Ng, Juan Bu, Xiaohua Liu," Multiuser collaborative work in virtual environment based CASE tool" , Elsevier Science, Information and Software Technology 45 (2003) p.p 253–267.
- [7] R.J. Norman, M. Chen, Working together to integrate CASE, IEEE Software 9 (3) (1992) 13–16.
- [8] Q. Lin, C.P. Low, J.M. Ng, A Virtual Environment Based System for Assisting Object-Oriented Software Development, Final Report for AcRF project 20/98, Nanyang Technological University of Singapore,2000.
- [9] Ponniah, Paulraj., (2007), "Data modeling fundamentals: a practical guide for IT professionals", John Wiley & Sons, Inc., Hoboken, New Jersey.
- [10] Faouzi Boufares, Naoufel Kraiem," A new tool to analyze ER-schemas", IEEE Software, (2001) ,p.p 302-307.
- [11] Graeme C. Simsion, Graham C. Witt, (2005), "Data Modeling Essentials", Elsevier Inc., San Francisco.