دار المنظومة
DAR ALMANDUMAH
الرواد في قواعد المعلومات العربية

العنوان:	استخدام النظم الخبيرة في أعمال الإرشاد الأكاديمي : النظم الخبيرة والإرشاد الأكاديمي بواسطة الحاسب
المصدر:	على الطريق
الناشر:	المنظمة العربية للمسؤولين عن القبول والتسجيل في الجامعات بالدول العربية
المؤلف الرئيسي:	العبد، علي ابراهيم
المجلد/العدد:	ع 14
محكمة:	نعم
التاريخ الميلادي:	1996
الصفحات:	1 - 37
رقم MD:	776832
نوع المحتوى:	بحوث ومقالات
قواعد المعلومات:	EduSearch
مواضيع:	الذكاء الاصطناعي، الحاسبات الالكترونية، التعليم الجامعي، الارشاد الأكاديمي، النظم الخبيرة، تكنولوجيا التعليم، البرمجة
رابط:	http://search.mandumah.com/Record/776832

© 2021 دار المنظومة. جميع الحقوق محفوظة.
هذه المادة متاحة بناء على الإتفاق الموقع مع أصحاب حقوق النشر، علما أن جميع حقوق النشر محفوظة. يمكنك تحميل أو طباعة هذه المادة للاستخدام الشخصي فقط، ويمنع النسخ أو التحويل أو النشر عبر أي وسيلة (مثل مواقع الانترنت أو البريد الالكتروني) دون تصريح خطي من أصحاب حقوق النشر أو دار المنظومة.

إستخدام النظم الخبيرة في أعمال الإرشاد الأكاديمى

الدكتور / علـــى العـــبد

قسم الحاسب الآلى

الاكاديمية العربية للعلوم والتكنولوجيا

النظم الخبيرة والإرشاد الأكاديمي بواسطة الحاسب

د. علي ابراهيم العبد

قسم هندسة الحاسب ـ كلية الهندسة والتكنولوجيا

الأكاديمية العربية للعلوم و التكنولوجيا

ص. ب. 1029 الاسكندرية - ج. م. ع. - ت. 5601785 - فاكس 5602144 (03)

◆ نبذة عن موضوع البحث

يهدف البحث الى إبراز أهمية إستخدام تقنيات الحاسب الحديثة سواء من ناحية المعدات (Equipment) أو البرمجيات (Software) في مجال الإرشاد الأكاديمي للجامعات والمعاهد التي تستخدم نظام الساعات المعتمدة كأسلوب للنظام التعليمي .

ويتعرض البحث لمشكلة الإرشاد الأكاديمي الأمثل للطالب في حالات الأعداد الكبيرة مع قلة عدد المرشدين الأكاديمين ذوي الخبرات المتراكمة في هذا المجال ، خاصة مع بداية الفصول الدراسية وفي ظل الإلتزام بجدول زمني محدد لإتمام عملية تسجيل الطلاب.

ويركز مجال البحث على إستخدام النظم الخبيرة (Expert Systems) كأسلوب برمجة متقدم مع إستخدام الوسائط المتعددة (Multimedia) كأسلوب متطور مع مستخدمي الحاسبات ، لإستخدام الحاسب كبديل للخبير البشري في مجال الإرشاد الأكاديمي في حالات عدم تواجده أو كمساعد للمرشد الأكاديمي المبتدئ .

ويمثل الطلاب بالأكاديمية العربية للعلوم و التكنولوجيا مجتمع البحث. ولقد تم إختيار طلاب قسم هندسة الحاسب بكلية الهندسة والتكنولوجيا عينة ممثلة لمجتمع البحث حيث تم إخذهم في الإعتبار لإختبار صحة البرنامج وفاعليته والتأكد من صلاحيته في هذا المجال .

ويستخدم الباحث نظام الخبرة المعتمد على الأطر (Frame Based Expert System) كهيكل بناء لنظام الأرشاد الأكاديمي حيث تستخدم الأطر (Frames) لتمثيل المعرفة (Knowledge Representation) في النظام . كذلك إستخدم الباحث حزمة البرمجيات Level 5 Object كوسيلة متطورة لبناء النظم الخبيرة على الحاسب ، حيث توفر بيئة فعالة لتحديث قواعد البيانات والمعرفة المرتبطة بالنظام وكذلك سهولة تصميم وسيط للمستخدم يحتوي على الأيقونات (Icons) والتأثيرات الصوتية (Sound Effects) اللازمة كأسلوب متطور للتعامل مع النظام لضبط الأداء .

وتتكون الورقة البحثية من جزئين أساسين ، الجزء الأول "النظم الخبيرة" ويلقي الضوء على النظم الخبيرة كأسلوب متطور في معالجة المعلومات والتعريفات الأساسية في هذا المجال ، كما يوضح المكونات الأساسية للنظام الخبير ويتعرض لإختيار الأسلوب الملائم لتمثيل المعرفة وطرق الإستدلال في النظم الخبيرة . أما الجزء الثاني فيركز

على "تصميم النظام الخبير" الخاص بعملية الإرشاد الأكاديمي في الأكاديمية العربية للعلوم و التكنولوجيا حيث يتم التركيز على وصف الأطر (Frames) والطرق (Methods) المرتبطة بها والقواعد (Rules) الخاصة بعملية الإرشاد .

♦ النظم الخبيرة

• مقدمة

فى كتابه "بنية الثورات العلمية The structure of scientific Revolution " وصف "توماس كون Thomas Kuhn " تطور العلوم بأنه عملية دورية تتعاقب فيها مراحل إعتيادية مع فترات ثورية. فى المراحل الإعتيادية يكون هناك شبه إجماع بين علماء مجال معين حول المسائل المهمة التى يجب دراستها فى هذا المجال، وأنواع الحلول والتفسيرات المراد التوصل إليها. ومع مرور الوقت تظهر بعض المشاكل فى النظريات أو النماذج المقترحة تؤدى إلى تزايد عدد الباحثين الغير راضين عن النظرية السائدة أو النموذج الحالى. بعد ذلك تأتى الفترة الثورية حيث يتم وضع الأسس لنظرية جديدة توفر فهما أفضل للأسئلة الغير مجاب عليها فى النظرية السابقة. وتصبح النظرية الجديدة مرحلة إعتيادية فى سلسلة تطور العلوم وهكذا.

بتطبيق وصف توماس كون على مجال برمجيات الحاسب نجد أن النظرية التقليدية فى برمجة الحاسب تبنى مفاهيميا على أساس المعادلة الآتية

Data + Algorithm = Application Program

بيانات + خوارزم = برنامج تطبيقى

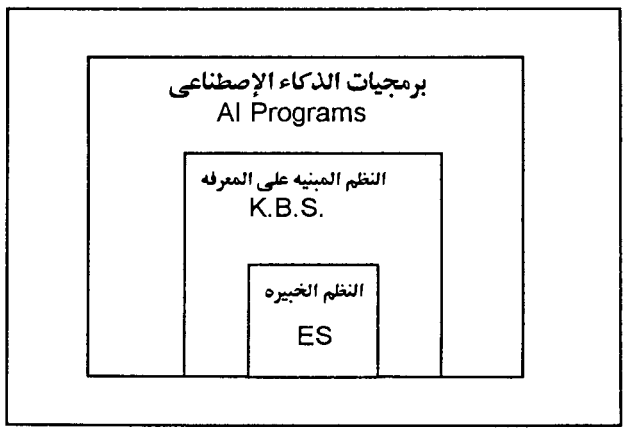
وهى تصلح لحل المشاكل التى يمكن إيجاد خطوات محددة لها لنحصل على المعلومات اللازمة لإتخاذ القرار. ومع التطور الكبير الذى بدأ يأخذ دورته المتصاعدة فى عالم الحاسبات ظهر دور بارز لما يطلق عليه "المعرفة Knowledge " وإتضح وجود فرق شاسع بين المعلومات والمعارف فمعظم المعارف إستنتاجية أكثر منها حسابية أو إجرائية. وبالتالى ظهر قصور النظرية التقليدية فى البرمجة حيث أظهرت عدم القدرة على تمثيل وهيكلة المعرفة ذلك المزيج المكون من المعلومات والخبرة والحكمة البشرية. ومن هنا بدأ العلماء فى تصميم أدوات وطرائق لتمثيل المعرفة وظهرت نظرية جديدة هى نظرية البرمجة المبنية على القواعد Rule-based Programming والتى تقوم مفاهيميا على أساس المعادلة الآتية

Knowledge + Inference = Expert System

معرفة + إستدلال = نظام خبير

وهكذا خرج إلى الوجود مفهوم قاعدة المعرفه (Knowledge Base) لتصبح وعاءاً لذخيرة المعارف وشبكة العلاقات والتفاعلات التى تربط بينها كبديل أرقى لقاعدة البيانات (Data Base) فى الأنظمة التقليدية. ونتيجة لترسخ الإعتقاد بأن المعرفة يمكن أن تخضع للسيطرة الهندسية ظهر فرع جديد من العلوم يطلق

عليه "هندسة المعرفة Knowledge Engineering " ليخدم أغراضاً يحتاجها الإنسان بدءاً من المسائل الطبية ووصولاً إلى تصميم منتج ومروراً بإتخاذ القرارات الإدارية والعسكرية وفى الزراعة والتعليم والتعدين ومعارف الحياة الأخرى. وذلك من خلال تصميم النظم الخبيرة التى بزغت كجزء من علوم الذكاء الإصطناعى كما يتضح من شكل (1)



شكل (1)

• تعريفـــات

يجدر بنا فى البداية أن نضع بعض التعريفات الواردة فى مجال النظم الخبيرة تيسيراً للقارىء على فهم الجوانب الخاصة ببناء هذه الأنظمة وتطويرها.

النظام الخبير Expert System

يعرف النظام الخبير بأنه تطبيق على الحاسب لحل مشكلة تتطلب خبرة إنسانية ويتم ذلك عن طريق محاكاة عملية الإستدلال البشرية.

وتعرف جمعية الحاسب البريطانية النظام الخبير بأنه تجسيد لنظام يعتمد على المعرفة – التى تمثل مهارة الخبير الإنسانى – بالشكل الذى يمكن عن طريقه أن يقدم هذا النظام نصيحة أو توصية تتسم بالذكاء لأحد المواقف أو المشاكل.

خبير المجال Domain Expert

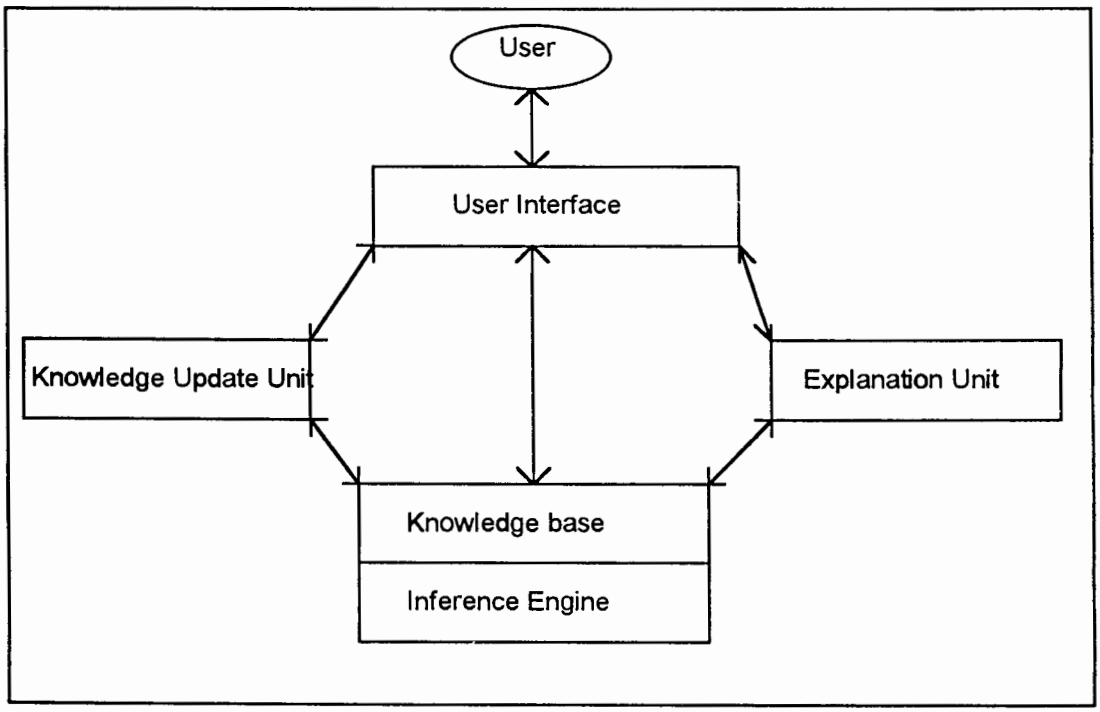
هو الشخص الذى تتوافر لديه المعرفة الواسعة والخبرة اللازمة فى مجال معين والذى يمكن من خلاله وضع الحقائق والقواعد الأساسية فى قاعدة المعرفة.

مهندس المعرفة Knowledge Engineer

هو الشخص الذى يقوم بصياغة المعرفة التى تم الحصول عليها من خبير المجال لبناء النظام الخبير فى صوره تناسب العمل على الحاسب.

• المكونات الأساسية للنظام الخبير

يوضح شكل (2) الأجزاء الخمسة الرئيسية التى تدخل فى بناء النظام الخبير وهى :



شكل (2)

(1) قاعدة المعرفة Knowledge Base

وتحتوى على مجموعة المعارف والخبرة اللازمة لحل مشكلة أو مجموعة من المشاكل فى المجال الذى تم إنشاء النظام الخبير من أجله.

(2) آلة الإستدلال Inference Engine

بواسطتها يتم إنتقاء المعرفة المناسبة لحل المشكلة تحت الدراسة وإستنباط المسببات التى تؤدى إلى هذا الحل. والجدير بالذكر أنه يمكن تصميم آلة الإستدلال عامة الأغراض تستخدم مع أكثر من قاعدة معرفة فى مجالات وأنشطة متعددة لإنشاء أنظمة خبيرة فى هذه المجالات وذلك لأن تحديد هوية النظام الخبير تعتمد أساساً على نوعية المعرفة والخبرة الموجودة بقاعدة المعرفة الخاصة بهذا النظام.

(3) وحدة الشرح Explanation unit

بواسطتها يقوم النظام الخبير بتفسير كيفية وصوله إلى النتائج والحلول والتوصيات الخاصة بالمشكلة المعروضة عليه. وهذه القدرة على التعليل هى إحدى الخصائص الهامة للنظام الخبير والتى تجعله إلى حد كبير يحاكى سلوك الخبير البشرى.

(4) وسيط المستخدم User Interface

بواسطته يتم الإتصال والتفاعل بين الإنسان المستخدم والحاسب فى صورة سهلة ومريحة للمستخدم.

(5) وحدة تحديث المعرفة Knowledge Update Unit

عن طريقها يتم إضافة أو تعديل أو حذف الخبرة والمعرفة الموجودة بقاعدة المعرفة.

• تمثيل المعرفة فى النظم الخبيرة

نجد أن الصعوبة الأساسية فى بناء النظم الخبيرة تتركز فى إختيار الأسلوب الملائم لتمثيل المعرفة ووصف خبرات الخبير ، بحيث يتعين الجمع بين قدرة الأسلوب المختار على التعبير (أى سهولة وصف معرفة الخبير وقراءتها) وبين كفاءة عمليات المعالجة الالية (أى الوقت المستغرق من الحاسب فى معالجة أسلوب التمثيل المختار).

وهنا نواجه تناقضاً ، فالتمثيل الذى يستخدم اللغة الطبيعية التى تلائم المستخدم ، بكل ما لها من حرية فائقة لوصف المعرفة ، يتطلب جهداً كبيراً من الحاسب لكشف مضمونه وفهمه فهماً سليماً، بينما التمثيل القائم على لغة برمجة مثلاً بكل ما فيها من قيود على حرية التعبير ، سيجعل عمليات الحاسب سريعة للغاية.

وهناك أساليب عديدة لتمثيل معرفة الخبراء فى النظم الخبيرة ويعتبر أهمها ثلاثة اساليب هى قواعد الإنتاج ، الأطر ، الشبكات الدلالية وسوف نقوم بإلقاء الضوء عليها مع التركيز على تمثيا المعرفة بإستخدام الأطر (Frames) والذي تم إستخدامه في تصميم نظام الإرشاد الأكاديمي في هذا البحث .

○ قواعد الإنتاج Production Rules

تعتبر قواعد الإنتاج من أكثر الأساليب الشائعة لتمثيل المعرفة وتخزينها، وأى قاعدة إنتاج تمثل عادة بإستخدام جملة أو مجموعة من الجمل الشرطية (IF-THEN) (إذا – حينئذ) كما يتضح من الشكل العام التالى

IF : C is true

THEN : conclude X

بمعنى أنه إذا : تحققت الشروط المنطقية (C)

حينئذ : نفذ الأنشطة والأعمال المشار إليها ب(X)

ويختلف تركيب الجملة فى قاعدة الإنتاج من نظام خبير لآخر. ويوجد عادة شكلان لتركيب الجملة فى أى نظام خبير. الأول هو التركيب الداخلى لتمثيل قاعدة المعرفة فى النظام وهو التمثيل الأقرب إلى لغة النظام. الثانى هو الشكل الخارجى وهو تركيب أكثر قبولاً وسهولة للمستخدم ويسمح له بإستعراض وقراءة قواعد الإنتاج بسهولة

- ويطلق على النظم الخبيرة التى تستخدم قواعد الإنتاج فى تمثيل المعرفه "نظم الخبرة المعتمدة على القواعدRule based Expert Systems " واحياناً تسمى "نظم الإنتاج Production systems "

o الأطـــر Frames

إن لفظ "إطار Frame " فى مجال الذكاء الإصطناعى يعنى طريقه خاصه لتمثيل السمات والمواقف العامة والمشتركة. وقد عرف" مارفن منسكى Marvin Minsky 1975 "وهو أول من إبتكر تمثيل المعرفة بإستخدام الأطر معنى كلمة إطار كما يلى:

"الإطار عباره عن هيكل بيانات (Data Structure) لتمثيل مواقف واراء تتكرر على نحو لايتغير ويتفق مع نمط عام تعوزه فقط بعض المفاهيم الفردية المميزة ويتصل بكل إطار أنواع مختلفة ومتعددة من المعلومات . بعض هذه المعلومات يتعرض إلى كيفية إستخدام الإطار والبعض الآخر يتعرض إلى ما يمكن توقع حدوثه فى خطوة تالية، بالإضافة إلى معلومات أخرى عن كيفية التصرف فى حالة عدم تأكد ثبوت التوقعات المحتملة"

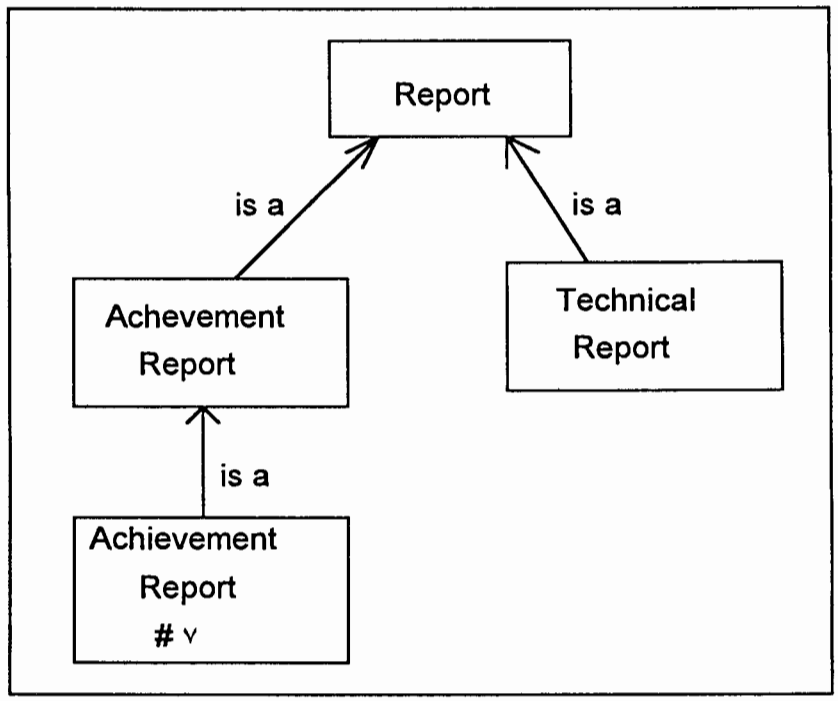
ويتكون الإطار من شبكة من العقد (Network of Nodes) بينها علاقات (Relations) منظمة بأسلوب هرمى ، حيث تمثل أعلى عقدة (Topmost node) المفاهيم العامة وتمثل العقد الأدنى مزيد من الأمثلة أوالمراحل الخاصة المرتبطة بالمفهوم الذى تمثله العقدة العليا. ومثال ذلك أن المفهوم "تقرير" مثلاً يمكن تمثيله بإستخدام الأطر كما هو مبين فى شكل (3).

ويرتبط بكل عقدة (node) مجموعة الخصائص والصفات (Attributes) التى تعرف المفهوم وتميزه. وتمثل كل خاصية أو صفة مميزة بإستخدام حيز (Slot) ويرتبط بكل حيز أى بكل صفة أو خاصية إسم (Name) وقيمة (Value) وربما فئة (Set) من المظاهر (facets) تعطى معلومات أكثر عن الخاصية أوالصفة مثل (نوع القيمة التى يمكن أن ترتبط بالخاصية – القيمة المبدئية للخاصية – توثيق عن الخاصية – قيود على القيم المسموح بها للخاصية – مجموعه من الإجراءات (Procedures) ويوضح شكل (4) نموذج لمكونات العقدة فى نظام الأطر.

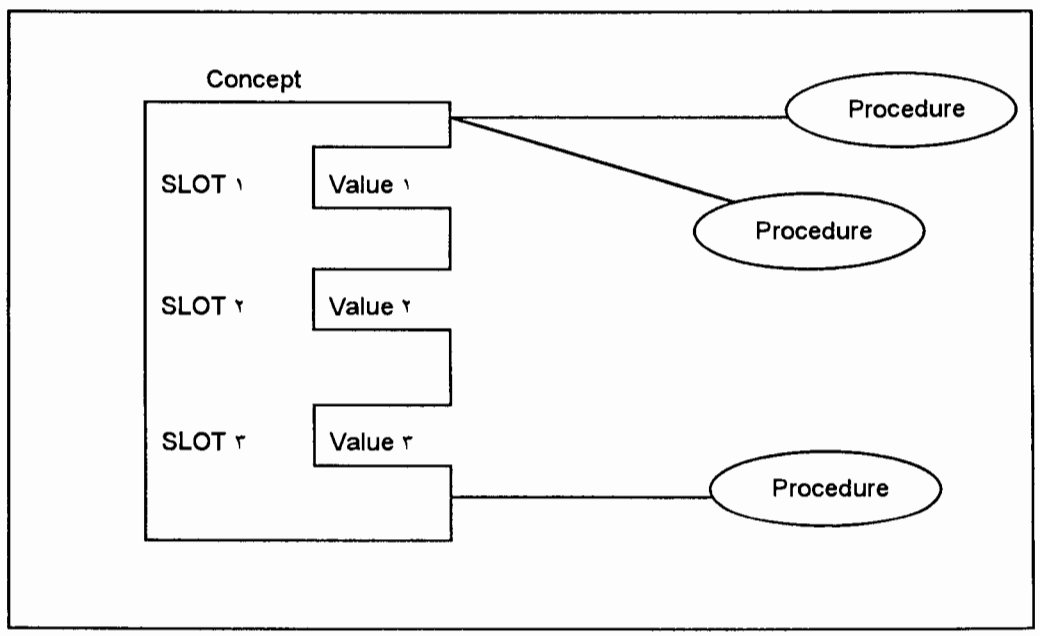
وعادة توجد ثلاثة أنواع من الإجراءات التى يمكن أن يرتبط بها أى حيز (slot)هى:

1- **If-added Procedure**

ويتم تنفيذه عند إضافة أى معلومات جديدة بالحيز.



شكل (3)



شكل (4)

2- If-removed Procedure

ويتم تنفيذه عند حذف أى معلومات من بالحيز.

3- If-needed Procedure

ويتم تنفيذه عندما يكون هناك حاجه إلى معلومات من الحيز ولا يحتوى عليها.

وهذه الإجراءات تقوم بمراقبة تخصيص المعلومات المناسبة إلى كل عقدة من عقد النظام والتأكد من تنفيذ الأنشطة المناسبة عند تغير القيم (Values) بالحيز.

ولتوضيح كيفيه عمل نظام الأطر (Frames) فإن هرمية التقرير المكتوب الموضحه فى الشكل السابق تم رسم كل عقده بها بالتفصيل موضحاً بها إسم الحيز (slot name) والقيمه المرتبطه به (Associated value) والإجراءات (Procedures) فى شكل (5).

ويلاحظ أن الحيز يمكن أن يحتوى على قيمة مبدئية (Default value) مرتبطة به ، مثال ذلك أن قيمة الحيز المسمى مؤلف (Author) لتقرير الإنجاز (Achievement Report) تكون هى رئيس القسم (Chairman) ما لم يتم إعطاء معلومات تخالف ذلك.

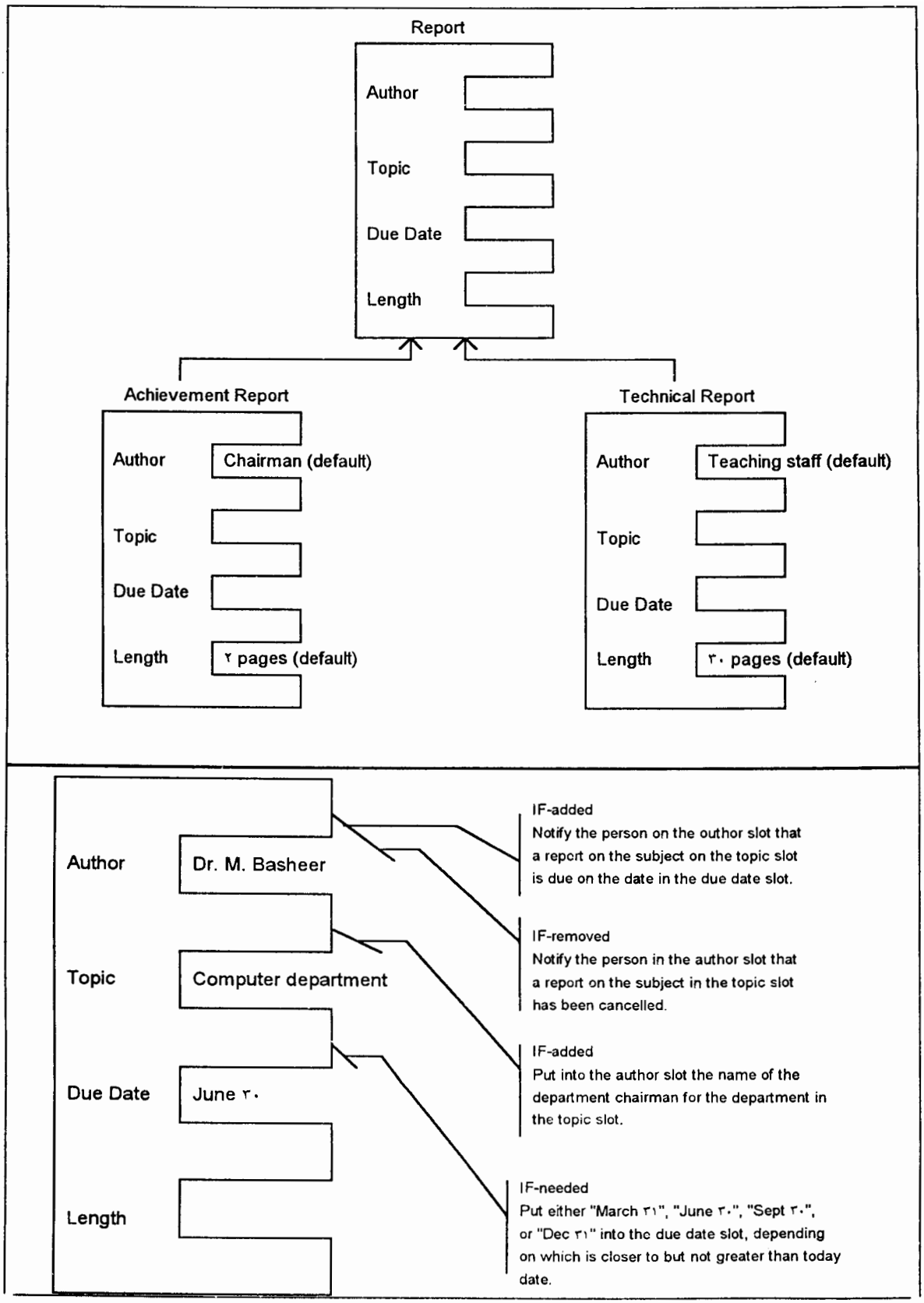
والسؤال الآن كيف يمكن للمستخدم التعامل مع المعارف المنظمة بهذا الأسلوب؟ وللإجابة على هذا السؤال سنعود مرة أخرى إلى الشكل السابق ونفرض أن عميد كلية فى جامعة ما قام بإستخدام نظام أطر بهذا الشكل من خلال النهاية الطرفية للحاسب ومن خلال برنامج وسيط المستخدم (User Interface Program) كتب بإستخدام لوحة المفاتيح ما يلى :

"I need an achievement report on the computer department"

بمعنى : "أنا أريد تقرير إنجاز عن قسم الحاسب".

فيقوم برنامج وسيط السمتخدم بتحليل هذه العبارة وإدخال قسم الحاسب (Computer Department) (فى الحيز الخاص بالموضوع (Topic Slot) فى أول عقدة فارغة لتقرير الإنجاز (Achievement report) وتكون فى هذه الحالة رقم (#7) مثلاً. ويبدأ بعد ذلك توالى العمليات بصورة آلية كما يلى :

1- يتم تنفيذ الإجراء (IF-added) المرتبط بـ (Topic Slot) وذلك لأنه تم إدخال قيمة (Value) وهى (Computer Department) فى ذلك الحيز (Slot) وهذا الإجراء عباره عن مجموعة من الخطوات نقوم بالبحث فى قاعدة بيانات مصاحبة للنظام وذلك للحصول على إسم رئيس القسم الخاص بالقيمة التى تم إدخالها وهى قسم الحاسب (Computer Department) وبفرض أن إسم رئيس القسم وجد (Dr. M. Basheer) فيتم إدخاله فى الحيز الخاص بالمؤلف (Author Slot) بعقدة تقرير الإنجاز رقم 7


Report
Author
Topic
Due Date
Length
Achievement Report
Author
Chairman (default)
Topic
Due Date
Length
٢ pages (default)
Technical Report
Author
Teaching staff (default)
Topic
Due Date
Length
٣٠ pages (default)
Author
Dr. M. Basheer
Topic
Computer department
Due Date
June ٣٠
Length
IF-added
Notify the person on the outhor slot that a report on the subject on the topic slot is due on the date in the due date slot.
IF-removed
Notify the person in the author slot that a report on the subject in the topic slot has been cancelled.
IF-added
Put into the author slot the name of the department chairman for the department in the topic slot.
IF-needed
Put either "March ٣١", "June ٣٠", "Sept ٣٠", or "Dec ٣١" into the due date slot, depending on which is closer to but not greater than today date.

شکل (5)

2- يتم تنفيذ الإجراء (IF-added) المرتبط بـ (Author slot) وذلك لوجود قيمة تم إدخالها وهى الإسم (Dr. M. Basheer)ويبدأ هذا الإجراءفى تكوين رسالة لإرسالها إلى

(Dr. M. Basheer) ولكنه يكتشف أن هناك قيمة يحتاجها وهى تاريخ الإستحقاق أى تاريخ إنهاء التقرير المطلوب (Due Date) وهذه القيمة غير موجودة.

3- بمجرد إكتشاف الإجراء (If-added) لعدم وجود قيمة فى الحيز الخاص ب (Due Date) فإنه يقوم بتشغيل الإجراء (If-needed) المرتبط بهذا الحيز حيث يقوم بإيجاد تاريخ اليوم الحالى (Today's date) ويقرر أن 30/6 هو أقرب يوم له. ويقوم الإجراء بإدخال (June 30) فى الحيز الخاص بتاريخ الإستحقاق.

4- يكتشف الإجراء (If-added) المرتبط بحيز المؤلف (Author Slot) عدم وجود القيمة التى تعبر عن طول التقرير ولكن يلاحظ أن الحيز الخاص بطول التقرير فى العقدة الخاصة بتقرير الإنجاز رقم 7 غير مرتبط بها أى إجراءات لكى تقوم بالمساعدة فى إيجاد هذه القيمة ويلاحظ أن العقدة الموجودة أعلى هذه العقدة والخاصة بالمفهوم العام لتقرير الإنجاز والمسماه (Achievement Report) يوجد بها قيمة مبدئية (Defullt Value)للطول وهى 2 صفحه فيقوم الإجراء بإستخدام هذه القيمة وتركيب الرساله التاليه:

"Dr. M. Basheer: Please complete an achievement report on the computer Department by June 30. The suggested length is 2 pages."

وإذا حدث فى أى وقت أن أزيل إسم (Dr. M. Basheer) من الحيز الخاص بالمؤلف فإن النظام سوف يرسل آلياً اليه رسالة تفيد بأنه تم إلغاء التقرير المطلوب. وذلك بسبب الإجراء (If-removed)المرتبط بحيز المؤلف كما هو موضح بالشكل السابق

ويعتقد بعض العاملين فى مجال الذكاء الإصطناعى أن أسلوب البشر فى تخزين معارفهم يكون فى كثير من جوانبه على هيئة سلسلة من الأطر النمطية التى تتفق مع كثير من المواقف. وتكون المهمة الأولى لهم عندما يواجهون أى موقف من المواقف هى إختيار الإطار المناسب لذلك الموقف وإتباع جميع الإجراءات المحددة والموصفة مسبقاً والتى تتفق مع ذلك الموقف. وعلى جانب آخر يرى كثير منهم أن أداء نظام الأطر يتأثر إلى حد كبير بتنظيم وتركيب هذه الإطر. لذلك يجب العناية الزائدة وبذل الجهد الكبير من أجل التصميم الجيد لمثل هذه النظم لضمان تميز أدائها.

○ الشبكات الدلالية Semantic Nets

تعتمد الشبكات الدلالية على تركيب شبكى لتمثيل المعرفة. وقد تم تصميم الشبكات الدلالية فى البداية لإستخدامها كنماذج سيكولوجية لذاكرة الإنسان، ولكنها حالياً تعتبر أسلوباً قياسياً لتمثيل المعرفة فى نظم الذكاء الإصطناعى والنظم الخبيرة.

وتتكون الشبكات الدلالية من مجموعة من العقد (nodes) تتصل فيما بينها بأقواس (Arcs) توضح العلاقات بين العقد المختلفة. وتمثل العقد الأهداف (objects) أو المفاهيم (Concepts) أو الأحداث (Events) أو المواقف

(Situations) وترسم العقدة على هيئة نقطة (dot) أو دائره (Circle) أو مستطيل (box) . وتوضح الأقواس العلاقات بين العقد مثل علاقه (is a) وعلاقه (has-part) لتمثيل التسلسل الهرمى بين العقد أو علاقات (agent) ، (Object) ، (Receipient) التى تستخدم فى وصف اللغات الطبيعية .

• الإستدلال فى النظم الخبيرة

إن خاصية الإستدلال (Inference) وخاصية التعليل (Resoning) من أهم السمات المميزة لمجال الذكاء الإصطناعى بشكل عام والنظم الخبيرة بشكل خاص. حيث تميز هذه السمات النظم الخبيرة عن البرامج الإجرائية التقليدية (Conventional Procedural Programs) المتعارف عليها. ولتوضيح خاصية الإستدلال، نفرض أن النظام الخبير يحتوى على الحقيقة "لكل طائر جناحان" ويحتوى أيضاً على الحقيقه "البلبل طائر "فإن النظام الخبير يستطيع أن يستدل (infer) على أن "البلبل له جناحان" دون أن يتم إخباره مباشرة بذلك. أما خاصية التعليل (reasoning) فهى تعنى قدرة النظام الخبيرعلى سرد الأسباب المنطقية المتتابعة التى أدت إلى وصول النظام الخبير إلى قرار محدد بشأن مشكلة معينة أو توصية بذاتها لحل هذه المشكلة.

وكما سبق أن وضحنا فإن الجزء الأساسى من النظام الخبير الذى يقوم بعملية الإستدلال المنطقى هو آلة الإستدلال (Inference Engine) . وتستخدم آلات الإستدلال أساليب عديدة للإستدلال المنطقى أكثرها إنتشاراً وإستخداماً أسلوب التسلسل المتقدم (Forward Chaining) وأسلوب التسلسل الراجع (Backward Chaining) ولا يتسع المجال هنا لتوضيح هذه الأساليب .

Frames

CLASS Dcourse

Class Name :	Dcourse	
Subclasses :	Course	
Properties :	Code(String)	Unknown
	Title (String)	Unknown
	Credit_hr(Num)	Unknown
	Pre1(String)	Unknown
	Pre2 (String)	Unknown
	Term(String)	Unknown
	Lab_Hr(Num)	Unknown
	Lect_Hr(Num)	Unknown

CLASS Course SINGLE EXTERNAL "dBASEIII " INHERITS Dcourse

CLASS Rooms

Class Name :	Rooms	
Subclasses :	Drooms, Halls	
Properties :	RoomNo(String)	Unknown

CLASS Halls SINGLE EXTERNAL "dBASEIII " INHERITS Rooms

CLASS Dlecturer

Class Name :	Dlecturer	
Subclasses :	Lecturer, Lecturer Table	
Properties :	Name(String)	Unknown

CLASS Lecturer SINGLE EXTERNAL "dBASEIII" INHERITS Dlecturer

CLASS Regist2n SINGLE EXTERNAL "dBASEIII

Class Name :	Regist2n	
Subclasses :		
Properties :	Course_no(String)	Unknown
	St_No(Num)	Unknown

CLASS Dresult

Class Name :	Dresult	
Subclasses :	Result	
Properties :	Term(String)	Unknown
	Course1(String)	Unknown
	Grade1(String)	Unknown
	Course2(String)	Unknown
	Grade2(String)	Unknown
	Course3(String)	Unknown
	Grade3(String)	Unknown
	Course4(String)	Unknown
	Grade4(String)	Unknown
	Course5(String)	Unknown
	Grade5(String)	Unknown
	Course6(String)	Unknown
	Grade6(String)	Unknown
	Course7(String)	Unknown
	Grade7(String)	Unknown
	Course8(String)	Unknown
	Grade8(String)	Unknown
	Course9(String)	Unknown
	Grade9(String)	Unknown
	Course10(String)	Unknown
	Grade10(String)	Unknown
	Course11(String)	Unknown
	Grade11(String)	Unknown
	Course12(String)	Unknown
	Grade12(String)	Unknown
	St_No(Num)	Unknown

CLASS Result SINGLE EXTERNAL "dBASEIII" INHERITS Dresult

CLASS StClass SINGLE EXTERNAL "dBASEIII "

Class Name :	StClass	
Subclasses :		
Properties :	Code(String)	Unknown
	St_No(Num)	Unknown
	Class(String)	Unknown
	Type(String)	Unknown

CLASS Dstudent

Class Name :	Dstudent	
Subclasses :	Student	
Properties :	Name(String)	Unknown
	Nation(String)	Unknown
	Birth(String)	Unknown
	Picture(String)	Unknown
	State(String)	Unknown
	No(Num)	Unknown
	Address(String)	Unknown

CLASS Student SINGLE EXTERNAL "dBASEIII" INHERITS Dstudent

CLASS Dschedule

Class Name :	Dschedule	
Subclasses :	TABLES, Course Schedule	
Properties :	Day(String)	Unknown
	Term(String)	Unknown
	Class(String)	Unknown
	Hour(String)	Unknown
	RoomNo(String)	Unknown
	C_code(String)	Unknown
	Lect_code(Num)	Unknown
	Type(String)	Unknown

CLASS TABLES SINGLE EXTERNAL "dBASEIII " INHERITS Dschedule

CLASS Course Schedule

Class Name :	Course Schedule	
Subclasses :	Choose Schedule, Dtables	
Properties :	dcode(String)	Unknown
	tcode(String)	Unknown
	Lect_name(String)	Unknown

CLASS Dtables

Class Name :	Dtables	
Subclasses :		
Properties :	Types(Compound)	Lecture Lab

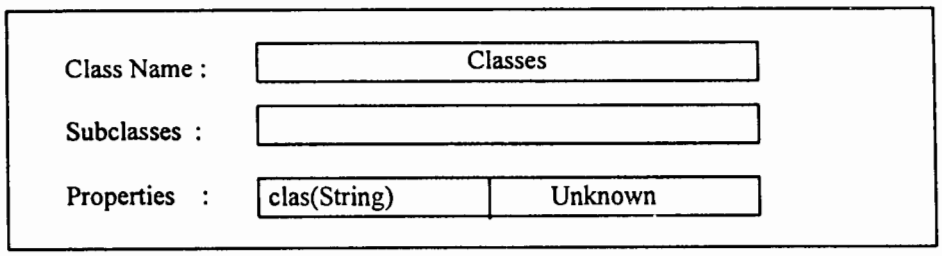
CLASS Grades

Class Name :		
Subclasses :	Grades	
Properties :	Grade(String)	Unknown

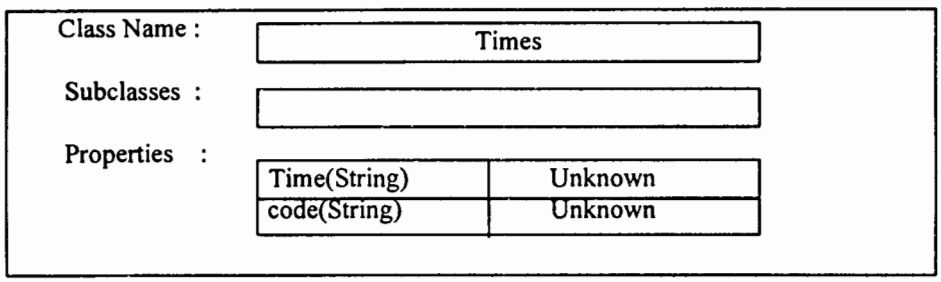
CLASS Days

Class Name :	Grades	
Subclasses :		
Properties :	Day(String)	Unknown
	Code(String)	Unknown

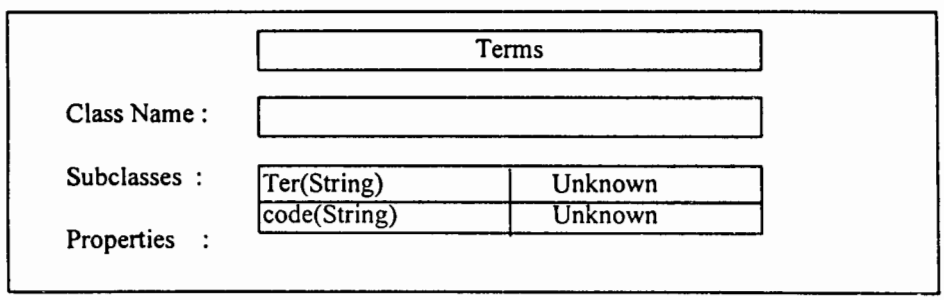
CLASS Classes



CLASS Times



CLASS Terms



CLASS Dterm

Class Name :	Dterm	
Subclasses :		
Properties :	Term1 (Simple)	Unknown(WCM)
	Term2 (Simple)	Unknown(WCM)
	Term3 (Simple)	Unknown(WCM)
	Term4 (Simple)	Unknown(WCM)
	Term5 (Simple)	Unknown(WCM)
	Term6 (Simple)	Unknown(WCM)
	Term7 (Simple)	Unknown(WCM)
	Term8 (Simple)	Unknown(WCM)
	Term9 (Simple)	Unknown(WCM)
	Term10 (Simple)	Unknown(WCM)
	T1 (String)	Unknown
	T2 (String)	Unknown
	T3 (String)	Unknown
	T4 (String)	Unknown
	T5 (String)	Unknown
	T6 (String)	Unknown
	T7 (String)	Unknown
	T8 (String)	Unknown
	T9 (String)	Unknown
	T10 (String)	Unknown

- WCM = WHEN CHANGEG METHOD

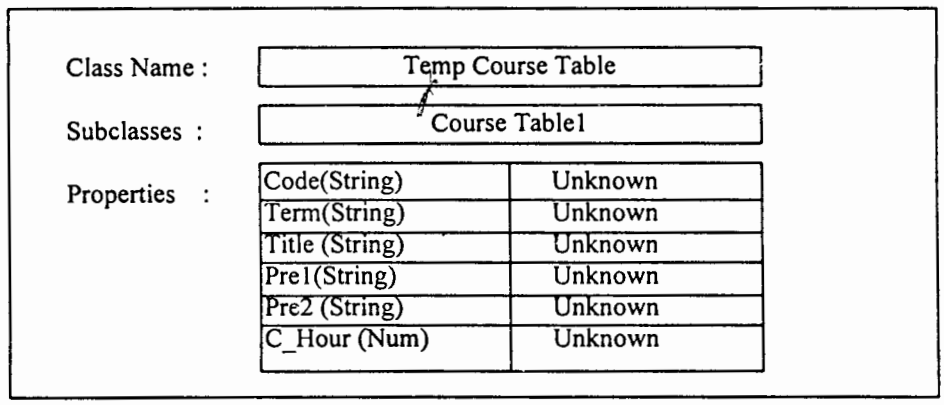
```
WITH Term1 SIMPLE
  WHEN CHANGED
   BEGIN
    IF T1 OF Dterm = "0" THEN
     BEGIN
      Choice := 1
      T1 OF Dterm := "1"
      FIND Course Table1
       WHERE  Term OF Course Table1 = T1 OF Dterm
       WHEN FOUND
        Ensure not studied := TRUE
      FIND END
     END
   END
 WITH Term2 SIMPLE
```

```
  WHEN CHANGED
   BEGIN
    IF T2 OF Dterm = "0" THEN
     BEGIN
      Choice := 1
      T2 OF Dterm := "2"
      FIND Course Table1
       WHERE  Term OF Course Table1 = T2 OF Dterm
       WHEN FOUND
        Ensure not studied := TRUE
      FIND END
     END
   END
 WITH Term3 SIMPLE
  WHEN CHANGED
   BEGIN
    IF T3 OF Dterm = "0" THEN
     BEGIN
      Choice := 1
      T3 OF Dterm := "3"
      FIND Course Table1
       WHERE  Term OF Course Table1 = T3 OF Dterm
       WHEN FOUND
        Ensure not studied := TRUE
      FIND END
     END
   END
WITH Term4 SIMPLE
  WHEN CHANGED
   BEGIN
    IF T4 OF Dterm = "0" THEN
     BEGIN
      Choice := 1
      T4 OF Dterm := "4"
      FIND Course Table1
       WHERE  Term OF Course Table1 = T4 OF Dterm
       WHEN FOUND
        Ensure not studied := TRUE
      FIND END
     END
   END
 WITH Term5 SIMPLE
  WHEN CHANGED
   BEGIN
    IF T5 OF Dterm = "0" THEN
```

```
      BEGIN
       Choice := 1
       T5 OF Dterm := "5"
       FIND Course Table1
        WHERE  Term OF Course Table1 = T5 OF Dterm
        WHEN FOUND
         Ensure not studied := TRUE
       FIND END
      END
   END
WITH Term6 SIMPLE
 WHEN CHANGED
   BEGIN
    IF T6 OF Dterm = "0" THEN
     BEGIN
      Choice := 1
      T6 OF Dterm := "6"
      FIND Course Table1
       WHERE  Term OF Course Table1 = T6 OF Dterm
       WHEN FOUND
        Ensure not studied := TRUE
      FIND END
     END
   END
WITH Term7 SIMPLE
 WHEN CHANGED
   BEGIN
    IF T7 OF Dterm = "0" THEN
     BEGIN
      Choice := 1
      T7 OF Dterm := "7"
      FIND Course Table1
       WHERE  Term OF Course Table1 = T7 OF Dterm
       WHEN FOUND
        Ensure not studied := TRUE
      FIND END
     END
   END
WITH Term8 SIMPLE
 WHEN CHANGED
   BEGIN
    IF T8 OF Dterm = "0" THEN
     BEGIN
      Choice := 1
      T8 OF Dterm := "8"
```

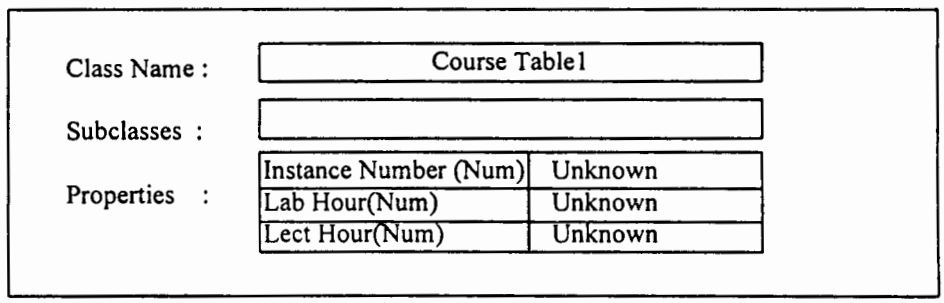
```
      FIND Course Table1
       WHERE  Term OF Course Table1 = T8 OF Dterm
       WHEN FOUND
        Ensure not studied := TRUE
      FIND END
     END
  END
WITH Term9 SIMPLE
 WHEN CHANGED
  BEGIN
   IF T9 OF Dterm = "0" THEN
    BEGIN
     Choice := 1
     T9 OF Dterm := "9"
     FIND Course Table1
      WHERE  Term OF Course Table1 = T9 OF Dterm
      WHEN FOUND
       Ensure not studied := TRUE
     FIND END
    END
  END
WITH Term10 SIMPLE
 WHEN CHANGED
  BEGIN
   IF T10 OF Dterm = "0" THEN
    BEGIN
     Choice := 1
     T10 OF Dterm := "10"
     FIND Course Table1
      WHERE  Term OF Course Table1 = T10 OF Dterm
      WHEN FOUND
       Ensure not studied := TRUE
     FIND END
    END
  END
```

CLASS Temp Course Table



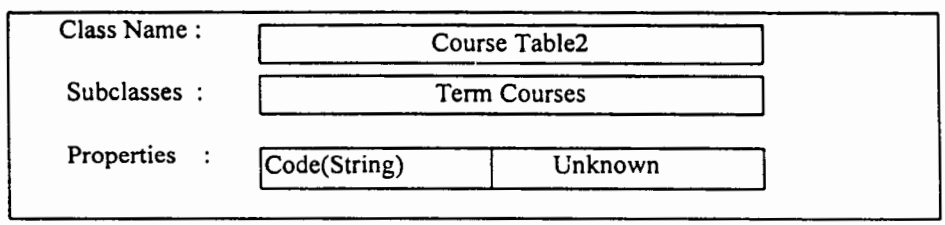
Class Name :	Temp Course Table	
Subclasses :	Course Table1	
Properties :	Code(String)	Unknown
	Term(String)	Unknown
	Title (String)	Unknown
	Pre1(String)	Unknown
	Pre2 (String)	Unknown
	C_Hour (Num)	Unknown

CLASS Course Table1



Class Name :	Course Table1	
Subclasses :		
Properties :	Instance Number (Num)	Unknown
	Lab Hour(Num)	Unknown
	Lect Hour(Num)	Unknown

CLASS Course Table2



Class Name :	Course Table2	
Subclasses :	Term Courses	
Properties :	Code(String)	Unknown

CLASS Term Courses INHERITS Course Table2

Class Name :	Term Courses	
Subclasses :		
Properties :	No(Num)	Unknown

CLASS Course Studied

Class Name :	Course Studied	
Subclasses :	Temp Course Studied	
Properties :	Code (String)	Unknown
	Grade (String)	Unknown
	Instance Number (Num)	Unknown
	C_Hour (Num)	Unknown

CLASS Temp Course Studied

Class Name :	Temp Course Studied	
Subclasses :		
Properties :	Temp(String)	Unknown

CLASS Regst Table

Class Name :	Regst Table	
Subclasses :	Shadow Regist Course	
Properties :	Code (String)	Unknown
	Title (String)	Unknown
	Instance Number(Num)	Unknown
	C_Hour (Num)	Unknown

CLASS Shadow Regist Course

Class Name :	Shadow Regist Course	
Subclasses :		
Properties :	Lect_hour (Num)	Unknown
	Lab_hour (Num)	Unknown

CLASS Shadow Full Courses

Class Name :	Course Table1	
Subclasses :	Reverse, Term Classes	
Properties :	Class (String)	Unknown
	Type (String)	Unknown
	Course Code (String)	Unknown

CLASS Reverse

Class Name :	Reverse	
Subclasses :		
Properties :	Dcode(String)	Unknown
	Tcode (String)	Unknown
	Room(String)	Unknown

CLASS Term Classes INHERITS Shadow Full Courses

Class Name :	Term Classes	
Subclasses :		
Properties :	No (Num)	Unknown

CLASS Shadow course

Class Name :	Shadow course	
Subclasses :		
Properties :	Code (String)	Unknown

CLASS Fail Courses

Class Name :	Fail Courses	
Subclasses :		
Properties :	Code (String)	Unknown

CLASS Elective Course

Class Name :	Elective Course	
Subclasses :		
Properties :	Code (String)Array(3)	Unknown

CLASS Password

Class Name :	Password	
Subclasses :		
Properties :	Pass1(String)	Unknown
	Pass2(String)	Unknown

CLASS prinfile

Class Name :	prinfile	
Subclasses :		
Properties :	filename (String)	Unknown
	print (Simple)	Unknown

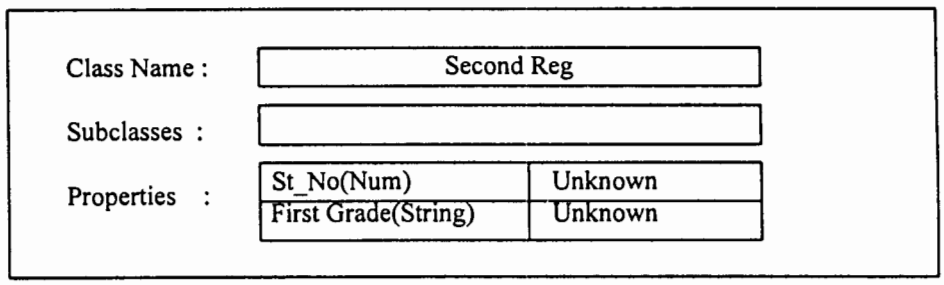
CLASS Print Schedule

Class Name :	Print Schedule	
Subclasses :		
Properties :	Dcode (String)Array(30)	Unknown
	Tcode (String)Array(30)	Unknown
	Course Code (String)Array(30)	Unknown
	Room (String)Array(30)	Unknown
	Class (String)Array(30)	Unknown
	Select (String)Array(8)	Unknown
	Type (String)Array(30)	Unknown
	Time Select (String)Array(8)	Unknown
	Dtime (String)Array(8)	Unknown

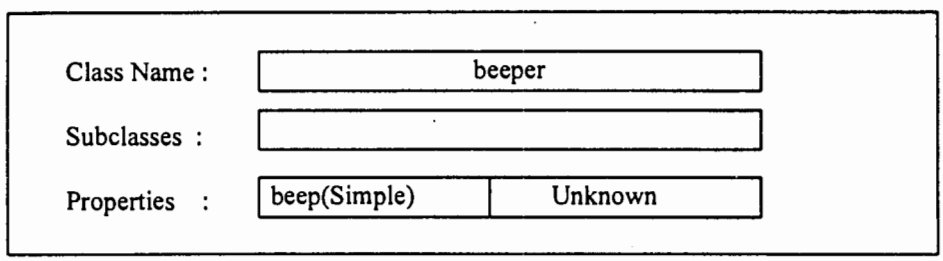
CLASS Results Course

Class Name :	Results Course	
Subclasses :		
Properties :	No(Num)	Unknown
	Name(String)	Unknown
	S_No(Num)	Unknown
	Term(String)	Unknown
	Room No(String)	Unknown

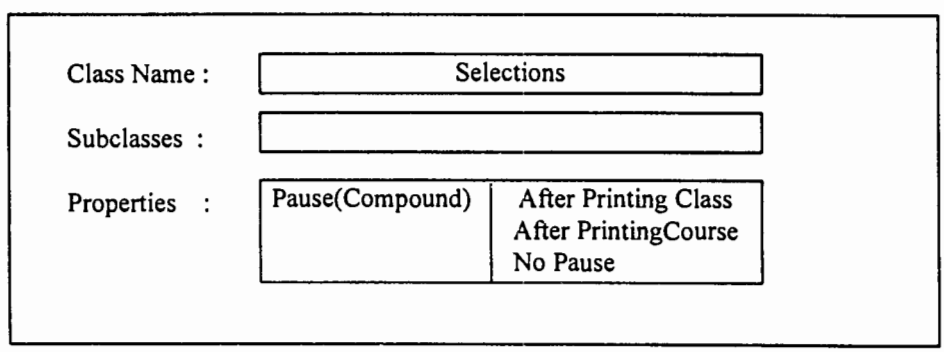
CLASS Second Reg



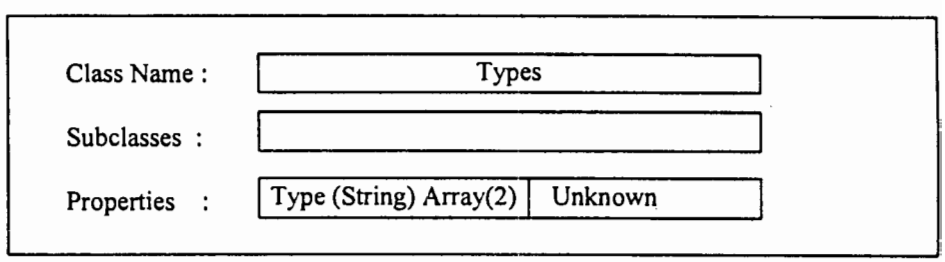
CLASS beeper



CLASS Selections



CLASS Types



RULES

RULE 1

IF Grade OF Course Studied = "A"

THEN Point := 4 * C_Hour OF Course Studied

RULE 2

IF Grade OF Course Studied = "A-"

THEN Point := 11 / 3 * C_Hour OF Course Studied

RULE 3

IF Grade OF Course Studied = "B+"

THEN Point := 10 / 3 * C_Hour OF Course Studied

RULE 4

IF Grade OF Course Studied = "B"

THEN Point := 9 / 3 * C_Hour OF Course Studied

RULE 5

IF Grade OF Course Studied = "B-"

THEN Point := 8 / 3 * C_Hour OF Course Studied

RULE 6

IF Grade OF Course Studied = "C+"

THEN Point := 7 / 3 * C_Hour OF Course Studied

RULE 7

IF Grade OF Course Studied = "C"

THEN Point := 6 / 3 * C_Hour OF Course Studied

RULE 8

IF Grade OF Course Studied = "C-"

THEN Point := 5 / 3 * C_Hour OF Course Studied

RULE 9

IF Grade OF Course Studied = "D+"

THEN Point := 4 / 3 * C_Hour OF Course Studied

RULE 10

IF Grade OF Course Studied = "D"

THEN Point := 1 * C_Hour OF Course Studied

RULE 11

IF Grade OF Course Studied = "F"

THEN Point := 0

RULE 12

IF Points / C_Hour >= 2

THEN State := "Y"

RULE 13

IF Points / C_Hour < 2 AND C_Hour > 36

THEN State := "N"

RULE 14

IF Points / C_Hour < 1.8 AND C_Hour <= 36

THEN State := "N"

RULE 15

IF Points / C_Hour < 2 AND C_Hour <= 36

THEN State := "Y"

RULE 16

IF State = "Y"

THEN Possible Load := 18

RULE 17

IF State = "N" AND Old State = "Y"

THEN Possible Load := 15

RULE 18

IF State = "N" AND Old State = "N"

THEN Possible Load := 12

RULE 19

IF AGPA OF Dstudent >= 11 / 3

THEN Improve Grade := "A-"

RULE 20

IF First Grade OF Second Reg = "F" AND (Grade OF Grades = "A" OR Grade OF Grades="A-" OR Grade OF Grades="B+" OR Grade OF Grades = "B" OR Grade OF Grades = "B-" OR Grade OF Grades = "C+" OR Grade OF Grades ="C")

THEN Correct Grade := "C"

RULE 21

IF First Grade OF Second Reg = "F"

THEN Correct Grade := Grade OF Grades

RULE 22

IF AGPA OF Dstudent >= 10 / 3

THEN Improve Grade := "B+"

RULE 23

IF AGPA OF Dstudent >= 9 / 3

THEN Improve Grade := "B"

RULE 24

IF AGPA OF Dstudent >= 8 / 3

THEN Improve Grade := "B-"

RULE 25

IF Grade_Temp >= Grade

THEN Co_Grade := Grade

ELSE Co_Grade := Grade_Temp

RULE 26

IF Grade OF Course Studied = "W"

THEN Point := 0

RULE 27

IF AGPA OF Dstudent >= 7 / 3

THEN Improve Grade := "C+"

RULE 28

IF AGPA OF Dstudent >= 6 / 3

THEN Improve Grade := "C"

RULE 29

IF AGPA OF Dstudent >= 5 / 3

THEN Improve Grade := "C-"

RULE 30

IF AGPA OF Dstudent >= 4 / 3

THEN Improve Grade := "D+"

RULE 31

IF Repeat Type

THEN ASK message 27

AND Type := 0

ELSE Type := 1

RULE 32

IF Class ok

THEN Class := 1

ELSE ASK message 28

AND Class := 0

RULE 33

IF Time ok

THEN Time := 1

ELSE ASK message 29

AND Time := 0

DEMONS

DEMON 1

IF Make one

THEN I := I + 1

AND MAKE Course Table1

WITH Code := Code OF Course

WITH Term := Term OF Course

WITH Title := Title OF Course

WITH C_Hour := Credit_hr OF Course

WITH Pre1 := Pre1 OF Course

WITH Pre2 := Pre2 OF Course

WITH Lab Hour := Lab_Hr OF Course

WITH Lect Hour := Lect_Hr OF Course

WITH Instance Number := I

AND MAKE Course Table2

WITH Code := Code OF Course

AND action OF Course 1 IS advance := TRUE

DEMON 2

IF selected OF table 1

THEN Pre1 OF Dcourse := Code OF Course Table1

DEMON 3

IF Make Lecturer

THEN MAKE Lecturer Table

WITH Name := Name OF Lecturer

WITH code := Lno

AND action OF Lecturer IS advance := TRUE

DEMON 4

IF selected OF table 3

THEN LecName OF Dlecturer := Name OF Lecturer Table

DEMON 5

IF Make Rooms

THEN MAKE Rooms

WITH Room := RoomNo OF Halls

AND action OF Halls IS advance := TRUE

DEMON 6

IF selected OF table 4

THEN Day OF Dtables := Day OF Days

AND Day code OF Dtables := Code OF Days

DEMON 7

IF selected OF table 5

THEN Term OF Dtables := Ter OF Terms

AND Term code OF Dtables := code OF Terms

AND Find First Course Code := TRUE

DEMON 8

IF selected OF table 6

THEN Class OF Dtables := clas OF Classes

DEMON 9

IF selected OF table 7

THEN Hour OF Dtables := Time OF Times

AND Hour code OF Dtables := code OF Times

DEMON 10

IF selected OF table 8

THEN Room OF Dtables := Room OF Rooms

DEMON 11

IF selected OF table 10

THEN Course_code OF Dtables := Code OF Course Table1

DEMON 12

IF selected OF table 9

THEN Lect_name OF Dtables := Name OF Lecturer Table

AND Lect_code OF Dtables := code OF Lecturer Table

DEMON 13

IF selected OF table 21

THEN Code OF Dcourse := Code OF Course Table2

AND Pause one class := 1

AND FORGET Term Classes

AND Search for Classes := TRUE

DEMON 14

IF selected OF table 23

THEN Term code := code OF Terms

AND Term := Ter OF Terms

DEMON 15

IF selected OF table 26

THEN Droomno OF Drooms := Room OF Rooms

DEMON 16

IF selected OF table 2

THEN Pre2 OF Dcourse := Code OF Course Table2

DEMON 17

IF double clicked OF table 4

THEN Day OF Dtables := "" CF UNDETERMINED

AND Day code OF Dtables := "" CF UNDETERMINED

DEMON 18

IF double clicked OF table 5

THEN Term OF Dtables := "" CF UNDETERMINED

AND Term code OF Dtables := "" CF UNDETERMINED

DEMON 19

IF double clicked OF table 6

THEN Class OF Dtables := "" CF UNDETERMINED

DEMON 20

IF double clicked OF table 7

THEN Hour OF Dtables := "" CF UNDETERMINED

AND Hour code OF Dtables := "" CF UNDETERMINED

DEMON 21

IF double clicked OF table 8

THEN Room OF Dtables := "" CF UNDETERMINED

DEMON 22

IF double clicked OF table 10

THEN Course_code OF Dtables := "" CF UNDETERMINED

DEMON 23

IF double clicked OF table 9

THEN Lect_name OF Dtables := "" CF UNDETERMINED

AND Lect_code OF Dtables := 0 CF UNDETERMINED

◆ المراجع

"الحاسبات الالكترونية حاضرها ومستقبلها" ، أ.د. محمد فهمى طلبه وآخرون ، مؤسسة دلتا كمبيوتر، القاهره ، 1992

"الذكاء الإصطناعى واقعه ومستقبله" ألان بونيه ، ترجمة د. على صبرى فرغلى ، عالم المعرفة عدد 172 ، الكويت ، 1993

" بنية الثورات العلمية" ، توماس كون ، ترجمة شوقى جلال ، عالم المعرفة عدد 168 ، الكويت، 1992.

"Expert Systems Design and Development" , Jhon Durkin, Macmillan, New York , 1994