القسم الاول والثاني

أساسيات هندسة البرمجيات

الكلمات المفتاحية:

تقانة المكونات، المكون، الجوهري، الطارئ، العنصر الثابت، العامل البشري، الزبون، المستخدم، إجرائية تطوير البرمجيات، الإجرائية التكرارية التزايدية، نموذج استحقاق الأهلية، المعيار ISO 9000، لغة النمذجة، لغة النمذجة الموحدة (UML)، استراتيجية العمل، الطريقة SWOT، الطريقة VCM، الأنشطة الرئيسية، الأنشطة الداعمة، الطريقة BPR، مخطط تدفق العمل (workflow)، الطريقة ISA، دورة حياة البرمجيات، المتطلبات، تحديد المتطلبات، توصيف المتطلبات، تصميم البنيان، التصميم التفصيلي، التحقيق، المكاملة، الصيانة، تخطيط المشروع، الجدوى، الاختبارات، المنهجية المهيكلة، المنهجية غرضية التوجه.

ملخص:

تركز هذه الوحدة على التعرف على المفاهيم الأساسية في هندسة البرمجيات، حيث تلقي هذه الوحدة الضوء على مفهوم الجوهري والطارئ في هندسة البرمجيات، كما تعرف المبادئ الأساسية في تخطيط البرمجيات، كما تتناول هذه الوحدة فضلاً عن ذلك مفهوم دورة حياة البرمجيات ومراحلها الأساسية.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- الجوهري والطارئ في هندسة البرمجيات.
  - التعريف
  - العنصر الثابت في تطوير البرمجيات
  - العامل البشري
  - الإجرائية
    - الإجرائية التكرارية التزايدية
    - نموذج استحقاق الأهلية
    - المعيار ISO 9000
  - لغة النمذجة وأدواتها
    - التعريف
  - لغة النمذجة الموحدة (UML)
- تخطيط النظام
  - التعريف
  - طرق وضع الخطط

المنارة للاستشارات
www.manaraa.com

        - الطريقة SWOT
        - الطريقة VCM
        - الطريقة BPR
        - الطريقة ISA
    - مستويات الإدارة الثلاث
- مراحل دورة حياة البرمجيات
    - مرحلة تحديد المتطلبات
    - مرحلة توصيف المتطلبات
    - مرحلة تصميم البنيان
    - مرحلة التصميم التفصيلي
    - مرحلة التحقيق
    - مرحلة المكاملة
    - مرحلة الصيانة
- التخطيط في دورة حياة المنتج البرمجي
- الاختبارات في دورة حياة المنتج البرمجي
- منهجيات تطوير البرمجيات
    - المنهجية المهيكلة
    - المنهجية غرضية التوجه

المنارة للاستشارات
www.manaraa.com

الجوهري والطارئ في هندسة البرمجيات

1-التعريف

- **الجوهري في هندسة البرمجيات:** إن الجوهري في هندسة البرمجيات متضمن في الصعوبات المتأصلة في البرمجيات بحد ذاتهـا، وما يمكن فعله هو التعرف على هذه الصعوبات فقط، وليس سهلاً توضيحها أو تبويبها في نقاط دقيقة محددة.

- **تحديد الجوهري في هندسة البرمجيات:** ليس من السهل تحديد جوهر هندسة البرمجيات وذلك نتيجة لكل مما يلي:
  1. التعقيد الملازم للبرمجيات
  2. الحاجة لتحقيق الانسجام بين البرمجيات والاحتياجات
  3. قابلية تغير البرمجيات
  4. طبيعة البرمجيات غير المرئية

- **الطارئ في هندسة البرمجيات:** الطوارئ هي الصعوبات الناجمة عن الخبرات المتعلقة بإنتاج البرمجيات والتي يمكن إعادتها إلـى عوامل بشرية.

- **فئات الصعوبات الطارئة:** تصنف الصعوبات الطارئة في ثلاث فئات:
  1. العامل البشري
  2. الإجرائية
  3. لغة النمذجة وأدواتها

الجوهري والطارئ في هندسة البرمجيات

2- العنصر الثابت في تطوير البرمجيات

- **تطوير (وليس تصنيع) البرمجيات:** نقول عادةً أننا "نطوّر" البرمجيات ولا نقول أننا "نصنّعها"، لكننا بالتأكيد لا ننكر أن التقدم الذي شهدته هندسة البرمجيات أضاف المزيد من الدقة والوثوقية إلى خبرات التطوير. لكن مع ذلك، وخلافاً للهندسة التقليدية، لا يمكـن ضمان نجاح المشروع البرمجي.

- **الحلول التجارية الجـاهزة** – COTS: تشجع الخبرات المكتسبة بالمزاولة على تطوير النظم انطلاقاً من حزم برمجية قابلة للتعديل والمواءمة، وتعرف هذه الحزم باسم الحلول التجارية الجاهزة COTS (Commercial-of-the-shelf). ويمكن أن تساعد هـذه الحزم في الحصول على إجرائية محاسبة أو تصنيع أو إدارة موارد بشرية، وبذلك أصبح العمل يركز على 'تخصيص' البرمجيات بدلاً من 'تطويرها من الصفر'.

- **خطة تطوير النظم:**
  1. عند تطوير نظام جديد تماماً يجب أن ننشئ البنى المفهومية (أي النماذج) اللازمة للعمل النهائي بحيـث تلبـي الاحتياجـات الخاصة بالمؤسسة.
  2. نقوم بعد ذلك بتخصيص وظائف الحزم البرمجية لتتلاءم مع هذه البنى المفهومية.

المنارة للاستشارات
www.manaraa.com

- قلما تستطيع مؤسسة ما أن تجد حزمة برمجية لأتمتة الأنشطة العملياتية الخاصة بها، فالنشاط المركزي في شركة اتصالات هاتفية يختلف عن النشاط المركزي في شركة محاسبة أو في شركة إدارة موارد بشرية. وما يميز المؤسسة يجب أن يطور من البدء (أو أن يعاد تطويره بدءاً من نظام قديم موجود).

- **تقانة المكونات:** يجب أن تستفيد إجرائية التطوير من استخدام تقانة المكونات:
  - **المكون** هو وحدة برمجية تنفيذية لها وظائف محددة بدقة (خدمات) وبروتوكولات اتصال (واجهات) مع المكونات الأخرى. إذ يمكن إعادة تشكيل المكونات بحيث تلبي احتياجات تطبيق محدد.
  - **تقانات المكونات الشائعة:** من أهم تقانات المكونات الشائعة اليوم نذكر:
    1. تقانة CORBA (Common Object Request Broker Architecture) التي وضعتها المجموعة OMG (Object Management Group)
    2. تقانة DCOM (Distributed Component Object Model) التي وضعتها شركة Microsoft
    3. تقانة EJB (Enterprise Java Beans) من شركة Sun

العامل البشري (Stakeholders)

تعريف العامل البشري: نعني بالعامل البشري كل الأشخاص المعنيين بالمشروع البرمجي بطريقة أو بأخرى، سواء كانوا سيتأثرون بالنظام أو سيؤثرون على تطويره، ونميز هنا فئتين:

1. الزبائن (المستخدمون ومالكو النظام).
2. المطورون (محللون، مصممون، مبرمجون....).

- **الزبون – المستخدم– المستخدم النهائي:** يفضل أن تستخدم كلمة "زبون" (Customer) بدلاً من كلمة "مستخدم" (User). وهذا التمييز ضروري من منظور تطوير النظام:

  - **الزبون:**
    1. هو الشخص الذي يموّل التطوير وهو مسؤول عن اتخاذ القرارات
    2. لا يمكن للمطور أن يتجاهل أو يعدّل متطلبات الزبون حتى لو لم يكن هذا الأخير محقاً، فعندما تعترض المطور متطلبات غير قابلة للتحقيق أو متناقضة عليه أن يتفاوض بشأنها من جديد مع الزبون.

  - **المستخدم:** علينا أن نقرَّ أن مصطلح "مستخدم" متداول على نطاق واسع جداً بمعنى "زبون"، ولذلك فقد نستخدمه بهذا المعنى

  - **المستخدم النهائي:** ينبغي تجنب استخدام مصطلح "مستخدم نهائي" (end-user) الذي استخدم فيما مضى للإشارة إلى ممثل المستخدم الذي ينتخب (بدلاً من المستخدم الحقيقي) للتخاطب مع المطورين.

العامل البشري (Stakeholders) تتمة

- **دور العامل البشري في فشل البرمجيات:**
  - **من ناحية الزبون** تفشل المشاريع لسبب أو أكثر مما يلي:
    1. لا تُفهم كل احتياجات الزبون أو يُفهم بعضها خطأ

المنارة للاستشارات
www.manaraa.com

2. تتغير متطلبات الزبون بتواتر كبير
3. الزبائن غير مهيئين لتقديم موارد كافية للمشروع
4. لا يريد الزبائن أن يتعاونوا مع المطورين
5. للزبائن تصورات وتوقعات غير منطقية
6. لم يعد النظام مفيداً للزبائن

- **من ناحية المطورين:** قد تفشل المشاريع أيضاً لأن المطورين ليسوا على المستوى المطلوب لأداء المهمة، فمع تزايد تعقيد البرمجيات أصبح لمعارف المطورين ومهاراتهم دور أساسي، فبينما يستطيع المطورون الجيدون أن يقدموا حلولاً يمكن أن يقدم المطورون الأكثر كفاءة حلولاً أفضل وأقل كلفة وخلال أزمنة أقصر.

- **إجرائيات ضمان جودة البرمجيات:** لكي تضمن مؤسسة تطوير البرمجيات تسليم منتجات ناجحة لزبائنها، ولكي تضمن أن يجني الزبائن الفوائد المتوخاة من الأنظمة عليها أن تتبع مجموعة الإجراءات التالية:
  1. استخدام أفضل المطورين.
  2. إجراء تدريب وتأهيل مستمر للمطورين الموجودين.
  3. تشجيع المطورين على تبادل المعلومات والخبرات.
  4. تحفيز المطورين بإزالة العقبات التي تعترضهم وبتنسيق جهودهم لتصب في عمل منتج.
  5. توفير بيئة عمل ممتعة (قد يكون هذا أكثر أهمية من زيادة الرواتب التي تحدث من حين لآخر).
  6. التوفيق بين أهداف المطورين الشخصية واستراتيجية المؤسسة وأهدافها.
  7. تنمية روح العمل الجماعي.

الإجرائية (Process)

- **إجرائية تطوير البرمجيات:** تحدد إجرائية تطوير البرمجيات الأنشطة والإجراءات التنظيمية اللازمة لرفع مستوى التعاون بين أعضاء فريق التطوير بما يؤدي إلى تسليم الزبائن منتجات عالية الجودة، فنموذج الإجرائية هو الذي:
  1. يثبت ترتيب تنفيذ الأنشطة.
  2. يحدد الجدول الزمني لمراحل التطوير ومواعيد تسليمها.
  3. يوزع الأنشطة والمهام على المطورين.
  4. يقدم معايير تسمح بمراقبة تقدم المشروع، لقياس الإنتاجية من جهة وللتخطيط لمشاريع مستقبلية من جهة أخرى.

- لا يمكن تقييس إجرائيات التطوير أو تنظيمها بحيث تستطيع مؤسسة ما تبنيها، بل على كل مؤسسة أن تطور نموذجاً خاصاً بها، أو مواءمة نموذج عام كالنموذج الذي تقدمه شركة Rational والمعروف باسم "الإجرائية الموحدة" Rational Unified Process.

- **تأثير حجم المشروع على إجرائية التطوير:** قد يكون لحجم المشروع الأثر الأهم على إجرائية التطوير:
  - **المشاريع الصغيرة:** قد لا يحتاج تنفيذ المشاريع الصغيرة (التي تتطلب حوالي عشرة مطورين) لاستخدام أي نموذج، إذ تنحو فرق العمل الصغيرة إلى التواصل والاستجابة للمتغيرات بأساليب غير صورية
  - **المشاريع الكبيرة:** يصبح اعتماد إجرائية معرفة بدقة أمراً لا مفر منه في المشاريع الكبيرة وذلك للسيطرة على عملية التطوير

المنارة للاستشارات
www.manaraa.com

الإجرائية (Process)

(1) – الإجرائية التكرارية التزايدية

- **التعريف:** تتصف إجرائيات التطوير الحديثة بكونها تكرارية وتزايدية، إذ تُحسَّن نماذج النظام وتحول خلال مراحل التحليل والتصميم والتنفيذ، فتضاف التفاصيل على مراحل تكرارية متعاقبة، ويجري إدخال التحسينات كلما دعت الحاجة وتسعى الإصدارات التزايدية من المجتزآت البرمجية إلى تلبية رغبات المستخدم وتزود المطورين بمعلومات هامة لتنفيذ المجتزآت التي ما تزال قيد التطوير.

- **التعريف حسب الإجرائية الموحدة لشركة Rational:**
  - الإجرائية التكرارية هي تلك التي تتطلب إدارة سيل من الإصدارات التنفيذية
  - الإجرائية التزايدية هي تلك التي تنطوي على جهد مستمر لمكاملة بنية النظام لإنتاج تلك الإصدارات بحيث يتضمن كل إصدار جديد تحسينات وإضافات غير متاحة في الإصدار الأسبق

- **شروط مجتزآت النظام:** يعتمد نجاح الإجرائية التكرارية التزايدية على تمييز وتعريف مجتزآت النظام البنيوية في المراحل الأولى:
  - يجب أن تكون حجوم هذه المجتزآت متقاربة
  - يجب أن تتمتع كل منها بتماسك داخلي قوي
  - يجب أن يبقى الترابط بين المجتزأت في الحدود الدنيا

- كما أن لترتيب تنفيذ المجتزآت أهمية أيضاً فقد لا يتمكن المطورون من إصدار بعض المجتزآت إذا كانت تعتمد على معلومات أو حسابات من مجتزآت أخرى لم تطور بعد. إن غياب أو ضعف التخطيط قد يؤدي إلى فقدان السيطرة على سير العمل في المشروع.

الإجرائية (Process)

(2) – نموذج استحقاق الأهلية (Capability Maturity Model)

- تواجه أي مؤسسة تعمل في مجال إنتاج البرمجيات تحدياً أساسياً إذ عليها أن تثبت جدارة وصحة إجرائية التطوير التي تتبناها، ولكي تستطيع المؤسسة أن تحسن إجرائيتها عليها أن تدرك أولاً مشكلات وعيوب إجرائيتها الحالية، ويمثل نموذج استحقاق الأهلية (CMM) طريقة شائعة لتقييم إجرائية التطوير وتحسينها.

- **طريقة العمل:** يعتمد النموذج CMM بشكل رئيسي على استمارة أسئلة يجب أن تملأها المؤسسة ليتم بعد ذلك التحقق منها والتأكد من صحة ما يرد فيها لتعطى المؤسسة مستوى من مستويات النموذج CMM الخمسة، التي يعبر فيها المستوى الأعلى عن نضج وأهلية أفضل لإجرائية التطوير.

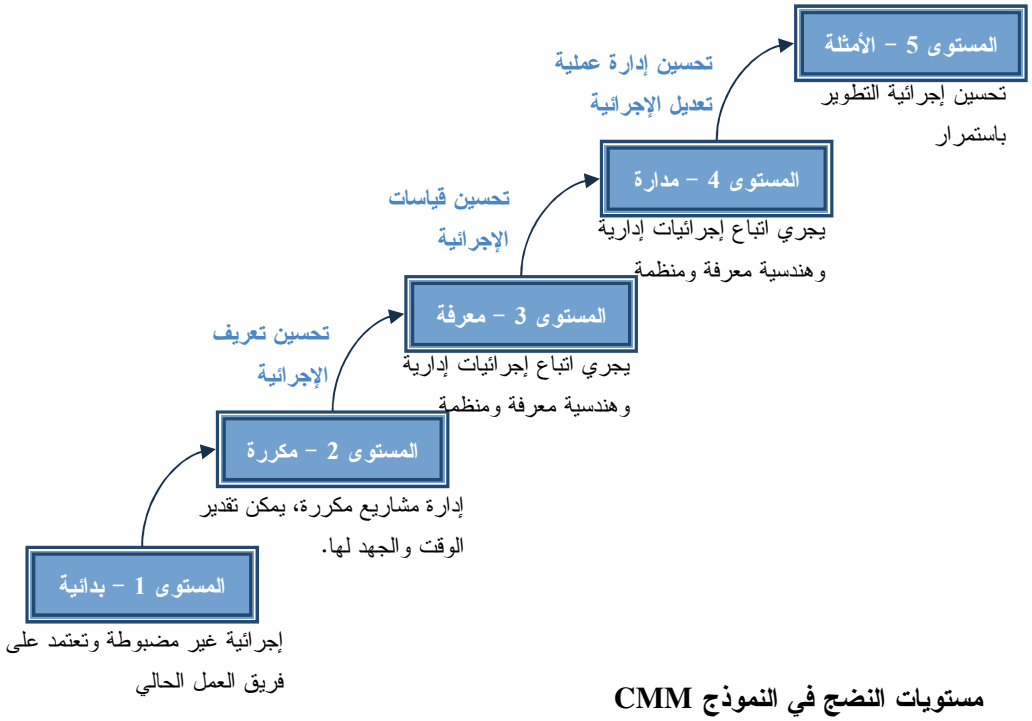
- **مستويات النموذج CMM الخمسة:**
  - **المستوى 5 الأمثلية :**
    - يجري تحسين إجرائية التطوير باستمرار
    - تحسين طريقة تعديل الإجرائية
  - **المستوى 4 مُدارة:**

المنارة للاستشارات
www.manaraa.com

- تستخدم مقاييس للتحكم بإجرائية التطوير
- تحسين مقاييس الإجرائية

- **المستوى 3 معرّفة:**
  - يجري إتباع إجرائيات إدارية وهندسية معرفة ومنظمة
  - تحسين تعريف الإجرائية
- **المستوى 2 مكررة:**
  - إدارة مشاريع مكرَّرة: إدارة مشاريع مكررة
  - إمكانية تقدير الوقت والجهد اللازمين لمشاريع متشابهة
  - تحسين مستوى تخصص الإجرائية
- **المستوى 1 بدائية:**
  - إجرائية غير ثابتة وغير واضحة تعتمد على الفريق الحالي

- **الانتقال بين المستويات:** لقد بينت الخبرات العملية والتجربة أن الانتقال من مستوى إلى المستوى الأعلى مباشرة يتطلب عدة سنوات، فمعظم المؤسسات مازالت في المستوى الأول وبعضها في المستوى الثاني، أما المؤسسات التي وصلت المستوى الخامس فعددها قليل جداً.



مستويات النضج في النموذج CMM

الإجرائية (Process)

المعيار ISO 9000

- **معايير الجودة ISO 9000:** وضعتها المنظمة العالمية للمقاييس، وتنطبق معايير ISO على إدارة الجودة وعلى إجرائية إنتاج منتج جيد، إلا أن هذه المعايير عامة فهي تنطبق على أية صناعة وعلى كل أنماط الأعمال بما فيها تطوير البرمجيات.

المنارة للاستشارات
www.manaraa.com

- تعدنا سلسلة المعايير ISO9000 عند استخدام إجرائية صحيحة بالحصول على خرج جيد (منتج أو خدمة): "إن الهدف من إدارة الجودة هو الحصول على منتجات ذات جودة عالية ببناء الجودة في المنتجات نفسها بدلاً من اختبار جودة المنتجات".
- لا تحدد المعايير ISO إجرائيات معينة ولا تلزمنا بأي منها فهي تزودنا فقط بنماذج "لما" يجب إنجازه وليس "كيف" يجب إنجاز الأنشطة، وعلى المؤسسة التي تطلب شهادة ISO أن تقول ما تفعل، وأن تفعل ما تقول وأن تثبت ما تم فعله.

- **الحصول على شهادة ISO:** لكي تحصل مؤسسة ما على شهادة ISO يجب أن تكون قادرة على صنع منتج جيد أو تقديم خدمة جيدة حتى لو تغيرت القوى العاملة فيها بأكملها، مما يتطلب أن توثق المؤسسة كل أنشطتها، وعليه يجب تعريف إجرائيات مكتوبة لكل نشاط بما في ذلك ما يجب فعله عند حدوث أخطاء أو عند تذمر الزبائن. وكما هو الحال في النموذج CMM لا تعطى الشهادة ISO إلا بعد إجراء تدقيق ميداني دوري لأعمالها، وقد أصبحت غالبية المؤسسات مضطرة للحصول على هذه الشهادة نتيجة المنافسة التي تفرضها متطلبات الزبائن.

لغة النمذجة وأدواتها

(1) – التعريف

- **لغة النمذجة:** يحتاج المطورون إلى "لغة" لبناء نماذج مرئية أو غيرها ومناقشتها مع الزبائن ومع بقية المطورين، ويجب أن تسمح هذه اللغة ببناء نماذج بمستويات تجريد مختلفة لتمثل الحلول المقترحة بمستويات تفصيل مختلفة.

- **أساسيات لغة النمذجة:**
  - يجب أن تتمتع اللغة بمكونات "مرئية" معبرة فثمة قول شائع مفاده أن "الصورة خير من ألف كلمة"
  - يجب أن تتمتع "بتصريحات دلالية" قوية، أي يجب أن تسمح بوصف التعبير "الإجرائي" بعبارات تصريحية. يجب أن نستطيع التواصل فيما بيننا بقول "ما" يجب فعله بدلاً من وصف "كيف" نفعله.

- الأدوات المساندة في هندسة البرمجيات: يحتاج المطورون أيضاً إلى أداة مساندة في هندسة البرمجيات (CASE Tool). وهي أداة تسمح بتخزين النماذج واسترجاعها بصيغ نصية وبيانية عبر خازنة (repository) تسمح لأكثر من مستخدم (أي أكثر من مطور) بالتشارك باستخدام النماذج.

- **وظائف خازنة الأداة النمطية:** يمكن تلخيص هذه الوظائف كما يلي:
  - تنظيم عمليات الوصول إلى النماذج.
  - تسهيل التعاون بين المطورين.
  - تخزين عدة إصدارات من النماذج.
  - تحديد الفروقات بين الإصدارات.
  - السماح بالتشارك في استخدام المفاهيم نفسها في نماذج مختلفة.
  - توليد تقارير ووظائف المشروع.
  - توليد بنى معطيات ورماز برمجية (الهندسة الأمامية).
  - توليد النماذج انطلاقاً من البرمجيات (الهندسة العكسية).

المنارة للاستشارات
www.manaraa.com

لغة النمذجة وأدواتها

(2) – لغة النمذجة الموحدة (Unified Modeling Language)

- **لغة النمذجة الموحدة (UML):** هي لغة نمذجة مرئية عامة الأهداف تستخدم لتعريف ومعاينة وبناء وتوثيق المكونات الصنعية للنظام البرمجي. وقد وضعت هذه اللغة شركة Rational Software Corporation لتجمع أفضل مميزات الطرائق الأسبق، وفي العام 1997 أعلنتها المجموعة OMG (Object Management Group) كلغة نمذجة معيارية، ومنذ ذلك الحين تطورت لغة UML وأصبحت شائعة في صناعة تقانات المعلومات.

- تسمح بنى لغة UML بنمذجة البنية السكونية والسلوك الديناميكي للنظام، إذ يظهر كمجموعة من الأغراض المتعاونة (مجتزآت برمجية) تستجيب لأحداث خارجية لتنجز مهاماً ذات فائدة بالنسبة للزبائن (المستخدمين). ويهتم كل نموذج بتبيان بعض المفاهيم المتعلقة بالنظام ويتجاهل مفاهيم أخرى لتهتم بها النماذج الأخرى، وبحيث تعطي هذه النماذج مجتمعة وصفاً كاملاً للنظام.

- **فئات نماذج لغة UML:** يمكن تصنيف نماذج لغة UML في ثلاث فئات:
  1. نماذج الحالة State models: تصف بنى المعطيات السكونية.
  2. نماذج السلوك Behavior models: تصف علاقات الأفعال بين الأغراض.
  3. نماذج تغير الحالة State Change models: تصف حالات النظام الممكنة عبر الوقت.

- **مفردات لغة UML البنيوية:** تتضمن لغة UML أيضاً عدداً قليلاً من المفردات البنيوية التي تسمح بتجزئة النظام بحيث يمكن تطويره.

تخطيط النظام

1- مقدمة

- يحتاج العمل في مشاريع نظم المعلومات إلى وضع خطط مناسبة، إذ يجب قبل بدء العمل تعريف المشاريع وتصنيفها وترتيبها بإعطائها أولويات تؤهلها للتطوير، أو التحسين وقد تؤدي أيضاً إلى حذف بعضها، ويجري ترتيب المشاريع عادة وفقاً لدى ملاءمتها لتقانات وتطبيقات نظم المعلومات التي تعود على المؤسسة بالفائدة العظمى، ولابد هنا أن يرتكز القرار على استراتيجية العمل والتخطيط المنهجي الدقيق.

- **استراتيجية العمل:** يجري تحديد استراتيجية العمل عبر إجرائيات متنوعة وبغض النظر عن تفاصيل المناهج المختلفة والفروق بينها يمكن القول أنها تهتم جميعها بدراسة إجرائيات العمل الأساسية في المؤسسة بهدف وضع تصور مستقبلي على المدى البعيد لنمط وآلية العمل ومن ثم إعطاء الأولوية للمواضيع والمشكلات التي يمكن حلها باستخدام تقانات المعلومات.

- **استراتيجيات العمل في المؤسسات الصغيرة:** تفتقر العديد من المؤسسات، خصوصاً الصغيرة منها، لوجود استراتيجية عمل واضحة، وغالباً ما تختار هذه المؤسسات نظم المعلومات التي ترغب بتطويرها تبعاً لأكثر مشكلات العمل إلحاحاً. فإذا تغيرت بنية عمل المؤسسة أو تغيرت شروط العمل داخلها تضطر لتعديل نظم المعلومات الموجودة فيها من جديد. ومع أن لهذا المنهج مساوئ واضحة لكنه يسمح للمؤسسات الصغيرة بإعادة تركيز الاهتمام على أوضاعها الحالية والاستفادة من الفرص الجديدة المتاحة لمقاومة التحديات الجديدة.

- **استراتيجيات العمل في المؤسسات الكبيرة:** لا تحتمل المؤسسات الكبيرة التغير الدائم في توجهات العمل، وهي في الواقع تملي على المؤسسات الأخرى التي تنحو منحى العمل نفسه توجهات جديدة، بل إنها تقوم إلى درجة معقولة بصياغة المحيط وتشكيله تبعاً لاحتياجاتها الحالية. لكن مع ذلك على المؤسسات الكبيرة أن تنظر إلى المستقبل بدقة وحرص وعليها أن تعرف مشاريع التطوير

المنارة للاستشارات
www.manaraa.com

تبعاً لمنهج يستند إلى خطة، إذ لدى هذه المؤسسات عادة مشاريع كبيرة يحتاج إنجازها إلى وقت طويل، ومن المزعج جداً تبـديلها أو تعديلها. لذلك نحتاج لدراسة فرص المستقبل وتحدياته.

تخطيط النظام

2- طرق وضع الخطط

- **طرق وضع الخطط:** يمكن وضع الخطط بعدد من الطرق المختلفة منها:
  1. SWOT (اختصاراً للكلمات الأربع: Strengths, Weakness, Opportunities, Threats أي: القوة، الضعف، الفرص، التحديات)
  2. استراتيجية المرونة VCM (Value Chain Model، أي نموذج سلسلة القيمة).
  3. الطريقة BPR (Business Process Reengineering أي إعادة هندسة إجرائية العمل)
  4. النموذج ISA (Information System Architecture، أي بنيان نظام المعلومات): لتقـدير احتياجـات المؤسـسة مـن المعلومات

- تشترك طرق التخطيط المذكورة كلها بخاصية هامة، فهي كلها تركز على واقعية الفعل (القيام بالفعل المطلوب فعلاً (effectiveness) بدلاً من التركيز على الفعالية (تنفيذ العمل بشكل صحيح، efficiency)، فحل المسألة الخطأ بطريقة فعالة لا يفيد بشيء.

تخطيط النظام

2- طرق وضع الخطط

(1) – الطريقة SWOT

- **تعريف الطريقة:** تسمح هذه الطريقة بتعريف وتصنيف وترتيب وانتقاء مشاريع تطوير نظم المعلومات بما يتماشى وقوة المؤسـسة وضعفها والفرص المتاحة لها والتحديات التي تواجهها، وهي طريقة تنازلية تبدأ بتحديد مهمة المؤسسة.

- **مهمة المؤسسة:** تجسد مهمة المؤسسة هويتها المميزة وهي تحدد رؤيتها لموقعها المستقبلي، ولكي يكون تعريف المهمة جيداً يجب التركيز على احتياجات الزبائن بدلاً من التركيز على المنتجات والخدمات التي تقدمها مؤسسة ما.

- **تحديد نقاط ضعف ونقاط قوة المؤسسة:** تأخذ استراتيجية العمل المطورة انطلاقاً من مهمة محددة بحسبانها نقاط ضعف ونقاط قوة المؤسسة في مجالات الإدارة والإنتاج والموارد البشرية والتمويل والتسويق والبحث والتطوير وغيرها، وتدرس نقاط القوة ونقـاط الضعف هذه بعناية فائقة، وتستطيع المؤسسة الناجحة أن تتعرف في أي وقت على نقاط ضعفها الحالية وعلى نقـاط قوتهـا التـي توجه تطوير استراتيجية العمل فيها.
  - إن تحديد نقاط ضعف ونقاط قوة المؤسسة هو شرط لازم لكنه غير كاف لنجاح التخطيط، إذ لا تعمل المؤسسة في الفراغ وهي مرتبطة بعوامل خارجية اقتصادية واجتماعية وسياسية وتقنية، ولذلك على المؤسسة أن تعرف التحديات الخارجيـة التي عليها مواجهتها، ومع أن هذه العوامل غير خاضعة لسيطرة الشركة لكن معرفتها ضرورية لتحديد أهداف المؤسـسة وغاياتها.

- **الغايات** (Objectives): تسعى المؤسسة في أي وقت إلى بلوغ غاية واحدة أو عدد قليل من الغايات، والغايات عادة تعرف علـى

المنارة للاستشارات
www.manaraa.com

المدى الطويل (من ثلاث إلى خمس سنوات) أو حتى بدون تحديد الزمن، ومن الأمثلة النمطية للغايات تحسين درجة رضى الزبون، تقديم خدمات جديدة، مواجهة تحديات المنافسة، رفع درجة السيطرة على المزودين وما إلى هنالك.

- **الأهداف** (goals): يجب أن يقترن بكل غاية استراتيجية أهداف معينة يعبر عنها عادة كخطط سنوية، فالغاية "تحسين درجة رضى الزبون" مثلاً يمكن بلوغها بتحقيق الهدف "الاستجابة لطلبات الزبائن بسرعة أكبر".

تخطيط النظام

2- طرق وضع الخطط

(2) – الطريقة VCM

- **تعريف الطريقة:** تعتمد هذه الطريقة على دراسة وتحليل سلسلة الأنشطة الكاملة في المؤسسة – بدءاً من المواد الخام وصولاً إلى بيع وتسليم المنتجات النهائية للزبائن، وتركز هذه الطريقة على أن الضعف في واحد من ارتباطات هذه السلسلة سيسبب فشل السلسلة بأكملها. يساعد هذا النموذج على معرفة وفهم تشكيلات السلسلة التي تمنح المؤسسة القدرة التنافسية الأفضل، ويمكن بعدئذ توجيه مشاريع تطوير نظم المعلومات لتدعم الأجزاء، أو العمليات، أو قنوات التوزيع أو طرق التسوية أو غيرها من الأنشطة التي تزيد قدرة المؤسسة على المنافسة.

- **تصنيف الوظائف التنظيمية:** تصنف الوظائف التنظيمية ضمن فئتين:

1. **أنشطة رئيسية (Primary activities):** تؤدي الأنشطة الرئيسية إلى إنشاء منتج نهائي أو تضييف قيمة إلى المنتج، وتقسم إلى خمس مراحل متعاقبة:
   (1) حسابات لوجستية داخلية، (inbound logistics)
   (2) عمليات (Operations)
   (3) حسابات لوجستية خارجية (Outbound logistics)
   (4) تسويق ومبيع
   (5) خدمات بعد البيع

2. **أنشطة داعمة (Support activities):** لا تضيف أية قيمة إلى المنتج، ليس مباشرة على الأقل، لكنها تبقى أساسية وضرورية، وهي تتضمن:
   (1) الإدارة والبنية التحتية
   (2) إدارة الموارد البشرية
   (3) البحث والتطوير
   (4) تطوير نظم المعلومات

- **الخطوات الخمس للاستفادة من تقانة المعلومات:** هناك خمس خطوات أساسية يمكن أن تتخذها المؤسسة للإفادة من الفرص التي توفرها تقانة المعلومات:

1. تقدير كثافة المعلومات المضمنة في المنتجات والإجرائيات
2. تقييم دور تقانة المعلومات في البنية الصناعية
3. تحديد الطرق الممكنة لاستخدام تقانة المعلومات لزيادة قدرة المؤسسة التنافسية، وترتيب هذه الطرق حسب احتمالات نجاحها.
4. دراسة كيفية الاستفادة من تقانة المعلومات لإنشاء أعمال جديدة
5. وضع خطة للاستفادة من تقانة المعلومات

المنارة للاستشارات
www.manaraa.com

تخطيط النظام

2- طرق وضع الخطط

(3) – الطريقة BPR

- **تعريف الطريقة:** تستند الطريقة BPR إلى حقيقة مفادها أن على المؤسسات الموجودة اليوم أن تعيد تشكيل أنفسها بحيث تتخلى عن مبدأ التقسيم الوظيفي والبنى الهرمية والمبادئ العملياتية التي تعتمدها اليوم.

- **الهدف الرئيسي للطريقة BPR:** إن الهدف الرئيسي للطريقة BPR هو إعادة تصميم إجرائيات العمل في المؤسسة بصيغة جذرية (ولذلك تُعرف الطريقة BPR أحياناً باسم إعادة تصميم الإجرائية (Process redesign).
  - توثق هذه الإجرائيات في مخططات تسمى مخططات تدفق العمل (Workflow) لتجري دراستها وتحليلها. تظهر هذه المخططات تدفق الأحداث والوثائق والمعلومات في إجرائية العمل ويمكن استخدامها لاحتساب الزمن والموارد والتكاليف اللازمة لإنجاز هذه الأنشطة.

- **عقبات التنفيذ:** تعترض تطبيق الطريقة BPR في المؤسسات عقبة أساسية إذ يجب إدراج إجرائية أفقية في بنية إدارية شاقولية تقليدية، ولذلك يجب أن يسبق تطبيق الطريقة مبادرة جدية لتعديل بنية المؤسسة لتتمحور حول فرق التطوير كوحدات تنظيمية رئيسية، وبحيث تكون هذه الفرق مسؤولة عن إجرائية كاملة أو أكثر. إلا أن التعديل الجذري ليس مقبولاً دوماً، إذ لا يمكن تغيير البنى التقليدية بين ليلة وضحاها، فقد يواجَه التعديل بمقاومة تلغي الفوائد المنتظرة من تطبيق الطريقة BPR، وفي ظروف كهذه يمكن أن تستمر المؤسسة بالاستفادة من نمذجة إجرائيات العمل ومحاولة تحسينها بدلاً من إعادة هندستها.

تخطيط النظام

2- طرق وضع الخطط

(4) – الطريقة ISA

- **تعريف الطريقة:** تعتبر الطريقة ISA، خلافاً للطرق السابقة، طريقة تصاعدية تُكسب حلول نظم المعلومات إطار عمل حيادي يمكن أن يلائم استراتيجيات عمل متنوعة. ولا تتضمن هذه الطريقة منهجية تخطيط لكنها توفر فقط إطار عمل يدعم معظم استراتجيات الأعمال.

- يُمثل إطار عمل الطريقة ISA بجدول من ثلاثين خلية موزعة في خمسة صفوف (من 1 إلى 5) وستة أعمدة (من A إلى F)

- تمثل الصفوف الرؤية المختلفة المستخدمة في بناء منتج هندسي معقد كنظام المعلومات، إذ يمثل كل سطر رؤية واحد من اللاعبين الخمسة الأساسيين.
  1. المخطط: يحدد نطاق النظام
  2. المالك: يضع نموذجاً مفهومياً للشركة
  3. المصمم: يضع نموذجاً فيزيائياً للنظام
  4. المنفذ: يعطي حلولاً تقنية تفصيلية
  5. المتعهد: يقدم مكونات النظام

المنارة للاستشارات
www.manaraa.com

- تمثل الأعمدة نماذج البنيان الستة التي يتعامل معها كل من اللاعبين، وهي عبارة عن توصيفات مختلفة لكنها مترابطة فيما بينها أيضاً. وتقدم هذه التوصيفات إجابات اللاعبين على الأسئلة الستة التالية:
  A. ممَّ يتكون الكائن؟ (أي المعطيات عند الحديث عن نظم المعلومات)
  B. كيف يعمل الكائن؟ (أي إجرائيات العمل)
  C. أين يتوضع الكائن؟ (مواقع مكونات المعالجة)
  D. من يتعامل مع الكائن؟ (أي المستخدمون)
  E. متى يتحرك الكائن؟ (الجدولة الزمنية للأحداث والحالات)
  F. لماذا يتواجد الكائن؟ (دوافع الشركة)

- من أهم ميزات الطريقة ISA أنها تقدم إطار عمل مرن بما يكفي للاستجابة للتغيرات المستقبلية التي قد تطرأ على ظروف العمل أو موارده، ويعود ذلك إلى عدم اشتقاق الطريقة ISA من أي استراتيجية عمل محددة فهي مجرد إطار عمل لوصف كامل لنظام المعلومات، يستند إلى خبرات من مجالات عمل متنوعة.

تخطيط النظام

3- مستويات الإدارة الثلاث

- **مستويات الإدارة الثلاث:** لنجاح التخطيط لابد من الانتباه لوجود ثلاثة مستويات إدارية في المؤسسة:
  1. استراتيجي (Strategic)
  2. تكتيكي (Tactical)
  3. عملياتي (Operational)

- **التطبيقات والحلول الموافقة لمستويات القرار:** تتمايز المستويات الثلاثة من حيث القرارات التي تتخذ في كل منها، وتطبيقات نظم المعلومات اللازمة لها، ومن حيث الدعم المطلوب من تقانة المعلومات، وتنطوي عملية التخطيط على تعريف توليفة من تطبيقات نظم المعلومات وحلول تقانات المعلومات الأكثر ملاءمة للمؤسسة في وقت معين.
  - المستوى الاستراتيجي: نجد في المستوى الاستراتيجي التطبيقات والحلول التي تعود على المؤسسة بالفائدة الأكبر، لكنها أيضاً الحلول الأصعب تحقيقاً فهي تحتاج لاستخدام أحدث التقانات كما تتطلب درجة عالية من المهارة والتصميم التخصصي. وهذه الأنظمة هي التي تعطي المؤسسة قدرتها التنافسية.
  - المستوى العملياتي: نجد في الطرف الآخر الأنظمة التي تدعم المستوى العملياتي، وهي أنظمة روتينية تستخدم تقانات قواعد المعطيات التقليدية وغالباً ما تبنى بتخصيص حلول موجودة مسبقاً. ومع أن هذه الأنظمة لن تميز المؤسسة أو تعطيها مقدرة تنافسية إضافية لكن لن تستطيع المؤسسة أن تعمل بدونها.

المنارة للاستشارات
www.manaraa.com

النظم والتقانات الداعمة لمستويات القرار المختلفة			
مستوى اتخاذ القرار	موضوع اهتمام القرار	تطبيقات نظم المعلومات النمطية	حلول تقانات المعلومات النمطية
استراتيجي	الاستراتيجات الداعمة لغايات المؤسسة على المدى الطويل	تحليل التسويق والمبيعات، تخطيط الإنتاج، تقييم الأداء.	تنقيب في المعطيات (طرق إحصائية)، إدارة المعرفة.
تكتيكي	السياسات الداعمة لأهداف المؤسسة على المدى القصير ولتأمين الموارد	تحليل الموازنة، تقدير كتل الرواتب، جدولة جرد الموجودات، خدمة الزبائن.	مخازن المعطيات، المعالجة التحليلية، أوراق الجدول.
عملياتي	الأنشطة اليومية ودعم الإنتاج	إصدار الرواتب، الفوترة، صفقات الشراء، المحاسبة	قواعد المعطيات، معالجة المناقلات، مولدات التطبيقات.

مراحل دورة حياة البرمجيات

- **دورة حياة البرمجيات – Software lifecycle**: يخضع تطوير البرمجيات إلى دورة حياة، وهي عبارة عن مجموعة أنشطة مرتبطة يُدار عبرها أي مشروع تطوير، وتشكل الإجرائيات والطرق آلية تحقيقها، وتعرِّف دورة الحياة المراحل التي على المنتج البرمجي أن يجتازها بدءاً من الاستطلاع الأولي وحتى النهاية.

- **مستويات دورة حياة البرمجيات**: يمكن تمثيل دورة حياة التطوير البرمجي بمستويات مختلفة من الدقة والتفصيل:
  - **مستوى التفصيل الأدنى**: تتضمن هذه الدورة ثلاث مراحل في مستوى التفصيل الأدنى:

1. **التحليل (Analysis)**: تركز مرحلة التحليل على دراسة متطلبات النظام (System Requirements) فتسعى إلى تعريفها وتحديدها من خلال وضع وتطوير نماذج المعطيات والوظائف، كما تسعى أيضاً إلى التعرف على متطلبات النظام غير الوظيفية والقيود الأخرى التي يخضع لها النظام.

2. **التصميم (Design)**: تقسم مرحلة التصميم إلى مرحلتين ثانويتين:
   1) تصميم البنيان
   2) التصميم التفصيلي

   تهتم هذه المرحلة على وجه الخصوص بوضع تصميم لبرنامج الزبون/ المخدم الذي يربط واجهة الاستخدام بأغراض قاعدة المعطيات، كما يجري في هذه المرحلة أيضاً دراسة وتوثيق العديد من العوامل التي تؤثر على إمكانية فهم النظام وصيانته وتوسيعه.

3. **التحقيق (Implementation)**: تنطوي مرحلة التحقيق على ترميز برامج الزبون التطبيقية وقواعد معطيات المخدم. وتظهر في هذه المرحلة أهمية الإجرائيات التكرارية والتزايدية، وتكتسب مراجعة تطبيقات الزبون وقواعد معطيات المخدم بمقارنتها مع نماذج التصميم أهمية جوهرية في نجاح تسليم المنتج.

  - **المستوى التالي**: وفي المستوى الأكثر تفصيلاً تقسم دورة حياة المنتج البرمجي إلى سبع مراحل هي التالية:
    1) تحديد المتطلبات (Requirements Determination)
    2) توصيف المتطلبات (Requirements Specification)

المنارة للاستشارات
www.manaraa.com

3) تصميم البنيان (Architectural Design)
4) التصميم التفصيلي (Detailed Design)
5) التحقيق (Implementation)
6) المكاملة (Integration)
7) الصيانة (Maintenance)

مراحل دورة حياة البرمجيات
1- مرحلة تحديد المتطلبات

- **المتطلب:** يعرف المتطلب بأنه "بيان لخدمة من خدمات النظام أو لقيد من قيوده". ويصف بيان الخدمة السلوك المطلوب من النظام بالنسبة لمستخدم مفرد أو بالنسبة لمجموعة كل المستخدمين.

- **بيان الخدمة:** يعرف بيان الخدمة قاعدة ضبط (business rule) يجب احترامها دوماً (مثلاً: "تدفع الرواتب نصف الشهرية يوم الأربعاء"). وقد يعبر بيان الخدمة عن عملية حسابية يجب أن يجريها النظام (مثلاً: "تحسب عمولة مندوب المبيعات على أساس مبيعات نصف الشهر الأخير باستخدام صيغة حسابية معينة").

- **بيان القيد:** يعبر بيان القيد عن قيد مفروض على سلوك النظام أو على تطويره. وكمثال على القيود المفروضة على سلوك النظام قد نجد قيداً أمنياً مثل: "يحق للمدير المباشر فقط الحصول على معلومات عن رواتب فريقه".

- **لقاءات المطورين والزبائن:** تتضمن عملية تحليل المتطلبات عقد لقاءات بين المطورين والزبائن، وهي خطوة أساسية لحذف المتطلبات المتناقضة والمتقاطعة فيما بينها، ولضمان التقيد بموازنة المشروع وموعد تسليمه.

- **وثيقة المتطلبات:** يجب أن نحصل في نهاية هذه المرحلة على ما نسميه وثيقة المتطلبات (requirements document) وهي عبارة عن وثيقة نصية ذات طابع سردي تحوي بعض الجداول والمخططات غير الصورية، ولن تحوي هذه الوثيقة نماذج صورية باستثناء بعض أساليب التدوين السهلة والشائعة والتي يستطيع الزبائن فهمها بسهولة والتي قد تساعد في تسهيل التواصل بين المطورين والزبائن.

مراحل دورة حياة البرمجيات
2- مرحلة توصيف المتطلبات

- تبدأ مرحلة توصيف المتطلبات عندما يبدأ المطورون بنمذجة هذه المتطلبات باستخدام طريقة محددة (مثل UML). وقد تستخدم هنا إحدى الأدوات المساندة في هندسة البرمجيات لإدخال الوثائق والنماذج وتحليلها، ليتم بالنتيجة إغناء وثيقة المتطلبات بنماذج بيانية وتقارير مهيكلة تنتجها هذه الأدوات، لتحل بذلك وثيقة المواصفات (Specifications document) محل وثيقة المتطلبات.

- تعتبر مخططات الصفوف ومخططات حالات الاستخدام أهم تقنيتي توصيف في منهجية التحليل غرضي التوجه، وهما تقنيتان خاصتان بتوصيف المعطيات والوظائف. وتصف وثيقة المواصفات النمطية متطلبات أخرى غير وظيفية كتلك الخاصة بالأداء والمظهر وقابلية الصيانة ومستوى الأمن وحتى متطلبات سياسية وقانونية.

- يجب أن تبقى نماذج التوصيف، نظرياً على الأقل، مستقلة عن البنية العتادية والبرمجية التي سيعمل عليها النظام النهائي، إذ يؤدي

المنارة للاستشارات
www.manaraa.com

إدخال اعتبارات كهذه إلى إثقال مفردات لغة النمذجة وتعقيدها وبحيث يصعب على الزبائن فهمها مما يعيق بالتالي إمكانية التواصل بين المطورين والزبائن.

مراحل دورة حياة البرمجيات

3- مرحلة تصميم البنيان

- **تصميم البنيان:** نطلق تسمية "تصميم البنيان" على توصيف النظام بدلالة المجتزآت التي تكونه، والذي يجب أن يتخذ قرارات بشأن استراتيجيات الحل بالنسبة لكل من الزبون والمخدم.

- **التصميم التفصيلي:** نطلق تسمية "التصميم التفصيلي" على توصيف العمل الداخلي لكل مجتزأ في النظام، والذي يجب أن يتضمن الخوارزميات التفصيلية وفق المعطيات الخاصة بالمجتزأ والتي يجب أن تخضع لقيود بنية العمل النهائية.

- **استراتيجية الحل:** يهتم تصميم البنيان بانتقاء استراتيجية الحل وتجزئة النظام إلى مجتزآت أساسية، ويجب أن تقدم استراتيجية الحل إجابات كاملة على:
  - المواضيع المتعلقة بالزبون (واجهة الاستخدام)
  - المواضيع المتعلقة بالمخدم (قاعدة المعطيات)
  - أي مشكلة تخص البرمجيات الوسطية اللازمة للربط بين الزبون والمخدم

ويكون القرار المتعلق بالكتل الأساسية (المجتزآت) مستقلاً نسبياً عن استراتيجية الحل، أما التصميم التفصيلي للمجتزآت فيكون مرتبطاً بحل معين.

مراحل دورة حياة البرمجيات

4- مرحلة التصميم التفصيلي

- **التصميم التفصيلي:** يصف تصميم البنيان المنتج بدلالة المجتزآت التي تكونه، أما التصميم التفصيلي فيصف كل مجتزأ من تلك المجتزآت. وفي نظام المعلومات النمطي يرتبط المجتزأ بمكون إما:
  1. لدى الزبون: ويكون المسؤول هو مصمم التطبيق
  2. أو لدى المخدم: ويكون المسؤول هو مصمم قاعدة المعطيات

- **تصميم واجهات الاستخدام البيانية:** من المساوئ الأساسية لتصميم واجهات الاستخدام البيانية (GUI) في المنهجية غرضية التوجه أن المستخدم هو صاحب التحكم وليس البرنامج، الذي عليه أن يستجيب لأحداث عشوائية يولدها المستخدم لينفذ بذلك الخدمات البرمجية الضرورية

- **تصميم قاعدة المعطيات:** يعنى تصميم قاعدة المعطيات بتعريف أغراض قاعدة معطيات المخدم. وهي غالباً علاقاتية وأحياناً غرضية علاقاتية. وتكون بعض هذه الأغراض حاويات لمعطيات (كالجداول مثلاً)، وقد يكون بعضها الآخر أغراضاً إجرائية (كالإجرائيات المخزنة مثلاً)

المنارة للاستشارات
www.manaraa.com

مراحل دورة حياة البرمجيات

5- مرحلة التحقيق

- **تحقيق نظام المعلومات:** يتضمن تحقيق نظام المعلومات تثبيت البرمجيات الجاهزة التي تم شراؤها وترميز البرمجيات المخصصة، كما تتضمن هذه المرحلة أيضاً أنشطة هامة أخرى كالاختبارات وتدريب المستخدمين وفحص العتاد.

- **أنواع المبرمجين:** نميز في فريق التحقيق بين مجموعتين من المبرمجين:
  - الأولى مسؤولة عن تحقيق برمجيات الزبون: تتضمن برامج الزبون النوافذ وخوارزميات التطبيق وعمليات استدعاء برامج قاعدة معطيات المخدم حسب الضرورة
  - والثانية مسؤولة عن تحقيق قواعد معطيات المخدم: تقع على عاتق برامج المخدم مسؤولية الحفاظ على انسجام وتوافق قواعد المعطيات وصحة المناقلات فيها

- **تحقيق واجهات الاستخدام:** قد يتعرض تصميم واجهات الاستخدام إلى بعض التعديلات أثناء تحقيقه، فقد يختار المبرمجون مظهراً مختلفاً للواجهات لتتوافق مع معايير واجهة الاستخدام البيانية المفروضة وبما يسهل عملية البرمجة ويرفع مستوى إنتاجية المستخدم.

- **تحقيق قواعد المعطيات:** قد يؤدي تحقيق قواعد معطيات المخدم إلى تعديل وثائق التصميم، فقد يضطر المبرمجون لتعديل التصميم لأسباب كثيرة نذكر منها على سبيل المثال: مشكلات في قواعد المعطيات لم يكن التنبؤ بها ممكناً، صعوبات في برمجة بعض الإجرائيات المخزنة، مشكلات التزامن، المكاملة مع إجرائيات الزبون، تحسين الأداء وغيرها.

مراحل دورة حياة البرمجيات

6- مرحلة المكاملة

- **المكاملة التزايدية:** يؤدي أسلوب التطوير التزايدي إلى مكاملة تزايدية لمجتزآت البرمجية، وهذه العملية ليست بديهية، ففي الأنظمة الكبيرة قد تتطلب هذه العملية وقتاً أطول وجهداً أكبر من الوقت المستغرق والجهد المبذول في مراحل دورة الحياة الأسبق بما فيها مرحلة التحقيق.

- **صعوبات المكاملة التزايدية:**
  - من أهم الصعوبات التي تعترض المكاملة التزايدية الارتباطات التي قد تظهر بين المجتزآت. ومع أن التصميم المتقن والجيد يخفف درجة الترابط بين المجتزآت إلى الحد الأدنى، لكن لا يمكن التخلص من هذه الارتباطات كلياً. فقد يعتمد أحد مجتزأين على المجتزأ الآخر بحيث لا يمكن أن يعمل أحدهما دون الآخر.
  - لكن ما العمل إذا كنا نحتاج لتسليم مجتزأ قبل أن يكون الآخر جاهزاً؟ قد نجد الحل في كتابة رماز خاص "لسد الفجوات" يمكننا من مكاملة كل المجتزآت.
  - ندعو الإجرائيات البرمجية التي تحاكي نشاط المجتزأ الناقص بالجذوع (Stubs).
- **مكاملة الأنظمة غرضية التوجه:** خلافاً للأنظمة البرمجية التقليدية التي تستند إلى مفهوم البرنامج الرئيسي (main Program) لن نجد في الأنظمة الحديثة غرضية التوجه والمساقة بالأحداث جزءاً مركزياً كالبرنامج الرئيسي. مما يعني أننا لن نستطيع تعريف بنية مكاملة واضحة في هذه الأنظمة، ولا يمكننا أن نطبق عليها استراتيجيات المكاملة التقليدية (التنازلية والتصاعدية). يجب أن تصمم الأنظمة غرضية التوجه مع أخذ هذه الملاحظة بالحسبان أي:
  - يجب أن تبقى المجتزآت مستقلة فيها بينها قدر الإمكان
  - يجب التعرف على الارتباطات المحتملة بين الواحدات وتقليصها في مرحلتي التحليل والتصميم.

المنارة للاستشارات
www.manaraa.com

- من الناحية النظرية المثالية يجب أن يمثل كل مجتزأ مسار معالجة يجري تنفيذه كاستجابة لطلب معين من الزبون، وعليه يجب أن نتجنب استخدام جذوع تحل محل عمليات فعلية كلما كان ذلك ممكناً.

مراحل دورة حياة البرمجيات

7- مرحلة الصيانة

- **مرحلة الصيانة:** تبدأ مرحلة الصيانة بعد تسليم الزبون أول مجتزأ برمجي وفي بعض الأحيان بعد تسليم المنتج بأكمله. وليست الصيانة مجرد جزء من دورة حياة المنتج البرمجي فهي تشكل الجزء الأكبر من هذه الدورة من حيث وقت وجهد فريق التطوير، يقدر الزمن اللازم لصيانة البرمجية بحوالي 67 % من دورة حياتها.

- **مراحل الصيانة الثلاث:** تتضمن الصيانة ثلاث مراحل مختلفة:
  - **صيانة التشغيل:** نقصد بصيانة التشغيل إجراءات الصيانة التكرارية اللازمة لإبقاء النظام قادراً على العمل ولضمان مقدرة المستخدمين على الوصول إليه
  - **صيانة المواءمة:** نقصد بصيانة المواءمة مراقبة عمل النظام وتدقيق وتصحيح وظائفه لتناسب بنية العمل المتغيرة وتحسين أداء النظام
  - **الصيانة الشاملة.** نقصد بالصيانة الشاملة تعديل النظام وإعادة تصميمه ليلبي متطلبات جوهرية جديدة

- **إخراج النظام من الخدمة:** قد تصبح صيانة النظام الدائمة غير ممكنة وقد نضطر لإخراج النظام من الخدمة لأسباب لا تتعلق أساساً بفائدته، فقد يبقى النظام مفيداً لكن صيانته غير ممكنة، هناك أربعة أسباب قد تدفعنا لإخراج النظام من الخدمة:
  1. تجاوز التعديلات المقترحة لإمكانيات الصيانة الشاملة المباشرة
  2. فقدان السيطرة على النظام وفقدان إمكانية التنبؤ بآثار التعديلات
  3. نقص التوثيق اللازم لبناء التوسعات المستقبلية
  4. ضرورة استبدال البنية العتادية والبرمجية وعدم توفر أي طريقة لتهجير النظام ومعطياته

التخطيط في دورة حياة المنتج البرمجي

- **تخطيط المشروع:** يعرف تخطيط المشروع (Project Planning) بأنه النشاط المتعلق بتقدير أجزاء المشروع القابلة للإنجاز، وتكاليفه، والزمن اللازم لإنجازه، والمخاطر التي قد يتعرض لها، وملامحه العامة والموارد اللازمة له. كما يتضمن أيضاً اختيار طرق التطوير وإجرائياته وأدواته والمعايير التي ستُعتمد بالإضافة إلى تنظيم فريق العمل

- إن تخطيط المشروع هدف متغير، فهو ليس ثابتاً يوضع لمرة واحدة ولا يتغير بعد ذلك أبداً، بل تتطور خطط المشروع خلال مراحل دورة حياته، لكن ضمن إطار عمل يحكمه عدد قليل من القيود الثابتة

- **قيود التخطيط:**
  - يعتبر الزمن والكلفة من أهم القيود النمطية التي تحكم تخطيط المشاريع، فلكل مشروع زمن إنجاز محدد وموازنة دقيقة، ولذلك يعتبر التحقق من إمكانية تنفيذ المشروع ضمن القيود الزمنية والمالية الخطوة الأولى في عملية التخطيط، فإذا تبين أن المشروع قابل للإنجاز توثق القيود الأساسية ولا يمكن أن تعدل إلا عبر إجرائية صورية تبرر أسباب التعديل بدقة ووضوح.
- **تقييم جدوى المشروع:** يتم تقييم جدوى المشروع بعد أخذ مجموعة من العوامل بعين الاعتبار:
  1. **الجدوى العملية:** وهي تعيد دراسة المواضيع التي أخذت بالحسبان أساساً عند تعريف المشروع في تخطيط النظام، وهي تهتم بدراسة تأثير النظام المقترح على البنى التنظيمية وإجراءات العمل والأشخاص.
  2. **الجدوى الاقتصادية:** وهي تقدر تكاليف المشروع وفوائده (وتعرف أيضاً بتحليل الكلفة والفائدة).

المنارة للاستشارات
www.manaraa.com

3. **الجدوى التقنية:** وتهتم بتقييم واقعية الحل التقني المقترح ومدى توفر المهارات التقنية والخبرات والموارد.
4. **الجدوى الزمنية:** وتهتم بدراسة وتقييم الجدول الزمني للمشروع.

- لا يمكن معرفة وتقييم كل القيود مع بداية المشروع، فقد تُكتشف قيود إضافية خلال مرحلة دراسة المتطلبات وتؤثر على دراسات الجدوى، وقد تكون هذه القيود قانونية أو تعاقدية أو سياسية أو أمنية.

الاختبارات في دورة حياة المنتج البرمجي

- يعتبر الاختبار نشاطاً يغطي كل مراحل دورة حياة البرمجية، فهو ليس كما يعتقد البعض مرحلة مستقلة تلي مرحلة التحقيق، فتأجيل البدء بالاختبار إلى ما بعد التحقيق هو أسوأ من مجرد التأخير، إذ تصبح كلفة تصحيح أخطاء مراحل دورة الحياة الأولى باهظة.

- يجب التخطيط لأنشطة الاختبار بدقة، ويتطلب ذلك البدء بتعريف حالات الاختبار أو خطط الاختبار التي تعرف الخطوات الواجب اتخاذها لاختبار أجزاء المنتج أو النموذج البرمجي.

- يجب تعريف حالات الاختبار لكل مجتزأ وظيفي (حالة استخدام) ورد وصفه في وثيقة المتطلبات، ويؤدي الربط بين حالات الاختبار وحالات الاستخدام إلى رسم طريق واضح لاختبار المنتج إزاء متطلبات المستخدم.

- يقوم كل مطور، كسلوك طبيعي، باختبار منتجات عمله، لكنه لن يتمكن من رؤية كل ما يحيط بالمنتج البرمجي الذي عمل عليه. لذلك لابد من إدخال طرف آخر يجري اختبارات منهجية، وقد يكون هذا الطرف هو مجموعة ضمان جودة البرمجيات الموجودة في المؤسسة، والتي يجب أن تضم خيرة المطورين لديها على أن يكون عملهم الاختبار وليس التطوير. وتقع على عاتق هذه المجموعة مسؤولية ضمان جودة المنتج.

- يمكن البدء بإجراء الاختبارات انطلاقاً من مرحلة المتطلبات لتشمل كل أنواع الوثائق (بما فيها الرماز المصدري للبرامج) بإتباع ما نسميه مراجعات صورية (formal reviews)، وكلما أكثرنا من الاختبارات في مراحل التطوير الأولى كلما كانت كلفة التطوير أقل.
1. المراجعات الصورية هي لقاءات يُحضَّر لها بعناية وتستهدف جزءاً محدداً من التوثيق أو من النظام. يدرس المراجع وثيقة ما من الوثائق ليطرح أسئلة متنوعة تدرس في اللقاءات دون أن يعني ذلك طرح حلول مباشرة، بل يترك للمطور أن يدرسها لاحقاً، ويؤدي تعاون أعضاء الفريق في هذه المهمة إلى اكتشاف العديد من الأخطاء وتصحيحها في المراحل المبكرة.

- وبعد أن تصبح إصدارات المنتج البرمجي الأولى أو نماذجه الأولية جاهزة يمكن البدء بإجراء الاختبارات المتعلقة بالتنفيذ وهي نوعان:
2. اختبارات التوصيف (اختبار الصندوق الأسود): تتعامل اختبارات التوصيف مع البرنامج كصندوق أسود لا نعرف عنه شيئاً سوى أنه يقبل مدخلات معينة ليولد مخرجات معينة، ولذلك يزودنا البرنامج ببعض المدخلات وتحلل نتائج الخرج بحثاً عن وجود أخطاء، وتعتبر هذه الاختبارات هامة خصوصاً لاكتشاف المتطلبات غير المحققة أو المحققة بطريقة غير صحيحة.
3. اختبارات الرماز (اختبار الصندوق الأبيض أو الصندوق الشفاف): تتعامل اختبارات الرماز مع منطق البرنامج لتستنبط منه المدخلات اللازمة لاختبار مسارات التنفيذ المختلفة، وهي اختبارات مكملة لاختبارات التوصيف، وتساعدان معاً في اكتشاف أنواع مختلفة من الأخطاء.

- كما يقتضي التطوير التزايدي مكاملة تزايدية لمجتزآت البرمجية فهو يقتضي أيضاً اختبارات تزايدية أو تدريجية، إذ يجب إعادة اختبار أي مجتزأ بعد توسيعه وزيادته لنضمن أن وظائفه القديمة مازالت محققة ولم تتأثر بالتوسع الجديد.
  - يمكن تدعيم الاختبارات التدريجية باستخدام أدوات تسمح بتسجيل مراحل وخطوات تخاطب المستخدم مع البرنامج

المنارة للاستشارات
www.manaraa.com

وإعادتها لاحقاً دون تدخل من المستخدم.

- تعترض الاختبارات التدرجية صعوبة أساسية، فالتطوير التزايدي لا يعني توسيع خوارزميات البرنامج فحسب، بل أيضاً توسيع (وتعديل) بنى المعطيات التي تستخدمها هذه الخوارزميات، وهذا يقتضي تعديل مجموعة معطيات الاختبار الأساسية، وبالتالي لن تبقى مقارنة نتائج الاختبارات المتتالية ذات معنى.

منهجيات تطوير البرمجيات

1- المنهجية المهيكلة

- **المنهجية المهيكلة:** انتشرت المنهجية المهيكلة لتطوير النظم في الثمانينات، وأصبحت قياسية إلى حد بعيد، وتستند هذه المنهجية إلى تقنيتين أساسيتين:
  1. مخططات تدفق المعطيات DFD (Data Flow Diagrams) لنمذجة الإجرائيات
  2. مخططات علاقات الكيانات ERD (Entity Relationship Diagrams) لنمذجة المعطيات

- **خصائص المنهجية المهيكلة:** تتميز المنهجية المهيكلة في التحليل والتصميم بعدد من الميزات التي لا ينسجم بعضها مع التوجهات الحديثة في هندسة البرمجيات:
  1. تأخذ المنهجية منحى تسلسلياً وتحويلياً أكثر منه تكرارياً وتزايدياً (أي أنها لا تسهل تسليم البرمجة عبر إصدارات تزايدية).
  2. تعطي المنهجية حلولاً غير مرنة تلبي وظائف العمل المعرفة مسبقاً ويصعب توسيعها تدريجياً في المستقبل.
  3. تفترض المنهجية أن التطوير يبدأ دوماً من الصفر ولا تدعم إعادة استخدام مكونات برمجية موجودة.

منهجيات تطوير البرمجيات

2- المنهجية غرضية التوجه

- انتشرت المنهجية غرضية التوجه في تطوير النظم في أعوام التسعينات، وقد وضعت المجموعة OMG (Object Management Group) معياراً لهذه المنهجية (لغة UML –Unified Modeling Language). تتمحور المنهجية غرضية التوجه حول المعطيات، وبصيغة أكثر تحديداً تعتمد هذه المنهجية على نماذج الصفوف، التي لا تتضمن في مرحلة التحليل أية عمليات بل صفات فقط (attributes).

- **ميزات المنهجية غرضية التوجه:** يستخدم المطورون المنهجية الغرضية لما تقدمه من مزايا تقنية جيدة كالتجريد والتغليف وإعادة الاستخدام والوراثة وتبادل الرسائل وتعدد الأشكال وغيرها، ويؤدي استخدام هذه المزايا التقنية إلى تسهيل إعادة استخدام الرماز والمعطيات وإلى اختصار أزمنة تطوير المشاريع، وزيادة إنتاجية المبرمجين وتحسين نوعية المنتج البرمجي وتسهيل فهمه وغيرها.

- **المشكلات الناجمة عن المنهجية غرضية التوجه:** تسمح المنهجية الغرضية بالتغلب على أهم العقبات التي تعترض المنهجية المهيكلة، لكنها تطرح بدورها عدداً قليلاً من المشكلات الجديدة:
  1. يجري التحليل على مستوى أعلى من التجريد، وتصبح الفجوة الدلالية بين المفهوم وتحقيقه هامة إذا كان تحقيق حل المخدم يعتمد على قاعدة معطيات علاقاتية، ومع أنه يمكن من حيث المبدأ إجراء التحليل والتصميم بطريقة تكرارية وتزايدية لكن قد يبلغ التطوير مرحلة التحقيق مما يتطلب التحويل إلى قاعدة معطيات علاقاتية، ويكون هذا التحويل أسهل إذا كانت منصة التحقيق قاعدة معطيات غرضية أو غرضية – علاقاتية.

  2. تصبح إدارة المشروع أكثر صعوبة. إذ يقيس المدراء تقدم التطوير عادة بتعريف بنى ونقاط علام وبقياس الأجزاء

المنارة للاستشارات
www.manaraa.com

المسلمة للزبائن. لكن الحدود بين مراحل التطوير في المنهجية الغرضية ليست واضحة ويتطور توثيق المشروع ويتغير باستمرار, ويمكن تجاوز هذه الصعوبة بتقسيم المشروع إلى مجتزآت صغيرة وإدارة تقدم الإصدارات التنفيذية لهذه المجتزآت (قد تكون بعض المجتزآت للتسليم وبعضها الآخر للاستخدام الداخلي أي لتطوير مجتزآت أخرى).

3. ثمة مشكلة هامة أخرى تعترض المنهجية الغرضية، وهي ناتجة عن تعقيد الحل الذي يؤثر بدوره على قابلية صيانة البرمجية وقابليتها للتوسيع.

المنارة للاستشارات
www.manaraa.com

القسم الثالث

دورة حياة المنتج البرمجي

ملخص:

سنعرض في هذه الجلسة لمحة عامة عن مفهوم تطوير البرمجيات من خلال التعرف إلى الخطوات التي تمر فيها دورة حياة المنتج البرمجي، إذ سنقوم أيضاً بدراسة مراحل دورة حياة البرمجيات، وسنركز في هذه الجلسة على دراسة النماذج المختلفة المستخدمة في تطوير البرمجيات، وعلى الاختلافات فيما بينها.

أهداف تعليمية:

سيتعرف الطالب في هذا الفصل على المفاهيم التالية:

- ما هو نظام المعلومات؟
- ما هي البرمجيات، وما هي خصائصها؟
- تصنيف البرمجيات وأنواعها:
  - تطبيقات نظم التشغيل
  - تطبيقات إدارة الأعمال
  - التطبيقات العلمية والهندسية
  - برمجيات الحاسب الشخصي
  - البرمجيات المضمّنه
  - تطبيقات نظم الزمن الحقيقي
  - تطبيقات الذكاء الصنعي
- دورة حياة البرمجيات
  - النموذج الشلالي
  - النموذج الحلزوني
  - نموذج الطراز البدئي
  - نموذج التطوير المتزامن
- مراحل دورة حياة البرمجيات:
  - التحليل
  - التصميم
  - التطبيق

المنارة للاستشارات
www.manaraa.com

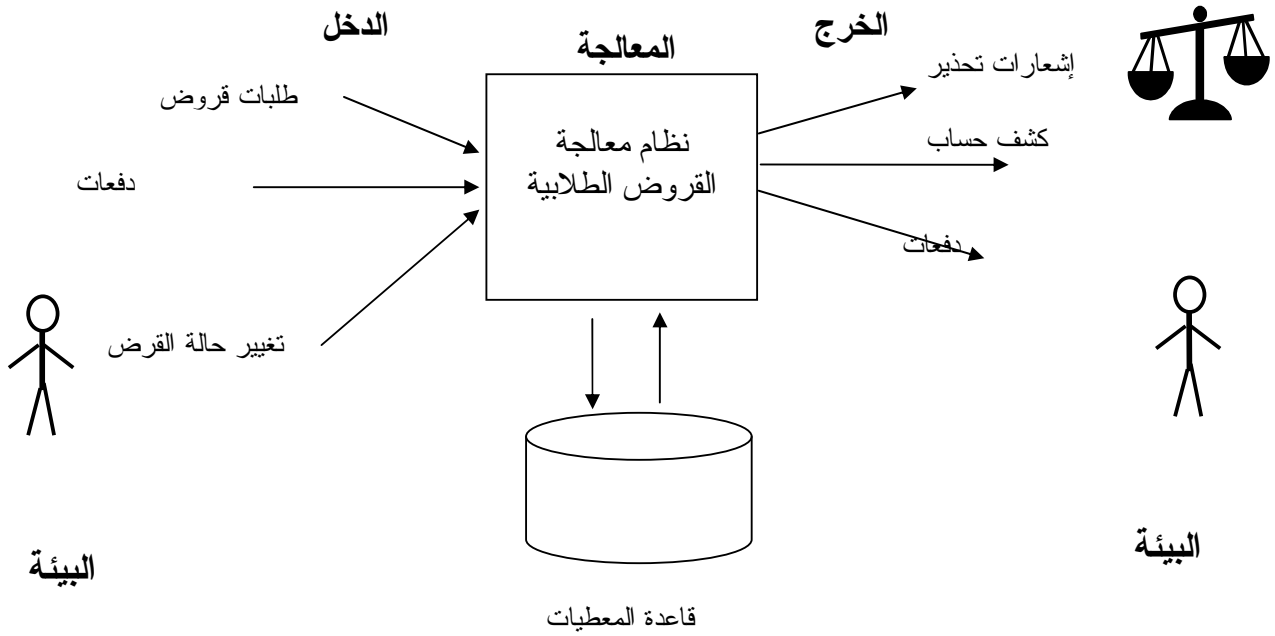
مقدمة

- سنناقش في هذه الجلسة مفهوم المنتج البرمجي وعلاقته بنظام المعلومات، سنتحدث عن مكونات نظم المعلومات بالإضافة إلى التعرف على دورة حياة نظام المعلومات أو دورة حياة المنتج البرمجي وذلك باعتباره جزءاً من نظام المعلومات أو أحد مكوناته
- سنقوم أيضاً بدراسة مراحل دورة حياة البرمجيات، وسنركز في هذه الجلسة على دراسة النماذج المختلفة المستخدمة في تطوير البرمجيات، وعلى الاختلافات فيما بينها.

نظام المعلومات

- تعريف:
  - يطلق اسم النظام على مجموعة المكونات المترابطة التي تعمل بعضها مع بعض لتحقيق أهداف معينة، والتي تترابط بدورها وتتفاعل مع البيئة المحيطة
  - يقوم نظام المعلومات بتلقي المعطيات من البيئة المحيطة به، ثم يقوم بمعالجتها وإصدار معطيات أخرى كخَرج مناسب، لتساهم بدورها في عمليات دعم واتخاذ القرارات
  - مثال: سنعرض فيما يلي مثالاً لنظام معلومات يُعنى بتقديم قروض طلابية:
    - تتكون بيئة هذا النظام من المُقرِضين والطلاب ووكلاء الحكومة
    - يقوم المُقرِضون بإدخال طلبات القروض المقبولة في النظام
    - يستلم الطلاب الدفعات من البنك بتواتر معين
    - يقوم النظام بإصدار كشوفات للطلاب لتسديد أقساط القرض بعد التخرج
    - يقوم النظام بإصدار إشعارات خاصة للحكومة في حال تأخر الطالب عن تسديد قسط القرض
  - تختلف مكونات نظم المعلومات وتتعدد، إذ لا تقتصر على قواعد المعطيات بل تتعداها لتشمل الأشخاص والإجرائيات ومعطيات الدخل ومعطيات الخرج والبرمجيات والعتاديات.

المنارة للاستشارات
www.manaraa.com



البرمجيات

- تعريف:
  - يطلق اسم البرمجية أو البرنامج على مجموعة التعليمات التي تؤدي وظيفة ما أو أداء محدد أثناء التنفيذ، أو هي عبارة عن الوثائق التي تصف أسلوب عمل برامج محددة وكيفية استخدامها

- خصائص البرمجيات:
  - تختلف البرمجيات عن العتاديات اختلافاً كبيراً، فعلى سبيل المثال تخضع البرمجيات لدورة حياة مغايرة لدورة الحياة التي تخضع لها العتاديات –كما سيمر معنا–، هذا بالإضافة إلى أن العتاديات عبارة عن مكونات فيزيائية في حين تشكل البرمجيات عناصر منطقية في نظام غير مادّي، فالبرمجيات تُطوّر ولا تُصنع، كما أن فترة إمكانية استخدام البرمجيات غير محدودة مقارنةً مع العتاديات، إذ أن البرمجيات لا تبلى ولا تتأثر بالبيئة المحيطة بها، مع العلم أنه يمكن أن يتم التخلي عن البرمجيات مع مرور الوقت وذلك بسبب التغيرات المرتبطة بجودة تلك البرمجيات وفعاليتها، أو العيوب التي يمكن أن تظهر أثناء استخدامها، خاصةً وأن عملية صيانة البرمجيات تعتبر عملية مكلفة للغاية، إذ لا يمكن مقارنتها مع عملية صيانة العتاديات

  - على الرغم من أن مبدأ إعادة استخدام البرمجيات الشهير، هو أحد أهم مبادئ هندسة البرمجيات وتطويرها، إلا أن ذلك لا يبرر إمكانية بناء البرمجيات وتجميعها كمجتزآت حسب الطلب انطلاقاً من مكونات محددة، إذ أن بناء وتصميم البرمجيات يعتبر وحدة متكاملة غير قابلة للتجزيء المطلق.

المنارة للاستشارات
www.manaraa.com

أنواع البرمجيات

- تختلف التطبيقات البرمجية وتتنوع، إذ يعتبر من الصعب جداً أن نوجد تصنيفات محددة لأنواع البرمجيات خاصةً مع اتساع مجال التطبيقات البرمجية وازدياد تعقيد البرمجيات، إلا أننا سنقوم من خلال هذه الشريحة بتحديد مجموعة من النقاط التي يمكن الاعتماد عليها كأصناف ممكنة لتطبيقات برمجية، ومنها:
  - تطبيقات نظم التشغيل:
    - وهي عبارة عن برمجيات مكتوبة لتخديم برمجيات أخرى، إذ تقوم برمجيات النظم بإدارة الموارد الحاسوبية وتأمينها لخدمة التطبيقات المختلفة بالشكل الأمثل، وتتميز هذه التطبيقات بتعقيدها وتعدد واجهاتها، إذ يمكن أن تدعم إمكانيات التشارك في العتاديات أو إمكانيات العمل على التوازي من قبل المستخدمين
  - تطبيقات إدارة الأعمال:
    - وهو المجال الأوسع في أنواع البرمجيات المتاحة، إذ تم تطوير النظم المتفرقة كنظم الأرشفة أو المستودعات أو الرواتب أو إدارة الموارد ... إلى تطبيقات نظم معلومات إدارية، بحيث يمكنها أن تقوم بالنفاذ إلى قواعد معطيات ضخمة لتقوم بتخديم تلك التطبيقات الفرعية
  - التطبيقات العلمية والهندسية:
    - وهي البرمجيات التي تُستخدم لإجراء عمليات تصميم مختلفة أومحاكاة لنظم محددة في المجال العلمي والهندسي
  - برمجيات الحاسب الشخصي:
    - وهي التطبيقات التي نجدها على الحواسيب الشخصية، والتي تقوم بالعديد من المهمات المختلفة كمعالجة النصوص أو الجداول الحسابية أو إدارة المعطيات أو إدارة الوسائط المتعددة أو تطبيقات الترفيه أو التطبيقات الشخصية أو تطبيقات إدارة الأعمال أو طرائق النفاذ عن بعد وغيرها...الخ
  - البرمجيات المضمّنه:
    - وهي عبارة عن التطبيقات المضمنة في ذواكر للقراءة فقط، وتستخدم للتحكم بوظائف محدودة كتشغيل جهاز محدد أو ضبط درجة حرارة معينة أو غيرها...
  - تطبيقات نظم الزمن الحقيقي:
    - وهي التطبيقات التي تقوم بمراقبة أحداث معينة تجري في الوسط الحقيقي المحيط، وتتميز هذه التطبيقات بضرورة تحقيق زمن استجابة محدد ينبغي عدم تجاوزه وإلا يفشل النظام
  - تطبيقات الذكاء الصنعي:
    - وهي عبارة عن تطبيقات ذات خوارزميات محددة تقوم بمعالجة مسائل معقدة، كالنظم الخبيرة والشبكات العصبونية.

المنارة للاستشارات
www.manaraa.com

تطوير البرمجيات

نماذج من دورة حياة البرمجيات – النموذج الشلالي

- تتنوع الطرائق والأساليب المُستخدمة في توصيف دورة حياة التطبيقات البرمجية وتطورها، إذ تمر بعدة أطوار أو مراحل يمكن أن تتراوح ما بين ثلاثة إلى عشرين مرحلة جزئية

- سنقوم باستعراض دورة الحياة التقليدية المعروفة باسم النموذج الشلالي الذي يتكون من مجموعة مراحل متتالية تكون نتيجة كل منها دخلاً للتالية، كما سنقوم في الشرائح التالية باستعراض نماذج أخرى تصف دورة حياة البرمجيات

- يُعتبر النموذج الشلالي نموذجاً مرجعياً لوصف دورة حياة البرمجيات، وهو يتميز بحدود مبهمة بين المراحل المكوّنة له، بالإضافة إلى وجود إمكانيات تنقل تراجعية بين تلك الأطوار
- يتكون النموذج الشلالي من خمسة مراحل، وهي:

1- مرحلة الاستطلاع التمهيدي:

ينتج عن هذه المرحلة بيان بالمشاكل التي تعترض النظام ودراسة جَدوى
بحيث يتضمن البيان معلوماتٍ عن كافة الأهداف والقيود بالإضافة إلى تحديد مجال النظام بشكل عام، في حين يتحدد من خلال دراسة الجدوى تكاليف النظام والفائدة منه، بحيث يتم الانتقال إلى المرحلة التالية ما أن تتم الموافقة على دراسة الجدوى

2- مرحلة تحليل النظام:

ينتج عن هذه المرحلة متطلبات تصف الارتباط والتفاعل ما بين الإجراءات والمعطيات والبيئة المحيطة
يتم الاعتماد في هذه المرحلة على تقنيات وأدوات تخطيط ورسم، وذلك لتوثيق كافة الارتباطات المختلفة ما بين الإجراءات والمعطيات والبيئة، بحيث يتم عرض المتطلبات بعد دراسة النظام من قبل المستخدمين وذلك بعد إجراء عدّة لقاءات واجتماعات، (سنقوم في الشرائح التالية بتوضيح أنواع الاجتماعات الممكن عقدها وكيفية إدارتها)

3- مرحلة تصميم النظام:

ينتج عن هذه المرحلة خطّة ملائمة لتنفيذ المتطلبات الموصّفة في المرحلة السابقة
تهتم خطة التنفيذ بشكل رئيسي بكيفية اختيار الطريقة المثلى لاستخدام الموارد في ضوء القيود المفروضة

4- مرحلة تنفيذ النظام:

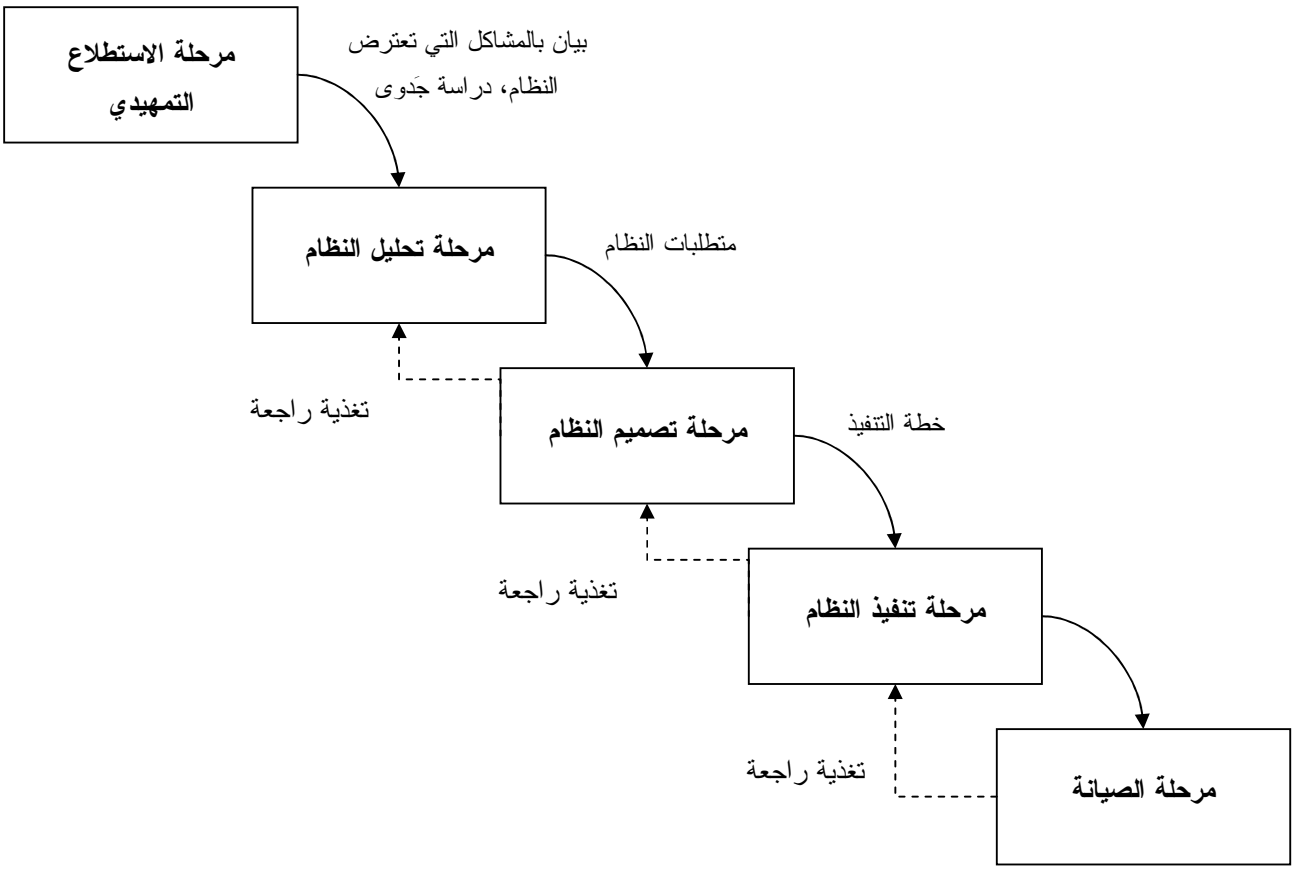
يتم في هذه المرحلة إنشاء الرماز التنفيذي للنظام الذي قمنا بدراسته في المراحل السابقة، كما يتم فيها بناء قواعد
تهتم مرحلة تنفيذ النظام بترميز واختبار الخطط المطروحة في مرحلة تصميم النظام المعطيات وتوثيق المعلومات

5- مرحلة الصيانة:

يتم في هذه المرحلة إنجاز التطويرات والتصحيحات والتعديلات على نظام المعلومات الذي تم إنشاؤه
تختلف مرحلة الصيانة بشكل جذري عن المراحل الأخرى، إذ أنها تشتمل على عمليات من كافة المراحل السابقة
تنتهي مرحلة الصيانة عندما يصبح من الضروري تطوير نظام معلومات جديد لتنفيذ العمليات التي يعجز عنها النظام

المنارة للاستشارات
www.manaraa.com

الحالي، ولكن نظراً لارتفاع تكلفة تطوير نظم المعلومات، يمكن أن تستمر مرحلة الصيانة لعدة عقود.



المشاكل التي يتعرض لها النموذج الشلالي

- تم انتقاد طريقة النموذج الشلالي في بناء نظم المعلومات، ويعود ذلك لأسباب عديدة، إذ لا يتم تنفيذ النظام إلا بعد فترة طويلة جداً من المعالجة، فمع مرور الوقت تتغير المتطلبات وتتبدل، هذا بالإضافة إلى الميل إلى التنفيذ بشكل سريع ممـــا لا يـــسمح بتـــوفير الوقت الكافي من أجل إجراء عمليات التحليل والتصميم

- تم اقتراح العديد من المنهجيات للتخفيف من الصعوبات الناتجة عن النموذج الشلالي في بناء نظم المعلومات، كالنموذج الحلزونـــي أو نموذج الطراز البدئي أو النموذج التسايري (سنقوم في الشرائح القادمة بدراسة هذا النموذج بالتفصيل).

النموذج الحلزوني

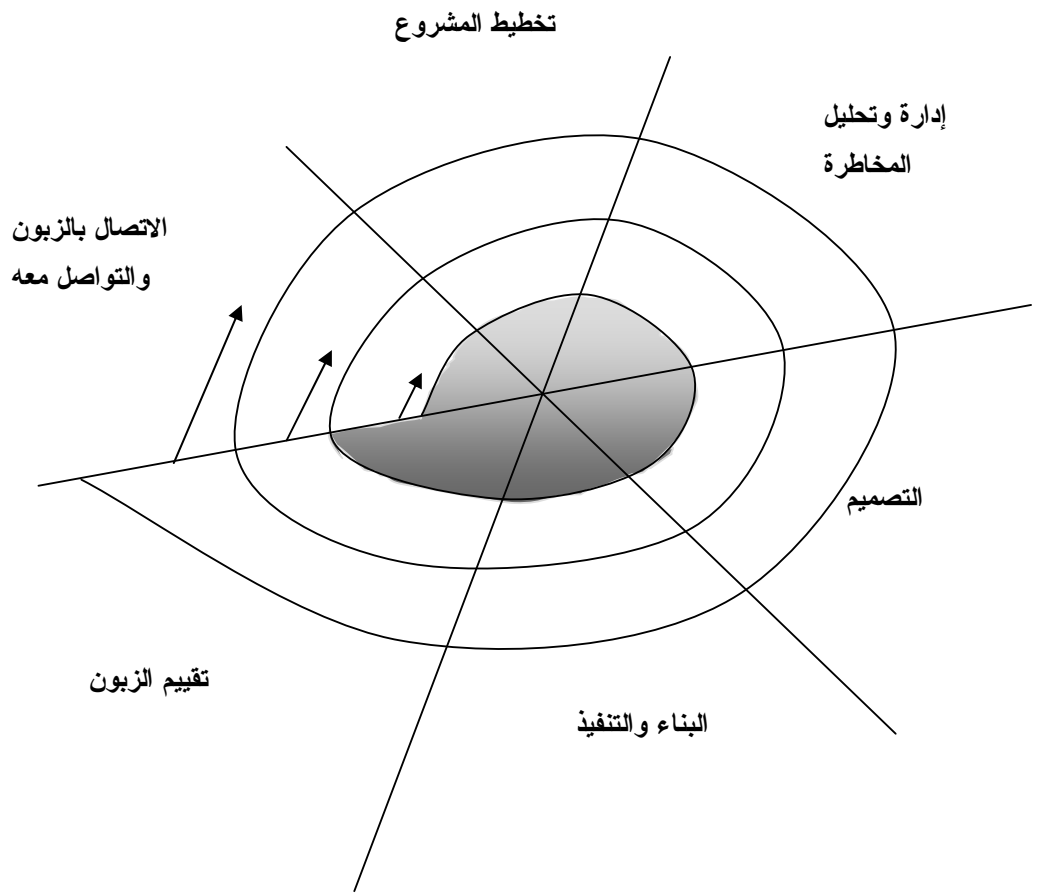
- يقترح النموذج الحلزوني تنفيذ مراحل دورة حياة النظام على شكل مجموعات جزئية منفصلة بحيث يتم تجميعها تراكمياً مع الزمن حتى تشكل بمجموعها النظام بالكامل، ويقدم هذا النموذج إمكانيات التطوير السريع لنسخ تزايدية من البرمجيات، بحيث يتم تطوير

المنارة للاستشارات
www.manaraa.com

البرمجيات اعتماداً على هذا النموذج من خلال سلسلة من الإصدارات التزايدية بحيث يتم استخدام كل إصدار في تطوير الإصـــدار الذي يليه

- يقسم مخطط النموذج الحلزوني في تطوير البرمجيات إلى عدّة مناطق أو أقاليم، تُصنّف عادةً كالتالي:
  - الاتصال بالزبون والتواصل معه:
    وهي المنطقة الخاصة بتأمين وسائل التواصل الفعّال ما بين مطوري النظام البرمجي وما بين الزبائن
  - تخطيط المشروع: وهي المنطقة الخاصة بتعريف الموارد ومتطلبات المشروع بالإضافة إلى التقدير الزمني ومعلومات أخرى ترتبط بالمشروع
  - إدارة وتحليل المخاطرة: وهو ما يتعلق بتقييم وإدارة المخاطرة التقنية والإدارية المرتبطة بالمشروع
  - التصميم: وهي المهمات المرتبطة بإمكانيات تمثيل التطبيق بطريقة أو أكثر
  - البناء والتنفيذ: وهي العمليات المرتبطة بمراحل بناء المنتج واختباره وتنصيبه وتأمين الخدمات ذات الصلة من خدمات الدعم الفني والمساعدة كأدوات التوثيق والتدريب
  - تقييم الزبون: وهي عبارة عن التغذية الراجعة التي يقدمها الزبون من تعليقات أو انتقادات أو تقييم للأعمال التي تم إنجازها في مراحل التصميم والبناء والتنفيذ

المنارة للاستشارات
www.manaraa.com



- ما أن تبدأ عملية تطوير المنتج البرمجي باستخدام النموذج الحلزوني حتى يبدأ فريق التطوير بالتحرك عبر مخطط النموذج الحلزوني باتجاه عقارب الساعة ابتداءً من المركز، إذ يمكن أن ينتج عن الدورة الأولى الخصائص الأولية للمنتج الذي نرغب بتطويره، كما يمكن استنتاج نموذج أولي من التطبيق بعد عدة دورات، وهكذا شيئاً فشيئاً حتى يتم الوصول إلى المنتج النهائي بعد أخذ كافة التعليقات التي تقدم بها الزبون عبر مراحل التطوير بعين الاعتبار

- يتميز النموذج الحلزوني بإمكانية استمراره حتى نهاية خدمة البرمجية التي استُخدم من أجل تطويرها، إذ يمكن، وفي أي وقت، أن يتابع هذا النموذج سير أعماله من النقطة التي توقف عندها أو من أي نقطة مناسبة، وذلك بغرض تطوير المنتج

- يُعتبر النموذج الحلزوني طريقة مناسبة وهامة في تطوير البرمجيات خاصة وأنه يحافظ على الطريقة التقليدية المقترحة في دورة حياة المنتج البرمجي ولكنه يقوم بتطبيقها بأسلوب تكراري وبطريقة مناسبة تعكس العالم الواقعي المحيط

مشاكل النموذج الحلزوني

- يعاني النموذج الحلزوني من مشاكل خاصة، فهو لا يعتبر الحل المطلق الذي يمكن استخدامه في تطوير البرمجيات، فعلى سبيل المثال، يصعب عادةً إقناع الزبون أثناء التعاقد معه، على أن هذه الطريقة في نمذجة وتحليل وتنفيذ البرمجيات هي طريقة قابلة

المنارة للاستشارات
www.manaraa.com

للتحكم، خاصةً وأن هذا النموذج يتطلب خبرة ملائمة فيما يتعلق بدراسة وتقدير المخاطرة المرتبطة به، فعدم اكتشاف وجود مخاطرة ما، قد يؤدي إلى مشاكل كبيرة أثناء تطوير المنتج

- يعتبر النموذج الحلزوني في تطوير البرمجيات، نموذجاً حديث العهد مقارنةً مع النموذج الشلالي أو النموذج التتابعي أو نموذج الطراز البدئي –الذي سنتحدث عنه لاحقاً–، ولم يُستخدم على نطاق واسع، خاصةً وأنه لا ينفع كثيراً مع التطبيقات أو النظم الصغيرة والمتوسطة، فعادةً ما يتم تطبيقه من أجل تطوير النظم الضخمة

نموذج الطراز البدئي

- يهتم نموذج الطراز البدئي بدراسة المتطلبات بشكل كبير، بحيث يتم من خلاله إنشاء واجهات بيانية وتقارير ورماز بشكل سريع وباستخدام أدوات خاصة بالتطوير المرئي، مما يسمح للمستخدمين بتزويد المطورين بمعلومات هامة وتغذية راجعة، خاصةً وأنه من الصعب استنتاج المتطلبات من الزبائن ما لم يقوموا بتجربة تطبيق بدئي كمثال، وهذا ما يقلل من المخاطر التي ترافق بناء نظم المعلومات.

- تعاني هذه الطريقة –وبشكل رئيسي– من أنها مكلفة وتستغرق زمناً إضافياً، إذ من الممكن ألا تتم الاستفادة من الواجهات البيانية التي صُمم الطراز البدئي على أساسها، بحيث يتم استخدامها لاستكشاف أفكار الزبائن، وبعد ذلك يتم الاستغناء عنها للانتقال إلى إتمام مراحل تطوير المنتج البرمجي.

نموذج التطوير المتزامن

- قبل أن نبدأ بدراسة وتعريف نموذج التطوير المتزامن، سنقوم أولاً بطرح الأسباب المباشرة التي أدت إلى ظهور هذا النوع من نماذج تطوير البرمجيات

- عانى مدراء المشاريع البرمجية من العديد من المشاكل أثناء تتبع أعمالهم من خلال أطوار المشروع المختلفة، وذلك في دورة الحياة التقليدية لمنتج برمجي محدد، وكانت المعاناة الرئيسية تنحصر في طريقة متابعة التطورات وملاحقتها، إذ كانوا لا يملكون التصوُّر المطلق عن وضع المشروع الحالي بكافة أجزائه ومكوناته، وكان من الصعب عليهم متابعة مجموعات معقدة جداً من الفعاليات التي تجري عبر مراحل عملية التطوير، خاصةً وأن المعنيين بتطوير المشروع يمكن أن يعملوا على عدة مسارات متوازية، سواء كان في التحليل أو التصميم أو التنفيذ أو الاختبار

- يتم من خلال نموذج التطوير المتزامن تنفيذ مراحل تطوير المنتج البرمجي بشكل متزامن ومتساير، بحيث لا يتم تقييد الفعاليات المختلفة، المشكلة للتطبيق، بمتوالية من الأحداث. وهذا النموذج قابل للتطبيق في الواقع العملي، في جميع أنواع البرمجيات، ويمكننا كذلك من خلاله أن نقدم تصوراً دقيقاً –نسبياً– عن الحالة الراهنة للمشروع قيد التطوير

- يمكننا أن نقوم بتمثيل نموذج التطوير المتزامن على شكل مخطط مكون من سلسلة من الفعاليات الرئيسية والمهمات والحالات الموافقة لها، بحيث يمكن أن يتم تطبيق كل فعالية في أي وقت. فعلى سبيل المثال، يمكن إجراء عمليات التواصل مع الزبون بأسلوب متزامن مع سير عمليات تحليل وتصميم النظام، كما يمكن أن يتم وضع فعاليات محددة في حالة انتظار لفعاليات أخرى قيد التطوير

المنارة للاستشارات
www.manaraa.com

مراحل دورة حياة التطبيقات البرمجية

- يخضع تطوير البرمجيات إلى دورة حياة، وهي عبارة عن مجموعة أنشطة مرتبطة بالمشروع الذي يتم تطويره، وتُعرِّف دورة الحياة، المراحل التي ينبغي على المنتج البرمجي أن يجتازها بدءاً من الاستطلاع الأولي وحتى النهاية

- يمكن تمثيل دورة حياة المنتج البرمجي بمستويات مختلفة من الدقة والتفصيل، بحيث يمكن أن يتم تقسيم الدورة إلى ثلاث مراحل رئيسية ، وهي:
  1. التحليل
  2. التصميم
  3. التحقيق

- تُعنى مرحلة التحليل بدراسة متطلبات النظام، إذ تسعى إلى تعريفها وتحديدها من خلال وضع المتطلبات والمعايير اللازمة لعملية التطوير، بالإضافة إلى التعرف على متطلبات النظام غير الوظيفية والقيود الأخرى التي يخضع لها النظام.

- أما بالنسبة إلى مرحلة التصميم، فتهتم على وجه الخصوص بوضع تصميم للمنتج، بحيث يتم ربط واجهة الاستخدام بأغراض قاعدة المعطيات، كما يجري في هذه المرحلة أيضاً دراسة وتوثيق العديد من العوامل التي تؤثر على إمكانية فهم النظام وصيانته وتوسيعه

- وفي مرحلة التحقيق يتم ترميز برامج الزبون التطبيقية وقواعد معطيات النظام التي تم توصيفها في مراحل سابقة، إذ تظهر في هذه المرحلة أهمية الإجرائيات التي تم تحليلها وتوصيفها، كما تكتسب مراجعة تطبيقات الزبون وقواعد معطيات المخدم، بمقارنتها مع نماذج التصميم، أهمية كبيرة في نجاح تسليم المنتج البرمجي.

وباختصار يدور التحليل حول ما يجب فعله، ويهتم التصميم بكيفية تحقيق ما هو مطلوب باستخدام التقانات المتاحة، ويُعني التحقيق بإنجاز المطلوب لغاية تسليم المنتج إلى الزبون.

أما في المستوى الأكثر تفصيلاً فتقسم دورة حياة المنتج البرمجي إلى المراحل التالية:

1. تحديد المتطلبات
2. توصيف المتطلبات
3. تصميم البنيان
4. التصميم التفصيلي
5. التحقيق
6. المكاملة
7. الصيانة

المنارة للاستشارات
www.manaraa.com

قد يضيف البعض مرحلتين إضافيتين هما التخطيط والاختبار، مع العلم أنه يمكن تطبيق هذين النشاطين على كافة مراحل دورة الحياة.

سنقوم من خلال الجلسات القادمة بدراسة المراحل الرئيسية من تطوير المنتج البرمجي بالتفصيل، إذ سنقوم في مرحلة تحليل النظام بدراسة كيفية تحديد وتوصيف المتطلبات، كما سنقوم في مرحلة تصميم النظام بدراسة الطرائق المتاحة في تصميم مكونات التطبيق البرمجي الذي نقوم بتطويره، هذا بالإضافة إلى التطرق إلى الأدوات المستخدمة لإنجاز هذه المراحل وتطوير المنتج عبر مراحل دورة حياته.
تحديد وتوصيف المتطلبات.

المنارة للاستشارات
www.manaraa.com

القسم الرابع والخامس

تحديد المتطلبات

ملخص:

سنقوم في هذه الجلسة بدراسة مرحلة تحديد المتطلبات وتوصيفها كمكونات أساسية من مرحلة التحليل في دورة حياة المنتج البرمجي، وسنشير إلى أهميتهما وضرورة إغنائهما بالوقت الكافي وذلك لتلافي المشاكل التي يمكن أن تنتج عن سوء التحليل في المراحل المتقدمة من عملية التطوير.

أهداف تعليمية:

سيتعرف الطالب في هذا الفصل على المهارات التالية:

- تحديد المتطلبات
- استنتاج المتطلبات
- الطرائق المستخدمة في استنتاج المتطلبات:
  - الطرائق التقليدية في استنتاج المتطلبات:
    - إجراء المقابلات
    - توزيع الاستبيانات
    - الملاحظة
    - دراسة وثائق العمل.
  - الطرائق الحديثة في استنتاج المتطلبات:
    - إعداد النماذج البرمجية البدئية
    - تطوير التطبيقات المشترك
    - تطوير التطبيقات السريع.
- تحليل المخاطر:
- أنواع المخاطر التي يمكن أن تتعرض لها المتطلبات:
  - المخاطر التقنية
  - مخاطر أداء
  - المخاطر الأمنية
  - مخاطر تكامل قواعد المعطيات
  - مخاطر إجرائية التطوير
  - المخاطر السياسية
  - المخاطر القانونية
  - مخاطر عدم الثبات.
- وثيقة المتطلبات
- مكونات وثيقة المتطلبات:
  - الجزء التمهيدي
  - خدمات النظام
  - قيود النظام

المنارة للاستشارات
www.manaraa.com

- مواد المشروع
- الملحقات.

- توصيف المتطلبات
- نماذج توصيف المتطلبات:
  - نموذج توصيف الحالة (مخططات الصفوف)
  - نموذج توصيف السلوك (مخططات حالات الاستخدام)
  - نموذج توصيف تغير الحالات (مخططات الحالات والانتقالات).

المنارة للاستشارات
www.manaraa.com

مقدمة

- تهتم مرحلة التحليل –وهي الخطوة الأولى في بناء المنتج البرمجي وتطويره– بتحديد وتوصيف متطلبات النظام التي ينبغي أن يتم تعريفها وتحديدها بشكل واضح وسليم قبل المباشرة ببناء المنتج، وفيها يتم وضع متطلبات عملية التطوير وجمعها وتحديدها وتوصيفها.

- يهدف تحديد المتطلبات إلى وضع تعريف موجز للمتطلبات الوظيفية وغير الوظيفية التي يتوقع المعنيون بالنظام أن يحققها

- يهتم توصيف المتطلبات بالتعبير عن هيكلية النظام الذي سيتم توصيفه، من خلال العديد من النماذج البيانية والنماذج الصورية، وذلك باستخدام أدوات مختلفة وتقنيات متعددة يمكن أن تعد أساساً لنجاح عملية النمذجة

- سنقوم فيما يلي بدراسة هاتين النقطتين بالتفصيل، وسنشير إلى أهميتهما وضرورة إغنائهما بالوقت الكافي وذلك لتلافي المشاكل التي يمكن أن تنتج عن سوء التحليل في المراحل المتقدمة من عملية التطوير.

تمهيد

- يخضع تطوير البرمجيات إلى عدة مراحل –كما مر معنا– ، وهي عبارة عن مجموعة أنشطة مرتبطة بالمشروع الذي يتم تطويره، والتي يمكن أن تُقسَم إلى ثلاث مراحل رئيسية ، وهي:
  1. التحليل
  2. التصميم
  3. التحقيق.

- تُعنى مرحلة التحليل بدراسة متطلبات النظام وتحديدها وتوصيفها من خلال وضع المتطلبات والمعايير اللازمة لعملية التطوير، بالإضافة إلى التعرف على متطلبات النظام غير الوظيفية والقيود الأخرى التي يخضع لها النظام.

يدور التحليل حول ما يجب فعله، ويهتم التصميم بكيفية تحقيق ما هو مطلوب باستخدام التقانات المتاحة، ويُعني التحقيق بإنجاز المطلوب لغاية تسليم المنتج إلى الزبون.

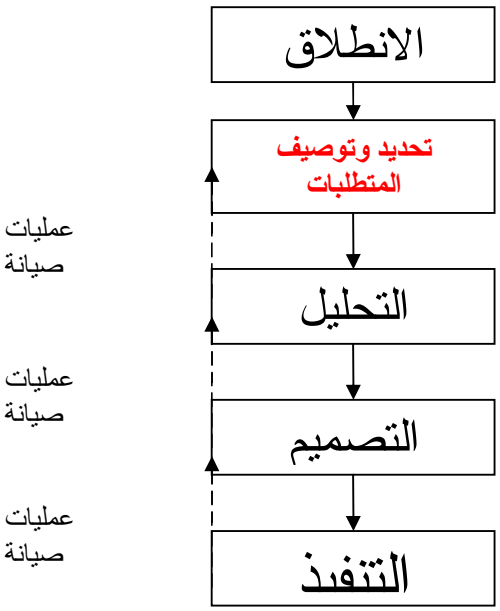
وإذا أردنا دراسة مرحلة التحليل بأسلوب أكثر تفصيلاً، يمكن أن نعبر عنها من خلال مرحلتين أساسيتين هما: مرحلة تحديد المتطلبات ومرحلة توصيف المتطلبات

سنقوم من خلال الشرائح القادمة بدراسة المكونات الرئيسية لمرحلة تحليل المنتج البرمجي، إذ سنقوم بدراسة كيفية تحديد وتوصيف المتطلبات، بالإضافة إلى التطرق إلى الأدوات المستخدمة لإنجاز هذه المهمات.

المنارة للاستشارات
www.manaraa.com

تحديد المتطلبات

- تعريف:
  - إن تحديد المتطلبات عمل ذو منحى اجتماعي نوعاً ما، إذ يعتمد على الخبرات الإدارية وعلى مهارات التواصل التي يمتلكها فريق التطوير
  - تعتبر هذه المرحلة نقطة انطلاق المشاريع البرمجية، إذ ينبغي فيها أن يتم تحديد كافة المتطلبات –وظيفية كانت أم غير وظيفية– التي يمكن أن تساهم في عملية التطوير، كما ينبغي أن يتم تحديد تلك المتطلبات بتمعن شديد، وذلك لأن إغفال متطلبات الزبون أو إساءة تفسيرها أو عدم اكتشافها يمكن أن يؤدي إلى زيادة تكاليف إجرائية التطوير، بحيث تظهر هذه النتائج السلبية بوضوح في المراحل المتقدمة



يخضع المخطط السابق والذي يعبر عن تطور نظم المعلومات, إلى تغيرات متعددة في مراحل التطبيق الواقعي، إذ من الممكن أن نرتكب الأخطاء من جهة، كما يمكن أن تتغير بعض متطلبات الزبون أثناء تطوير النظام، ونظراً للتكلفة المتزايدة الناتجة عن التعديلات التي يتم تطبيقها على النظام، خاصةً عندما تكون تلك التعديلات في المراحل المتقدمة، فلابد لنا أن نولي مرحلة تحديد وتوصيف المتطلبات الوقت

المنارة للاستشارات
www.manaraa.com

الكافي، وذلك لتجنب القدر الأكبر من التكاليف الإضافية ولتلافي التأخير المحتمل في عملية تطوير المنتج البرمجي

سنقوم من خلال الشرائح التالية بدراسة الطرائق المختلفة المستخدمة في استنتاج المتطلبات من خلال الحوار مع الزبون والتحقق من صلاحية متطلباته، إلى جانب عرض بعض المبادئ الأساسية في إدارة المتطلبات، التي تعنى بإدارة التغيرات وكشفها.

مبادئ تحديد المتطلبات

- تعرّف المتطلبات الخدمات المتوقعة من النظام والتي يفترض أن يوفرها، بالإضافة إلى القيود التي يجب أن يخضع لها:
  - يمكن توزيع الخدمات المتوقعة من النظام على مجموعات منها ما يوصّف نطاق النظام ومنها ما يوصِّف متطلباته الوظيفية ومتطلباته من المعطيات
  - يمكن تصنيف القيود التي يجب أن يخضع لها النظام تبعاً لفئات مختلفة، كمتطلبات الشكل والمظهر الخارجي ومتطلبات الأداء والأمن وغيرها.

- تُستخلص المتطلبات من الزبائن بشكل مباشر، من مستخدمي أو مالكي النظام، ويشرف المطور، أو محلل النظم، بشكل مباشر على عملية استنتاج المتطلبات تلك، كما يمكن أن يستخدم المحلل العديد من التقنيات بدءاً من إجراء مقابلات تقليدية مع الزبائن وانتهاءً ببناء نموذج أولي للنظام ليساعد على اكتشاف المزيد من المتطلبات.

- يجب أن تخضع المتطلبات التي جرى استنتاجها لعملية تحليل دقيقة بهدف التخلص من المتطلبات المكررة والمتناقضة، وقد يتطلب هذا مراجعة المتطلبات وإعادة التفاوض مع الزبائن.

- بعد أن يُظهِر الزبائن رضاهم وكفايتهم، يتم تعريف وتحديد المتطلبات وذلك من خلال تصنيفها وترقيمها لتظهر حسب أولويتها في وثيقة المتطلبات التي يجب أن تُعَّد وفقاً لقالب تختاره المؤسسة لتوصيف متطلباتها. وعلى الرغم من كون وثيقة المتطلبات عبارة عن وثيقة سردية، إلا أنه يفضل أن تحوي مخططاً يبين نموذج صفوف العمل.

- إن متطلبات الزبون هي هدف متغير، وللتعامل مع هذه المتطلبات يجب أن نكون قادرين على إدارة تلك التغيرات، وتتضمن إدارة المتطلبات أنشطة عديدة، منها مثلاً توقع تأثير تغير بعض المتطلبات على بقية النظام.

استنتاج المتطلبات

- تتنوع الطرائق والأساليب المُستخدمة في استنتاج المتطلبات، وتقع هذه المهمة على عاتق شخص يمكن أن يتم تعريفه على أنه محلل الأعمال، الذي يسعى إلى اكتشاف متطلبات النظام وذلك باستشارة الزبائن أو خبراء من مجال الأعمال قيد التطوير

- قد يمتلك محلل الأعمال –في بعض الحالات– خبرة كافية في مجال العمل الذي تجري نمذجته بحيث يمكنه الاستغناء عن مساعدة الخبراء في بعض المجالات، وتعتبر معرفة المحلل بقواعد العمل الذي يتم جمع وتحديد متطلباته قيمة مضافة لعملية التحليل يمكن أن تساهم إيجابياً في تسهيل آلية التواصل مع الزبائن

المنارة للاستشارات
www.manaraa.com

- تشكل المتطلبات التي يتم استنتاجها من قبل الخبراء، قاعدة معارف في مجال العمل، فهي تصور قواعد العمل المستقلة عن الزمن، والتي يمكن تطبيقها على معظم المؤسسات والأنظمة، أما المتطلبات التي تُستنتج من الزبائن فيُعبر عنها بحالات الاستخدام التي تتخطى المعارف الأساسية لتصور طريقة أداء الأعمال التي تجري في هذه المؤسسة بالذات (سنقوم في الجلسات القادمة بدراسة كيفية استنتاج حالات الاستخدام واستخدامها بالتفصيل).

الطرائق المستخدمة في استنتاج المتطلبات

- تُصنَّف الطرائق المستخدمة في استنتاج المتطلبات عادةً ضمن مجموعتين رئيسيتين، وهما:
  - الطرائق التقليدية في استنتاج المتطلبات
  - الطرائق الحديثة في استنتاج المتطلبات

- تتناسب فاعلية الطرائق التقليدية في استنتاج المتطلبات مع درجة خطورة المشروع تناسباً عكسياً، فكلما ارتفعت خطورة النظام ازدادت صعوبة تحقيقه، وبالتالي لا تعتبر الطرائق التقليدية البسيطة ومنخفضة الكلفة كافية لمثل هذه المشاريع

- من الطرائق التقليدية المتبعة في استنتاج المتطلبات: إجراء المقابلات وتوزيع الاستبيانات والملاحظة ودراسة وثائق العمل (سنقوم بشرح كل من هذه الطرائق في الشرائح التالية).

- تستخدم الطرائق الحديثة عادة عندما تكون درجة خطورة المشروع عالية، والتي تنتج بدورها عن عوامل عديدة من بينها عدم وضوح الأهداف، وعدم وجود توثيق للإجراءات، وعدم استقرار المتطلبات، وانخفاض خبرات المستخدمين، وعدم خبرة المطورين، وعدم التزام المستخدمين بشكل كاف، وغيرها
- تعطي الطرائق الحديثة أهمية أكبر لاستكشافات المتطلبات لكنها تحتاج بالمقابل إلى جهد وكلفة أكبر، لكن تبقى المحصلة على المدى الطويل أفضل.

- من بين الطرائق الحديثة في استنتاج المتطلبات: إعداد النماذج البرمجية الأولية البدئية، وتطوير التطبيقات المشترك وتطوير التطبيقات السريع (سنقوم بشرح كل من هذه الطرائق في الشرائح التالية).

الطرائق التقليدية في استنتاج المتطلبات

الاستبيانات

- تعتبر الاستبيانات طريقة فعالة لجمع المعلومات من عدة زبائن، وتستخدم الاستبيانات عادة إلى جانب المقابلات ولا تشكل بديلاً لها، إلا إذا كانت أهداف المشروع مفهومة جيداً ومستوى الخطورة فيه منخفض

- إن ما نحصل عليه من الاستبيانات أقل عموماً مما نحصل عليه من المقابلات، إذ لا يمكن عند استخدام الاستبيانات توضيح الأسئلة أو الأجوبة الممكنة

- إن لاستخدام طريقة الاستبيانات في استنتاج المتطلبات إيجابيات وسلبيات، فهي إيجابية لأن من سيجيب على الأسئلة يملك الوقت الكافي لمقارنة الإجابات الممكنة ويستطيع أن يحافظ على بقائه مجهولاً، وهي سلبية لأنها لا تعطيه الفرصة لاستيضاح الأسئلة،

المنارة للاستشارات
www.manaraa.com

لذلك ينبغي عادةً تجنب الأسئلة مفتوحة النهاية في الاستبيانات وأن تبقى معظم الأسئلة مغلقة النهاية

- يمكن أن تطرح ثلاثة أنواع من الأسئلة في الاستبيانات، وهي:
  - أسئلة مع خيارات متعددة: حيث يجب على المجيب أن ينتقي إجابة أو أكثر من مجموعة من الإجابات المقترحة مع السؤال، كما يجب السماح له بكتابة تعليق إضافي
  - أسئلة ترتيب درجة الموافقة: حيث يعبر المجيب عن رأيه ببيان ما، وتكون الإجابات عادة، موافق تماماً، موافق، حيادي، غير موافق، غير موافق إطلاقاً، ولا أعلم
  - أسئلة مع إجابات مرتبة: حيث ترتب الإجابات المرافقة للسؤال بأرقام تسلسلية، أو بنسب مئوية أو بوسائل ترتيب مشابهة.

الطرائق التقليدية في استنتاج المتطلبات

الملاحظة

- قد يجد محلل الأعمال في بعض الحالات صعوبة في الحصول على معلومات كاملة من المقابلات والاستبيانات، فقد لا يكون الزبون قادراً على إيصال معلومات للمحلل بفاعلية وقد لا يملك إلا معرفة جزئية بإجرائية العمل، وفي مثل هذه الحالات قد تكون الملاحظة تقنية فعالة لمعرفة الحقائق.

- ويمكن أن تأخذ الملاحظة إحدى صفتين:
  - ملاحظة منفعلة:

وفيها يراقب المحلل أنشطة العمل دون أن يقاطعها ودون أن يتدخل فيها مباشرة

  - ملاحظة فاعلة:

وفيها يشارك محلل الأعمال في أنشطة العمل ليتحول إلى جزء فعال في الفريق.

- ولكي تعطي المراقبة معلومات حقيقية يجب أن تدوم لفترة طويلة، كما يجب أن تتم في أوقات مختلفة، وفي ظروف عمل متباينة، وتبقى الصعوبة الأساسية هنا أن الناس يميلون للتصرف بطرق مختلفة أثناء مراقبتهم، ويميلون خصوصاً إلى العمل تبعاً للإجراءات والقواعد الرسمية الصورية. وهذا يشوه الحقيقة بإخفاء الاختزالات التي قد يجريها العامل سواء كانت سلبية أم إيجابية.

الطرائق التقليدية في استنتاج المتطلبات

دراسة الوثائق

- تعتبر دراسة الوثائق والأنظمة البرمجية –إن وجدت– تقنية لا غنى عنها لاستنتاج المتطلبات في النظام الذي يتم تطويره، ومن أهم ما يجب التركيز عليه إذا ما وُجد نظام قيد الاستخدام، هو سجل الأخطاء وطلبات التعديل

- تتضمن الوثائق التي ينبغي دراستها، استمارات عمل حقيقية وإجراءات العمل والأنظمة الداخلية وخطط العمل ومخططات البنية التنظيمية والعلاقات داخل المؤسسة أو خارجها ومحاضر الاجتماعات وسجلات المحاسبة وشكاوى الزبائن وغيرها.

- تتضمن الاستمارات والتقارير التي يجب دراستها: شاشات الاستخدام والتقارير إلى جانب التوثيق المرافق، أي أدلة عمل النظام أو

المنارة للاستشارات
www.manaraa.com

دليل المستخدم أو التوثيق التقني أو نماذج تحليل وتصميم النظام.

الطرائق الحديثة في استنتاج المتطلبات
النماذج البرمجية البدئية

- تعتبر النماذج البرمجية البدئية أو ما يُعرف باسم نمذجة الطراز البدئي prototyping من أكثر الطرائق الحديثة استخداماً، إذ تبنى النماذج البرمجية البدئية ليعاين الزبائن النظام، أو جزءاً منه، بهدف معرفة انطباعاتهم.
- يعتبر النموذج البدئي نظاماً توضيحياً، وهو نموذج عمل سريع وناقص، يتضمن واجهة استخدام بيانية ويحاكي سلوك النظام إزاء أحداث مختلفة، وتثبت المعلومات المحتواة في واجهات الاستخدام ضمن برنامج النموذج الأولي بدلاً من الحصول عليها ديناميكياً من قاعدة المعطيات
- تعتبر النمذجة البدئية حالياً عنصراً أساسياً في عملية تطوير البرمجيات، وذلك نظراً لتعقيد واجهات الاستخدام البيانية الحديثة وتنامي توقعات الزبائن، وبفضلها أصبح بالإمكان تقدير جدوى النظام وفائدته قبل البدء بتحقيقه فعلياً.
- تشكل النماذج البدئية عموماً طريقة فعالة جداً لاستنتاج المتطلبات التي يصعب الحصول عليها من الزبائن بالوسائل الأخرى، ونصادف هذه الحالة غالباً عند تطوير أنظمة تحقق وظائف أعمال جديدة، كما نصادف هذه الحالة أيضاً عندما تظهر تناقضات بين المتطلبات أو عندما تظهر مشاكل في التواصل بين الزبائن والمطورين.

- يوجد نوعان من النماذج البدئية:

النموذج الأولي الذي يُهمل كلياً بعد اكتمال استنتاج المتطلبات، والنموذج الذي يُحتفظ به بعد استنتاج المتطلبات ويُستخدم لإعداد المنتج النهائي، بحيث يهدف الأخير إلى تسريع تسليم المنتج، ويركز عادة على المتطلبات الأكثر وضوحاً بحيث يمكن تسليم الإصدار الأول من المنتج بسرعة حتى لو كان غير كامل وظيفياً.

ملاحظة:

يُفضل عادةً استخدام النماذج البدئية المعدة للإهمال، وذلك لتجنب مخاطر بقاء حلول غير فعالة أو إجراءات سريعة وناقصة في المنتج النهائي، لكن تطور ومرونة أدوات إنتاج البرمجيات الموجودة حالياً قد أضعفت أهمية هذه الحجة فبوجود إدارة جيدة للمشروع ليس ثمة ما يمنع من التخلص من الحلول غير الفعالة الموجودة في النماذج البدئية.

الطرائق الحديثة في استنتاج المتطلبات
تطوير التطبيقات المشترك

- تعتمد هذه الطريقة على استنتاج المتطلبات من خلال إقامة ورشة عمل أو أكثر تجمع كل المعنيين بالمشروع من زبائن ومطورين
- يمكن أن يتم تطبيق هذه الطريقة بعدة أشكال، هذا وتعرض العديد من الشركات الاستشارية خدمة تنظيم وإجراء جلسات التطـــوير المشترك
- يمكن أن يستغرق الاجتماع عدة ساعات أو عدة أيام أو حتى أسبوعين، ويجب ألا يتجاوز عدد المشاركين فيه 25 إلى 30 شخصاً

المنارة للاستشارات
www.manaraa.com

- يتم تصنيف المشاركين في اجتماع تطوير التطبيقات المشترك كما يلي:
  - القائد: وهو الشخص الذي يدير الاجتماعات ويضبطها، ويتمتع عادة بمهارات تواصل ممتازة، وهو ليس معنياً بالمشروع، ويمتلك معرفة جيدة بمجال العمل لكن لا يمتلك بالضرورة معرفة جيدة بتطوير البرمجيات
  - المقرر: وهو الشخص الذي يسجل جلسات التطوير المشترك على الحاسوب، ويجب أن يمتلك مهارات طباعية ومعرفة جيدة بتطوير البرمجيات. ويمكن للمقرر أن يستخدم الأدوات المساندة في هندسة البرمجيات لتوثيق الجلسات وتطوير نماذج بدائية للعمل
  - الزبائن (المستخدمون والمدراء): وهم المشاركون الرئيسيون الذين يتواصلون فيما بينهم ويناقشون المتطلبات ويأخذون القرارات ويحددون أهداف المشروع
  - المطورون: وهم محللو الأعمال وأعضاء آخرون من فريق التطوير، وهم يصغون أكثر مما يتكلمون، فهم موجودون في الاجتماع بهدف البحث عن الحقائق وجمع المعلومات وليس بهدف السيطرة على العملية.

- تستفيد طريقة التطوير المشترك من القيمة المضافة الناتجة عن العمل كمجموعة، والتي تعطي غالباً حلولاً أفضل، فالمجموعة تزيد الإنتاجية، وتتعلم أسرع، وتصل إلى أحكام أكثر نضجاً، وتحذف أخطاء أكثر، وتأخذ قرارات أخطر، وتشد اهتمام المشاركين نحو المواضيع الأكثر أهمية، وتكامل جهود الأفراد.

الطرائق الحديثة في استنتاج المتطلبات

تطوير التطبيقات السريع

- يعتبر أسلوب تطوير التطبيقات السريع منهجاً في تطوير البرمجيات بحيث يهدف إلى تسليم حلول النظام بأقصى سرعة ممكنة وذلك بغض النظر عن الجودة التقنية

- تجمع هذه الطريقة بين خمس تقنيات، وهي:
  - إنشاء النماذج البدئية
  - الأدوات المساندة في هندسة البرمجيات
  - أخصائيين يملكون أدوات متقدمة وهم في الواقع فريق التطوير المؤلف من أفضل المحللين والمصممين والمبرمجين الذين تستطيع المؤسسة توظيفهم. يعمل الفريق تحت قيود زمنية دقيقة ويبقى مع المستخدمين في الموقع نفسه.
  - التطوير المشترك التفاعلي
  - التقييد الزمني، وهو طريقة في إدارة المشاريع تفرض على الفريق أن ينجز المشروع خلال فترة زمنية ثابتة، بحيث تمنع هذه الطريقة امتداد نطاق المشروع.

- تعتبر طريقة التطوير السريع مناسبة جداً للعديد من المشاريع الصغيرة

- لكن بالمقابل تعترضها العديد من المشكلات، نذكر منها:
  - عدم انسجام تصاميم واجهات الاستخدام البيانية
  - الوصول إلى حلول تخصصية بدلاً من حلول عامة تسهل إعادة استخدام البرمجيات
  - الحصول على توثيق ضعيف أو ناقص

المنارة للاستشارات
www.manaraa.com

- صعوبة صيانة البرمجيات وتوسيعها تدريجياً

تحليل المخاطر

- ينبغي بعد الانتهاء من عملية تحديد واستنتاج المتطلبات، وقبل الانتقال إلى أي مرحلة أخرى أو تبني أي عمل آخر أن يتم إخضاع تلك المتطلبات إلى عملية تحليل للمخاطر وتحديد للأولويات

- يتم من خلال عملية تحليل المخاطر، تحديد المتطلبات التي يعتقد أنها ستضع صعوبات في وجه عملية التطوير. أما تحديد الأولويات فيسهل إعادة تعريف نطاق المشروع عند حدوث تأخير في عملية تطويره.

- تُصنّف المخاطر التي يمكن أن تتعرض لها المتطلبات ضمن الفئات التالية:
  - مخاطر تقنية، أي أن يكون تحقيق المتطلبات صعباً تقنياً
  - مخاطر أداء، أي أن يؤثر تحقيق المتطلبات سلبياً على زمن استجابة النظام
  - مخاطر أمنية، أي أن يعرِّض تحقيق المتطلبات النظام لاختراقات أمنية
  - مخاطر تكامل قواعد المعطيات، أي أن تؤدي المتطلبات إلى عدم انسجام في المعطيات
  - مخاطر إجرائية التطوير، وذلك عندما يحتاج تحقيق المتطلبات لاستخدام طرائق تطوير غير تقليدية لا يألفها المطورون
  - مخاطر سياسية، أي أن يصعب تحقيق المتطلبات لاعتبارات سياسية داخلية
  - مخاطر قانونية، أي أن تخرق بعض المتطلبات القوانين السارية المفعول، أو التعديلات المتوقعة على القوانين
  - مخاطر عدم الثبات، كأن تسبب المتطلبات حدوث تبدلات وتغيرات مستمرة طيلة فترة إجرائية التطوير.

وثيقة المتطلبات

- تشكل وثيقة المتطلبات الناتج النهائي الملموس لمرحلة تحديد المتطلبات، وتكتب معظم المؤسسات هذه الوثيقة تبعاً لقالب محدد مسبقاً يعرف بنية هذه الوثيقة ونمطها

- تتكون وثيقة المتطلبات من بيانات المتطلبات التي يمكن تصنيفها كمتطلبات وظيفية ومتطلبات معطيات، وفي بيانات قيود

- وينبغي أن تشير وثيقة المتطلبات إلى مفاتيح المشروع الأساسية، والتي تذكر عادة في بداية الوثيقة ويعاد ذكرها في نهايتها

- تحتوي الوثيقة على معلومات عن الهدف من المشروع بالإضافة إلى الأشخاص ذوي الصلة به والقيود الرئيسية المتعلقة بالمشروع. وفي نهاية الوثيقة تتم مناقشة المواضيع الأخرى، بما فيها جدولة المشروع الزمنية، موازنته، الأخطاء، التوثيق وغيرها. (سنقوم من خلال الشرائح التالية بدراسة مكونات وثيقة المتطلبات بالتفصيل) .

المنارة للاستشارات
www.manaraa.com

وثيقة المتطلبات		
التمهيد للمشروع:		
1	هدف المنتج ونطاقه:	
2	سياق العمل:	
3	الأشخاص المعنيون:	
4	أفكار العمل:	
5	نظرة عامة للوثيقة:	
خدمات النظام:		
1	نطاق النظام:	
2	متطلبات النظام:	
3	متطلبات المعطيات:	
قيود النظام:		
1	متطلبات الواجهة:	
2	متطلبات الأداء:	
3	متطلبات الأمن:	
4	متطلبات التشغيل:	
5	متطلبات سياسية وقانونية:	
6	قيود أخرى:	
مواد المشروع:		
1	مواضيع مفتوحة:	
2	الجدول الزمني الأولي:	
3	الموازنة الأولية:	
الملحقات:		
1	معجم المصطلحات:	
2	استمارات ووثائق العمل:	
3	المراجع:	

المنارة للاستشارات
www.manaraa.com

- يتوجه **الجزء التمهيدي** من وثيقة المتطلبات بشكل رئيسي إلى المدراء وصانعي القرار الذين نادراً ما يرغبون بدراسة الوثيقة كلها بالتفصيل

- يتم عادةً توصيف الجزء التمهيدي وشرح هدف المشروع ونطاقه بوضوح في بداية الوثيقة، ومن ثم شرح وتوضيح سياق العمل

- يجب أن تصف وثيقة المتطلبات، في جزئها التمهيدي، حالة عمل للنظام وكيف سيساهم في تحقيق أهداف وغايات عمل المؤسسة

- يتم كذلك في الجزء التمهيدي من وثيقة المتطلبات تعريف جميع الأشخاص ذوي الصلة بالمشروع بحيث ينبغي أن تتم الإشارة إلى الزبون بشكل مباشر، كأن يتم ذكر أسماء الأشخاص المعنيين بشكل مباشر

- من المفيد أن تتم الإشارة في الجزء التمهيدي من وثيقة المتطلبات إلى بعض أفكار الحل منذ المراحل الأولى لدورة حياة التطوير، ويجب الاهتمام هنا بالحلول البرمجية الجاهزة فشراء منتج جاهز يبقى أفضل من تطوير منتج جديد من البداية، وبالتالي يجب أن تعرض الوثيقة قائمة بالحزم والمكونات البرمجية الموجودة التي ستستخدم أو التي سيتم التحري عن مدى صلاحية استخدامها كحلول جاهزة

- نشير أخيراً إلى أنه من الجيد أن تتضمن الفقرة التمهيدية نظرة عامة لبقية محتويات الوثيقة، فقد يجذب هذا الأمر القارئ ويحثه على دراسة أجزاء أخرى من الوثيقة كما يسهل فهم محتوياتها، كما يفضل أن توضح هذه النظرة العامة منهجية التحليل والتصميم التي يتبعها المطورون.

المنارة للاستشارات
www.manaraa.com

وثيقة المتطلبات	
التمهيد للمشروع:	
1 هدف المنتج ونطاقه:	
2 سياق العمل:	
3 الأشخاص المعنيون:	
4 أفكار العمل:	
5 نظرة عامة للوثيقة:	
خدمات النظام:	
1 نطاق النظام:	
2 متطلبات النظام:	
3 متطلبات المعطيات:	
قيود النظام:	
1 متطلبات الواجهة:	
2 متطلبات الأداء:	
3 متطلبات الأمن:	
4 متطلبات التشغيل:	
5 متطلبات سياسية وقانونية:	
6 قيود أخرى:	
مواد المشروع:	
1 مواضيع مفتوحة:	
2 الجدول الزمني الأولي:	
3 الموازنة الأولية:	
الملحقات:	
1 معجم المصطلحات:	
2 استمارات ووثائق العمل:	
3 المراجع:	

المنارة للاستشارات
www.manaraa.com

- يتم تخصيص الجزء الرئيسي والأكبر من وثيقة المتطلبات لتعريف خدمات النظام، وهو عملياً الجزء الوحيد من وثيقة المتطلبات الذي قد يحتوي على نماذج متطلبات العمل

- يمكن نمذجة نطاق النظام برسم مخطط سياق مناسب، كمخطط تدفق معطيات محدد DFD، إذ يجب أن تتضح حدود المشروع المقترح من خلال شرح هذا المخطط، ومن دون تعريف هذه الحدود لن يستطيع المشروع أن يصمد في مواجهة مشكلات اتساع النطاق

- ويمكن نمذجة متطلبات الوظائف بمخطط حالات استخدام العمل (سنقوم بشرح هذه المخططات بالتفصيل في الشرائح التالية)

- بينما تمكن نمذجة متطلبات المعطيات من خلال مخطط صفوف العمل (سنقوم بشرح هذه المخططات بالتفصيل في الشرائح التالية).

المنارة للاستشارات
www.manaraa.com

مكونات وثيقة المتطلبات
قيود النظام

وثيقة المتطلبات		
التمهيد للمشروع:		
1	هدف المنتج ونطاقه:	
2	سياق العمل:	
3	الأشخاص المعنيون:	
4	أفكار العمل:	
5	نظرة عامة للوثيقة:	
خدمات النظام:		
1	نطاق النظام:	
2	متطلبات النظام:	
3	متطلبات المعطيات:	
قيود النظام:		
1	متطلبات الواجهة:	
2	متطلبات الأداء:	
3	متطلبات الأمن:	
4	متطلبات التشغيل:	
5	متطلبات سياسية وقانونية:	
6	قيود أخرى:	
مواد المشروع:		
1	مواضيع مفتوحة:	
2	الجدول الزمني الأولي:	
3	الموازنة الأولية:	
الملحقات:		
1	معجم المصطلحات:	
2	استمارات ووثائق العمل:	
3	المراجع:	

المنارة للاستشارات
www.manaraa.com

تعبر قيود النظام كل الحالات التي تؤدي بطريقة أو بأخرى إلى تقييد عمل النظام ومنعه من القيام بخدماته أو تطويرها، وتحدد قيود النظام وفقاً لما يلي:

- قيود الواجهة
- قيود الأداء
- قيود الأمن
- قيود التشغيل
- قيود سياسية وقانونية

- تعرِّف قيود الواجهة كيف يتخاطب المنتج مع المستخدمين، فنعرف في وثيقة المتطلبات المظهر العام لواجهات الاستخدام البيانية فقط، أما التصميم الأولي لهذه الواجهات فيتم خلال مرحلة توصيف المتطلبات ولاحقاً أثناء تصميم النظام

- تعبر قيود الأداء عن السرعة التي يجب أن ينجز فيها النظام مهامه المختلفة، أو بطريقة أخرى، زمن استجابة النظام. كما يمكن أن تشتمل هذه المتطلبات على قيود أخرى تتعلق مثلاً بوثوقية النظام وجاهزيته للعمل الدائم وغيرها

- تصف قيود الأمن الحقوق والسماحيات الممنوحة للمستخدمين للوصول إلى المعلومات تحت سيطرة النظام، فقد يُعطى المستخدمون إمكانيات مقيدة للوصول إلى المعلومات أو حقوق معينة لتنفيذ عمليات محددة على المعطيات

- تحدد قيود التشغيل البنية العتادية و البرمجية التي سيعمل ضمنها النظام، وقد يكون لهذه المتطلبات أثر مباشر على مواضيع أخرى ذات صلة بالمشروع، كتدريب المستخدمين أوصيانة النظام

- تعبر القيود السياسية والقانونية عن إمكانية أو عدم إمكانية توزيع المنتج لأسباب سياسية أو قانونية وغالباً ما تعتبر معروفة ضمنياً بحيث لا تتم الإشارة إليها صراحةً في الوثيقة.

المنارة للاستشارات
www.manaraa.com

مكونات وثيقة المتطلبات
مواد المشروع

وثيقة المتطلبات		
التمهيد للمشروع:		
1	هدف المنتج ونطاقه:	
2	سياق العمل:	
3	الأشخاص المعنيون:	
4	أفكار العمل:	
5	نظرة عامة للوثيقة:	
خدمات النظام:		
1	نطاق النظام:	
2	متطلبات النظام:	
3	متطلبات المعطيات:	
قيود النظام:		
1	متطلبات الواجهة:	
2	متطلبات الأداء:	
3	متطلبات الأمن:	
4	متطلبات التشغيل:	
5	متطلبات سياسية وقانونية:	
6	قيود أخرى:	
مواد المشروع:		
1	مواضيع مفتوحة:	
2	الجدول الزمني الأولي:	
3	الموازنة الأولية:	
الملحقات:		
1	معجم المصطلحات:	
2	استمارات ووثائق العمل:	
3	المراجع:	

المنارة للاستشارات
www.manaraa.com

- يعبر مقطع "مواضيع مفتوحة" في قسم مواد المشروع من وثيقة المتطلبات عن كل ما يمكن أن يؤثر على نجاح المشروع ولم نتطرق لذكره في موضع آخر من الوثيقة، إذ يمكن أن يتضمن ذلك، النمو المتوقع لأهمية بعض المتطلبات التي تقع حالياً خارج نطاق المشروع، كما يمكن أن يتضمن أي مشكلات محتملة قد تظهر عند توزيع النظام.

- يعبر مقطع " جدول زمني أولي" عن الفترة الزمنية الأولية التقديرية للمشروع، بحيث يشمل هذا تحديداً أولياً للموارد البشرية اللازمة ولغيرها من الموارد.

- يمكن أن تُستخدَم إحدى أدوات إدارة المشاريع لوضع مخططات لخطط قياسية كمخططات PERT على سبيل المثال

- ويعبر مقطع "موازنة أولية" عن الكلفة التقريبية الأولية للمشروع، بحيث يتم استنتاج هذه القيمة اعتماداً على الجدول الزمني الأولي الذي تم تحديده.

المنارة للاستشارات
www.manaraa.com

مكونات وثيقة المتطلبات
الملحقات

وثيقة المتطلبات		
التمهيد للمشروع:		
1	هدف المنتج ونطاقه:	
2	سياق العمل:	
3	الأشخاص المعنيون:	
4	أفكار العمل:	
5	نظرة عامة للوثيقة:	
خدمات النظام:		
1	نطاق النظام:	
2	متطلبات النظام:	
3	متطلبات المعطيات:	
قيود النظام:		
1	متطلبات الواجهة:	
2	متطلبات الأداء:	
3	متطلبات الأمن:	
4	متطلبات التشغيل:	
5	متطلبات سياسية وقانونية:	
6	قيود أخرى:	
مواد المشروع:		
1	مواضيع مفتوحة:	
2	الجدول الزمني الأولي:	
3	الموازنة الأولية:	
الملحقات:		
1	معجم المصطلحات:	
2	استمارات ووثائق العمل:	
3	المراجع:	

المنارة للاستشارات
www.manaraa.com

- يتضمن قسم الملحقات معلومات أخرى يمكن أن تكون مفيدة لفهم المتطلبات التي يتم جمعها وتحديدها، ويمكن أن تشتمل تلك الملحقات على المقاطع التالية:
  - معجم المصطلحات: وفيه يتم تعريف المفردات والاختصارات المستخدمة في وثيقة المتطلبات. ويجب عدم التقليل من أهمية وجود قائمة مصطلحات جيدة إذ قد يؤدي سوء تفسير المفردات إلى أخطاء مكلفة جداً
  - استمارات ووثائق العمل: وفيه يتم جمع وإدراج كافة وثائق واستمارات العمل، مما يحسن بوضوح من إمكانية فهم مجال العمل الخاص بالمشروع، كما يعتبر من الضروري أن يتم إدراج استمارات العمل بعد ملئها بالمعلومات كلما كان ذلك ممكناً، إذ لا تعطي الاستمارات الفارغة الفائدة المرجوة نفسها
  - المراجع: وفيه يتم تحديد الوثائق التي تم استخدامها أو الرجوع إليها عند تحضير وثيقة المتطلبات، وقد تتضمن هذه المراجع كتباً أومصادر أخرى للمعلومات، كمواقع انترنت على سبيل المثال.

توصيف المتطلبات

- استطعنا حتى هذه المرحلة أن نقوم بإلقاء الضوء على مرحلة تحديد المتطلبات من خلال جمع كافة المتطلبات وعلى اختلاف أنواعها ضمن ما يعرف باسم وثيقة المتطلبات
- استخدمنا في مرحلة تحديد المتطلبات عدة أنواع من المخططات التي تساهم بشكل كبير في نمذجة العمليات التي تجري في هذه المرحلة، والتي تصنّف على أنها من أدوات CASE أو ما يعرف باسم الأدوات المساندة في هندسة البرمجيات، كمخططات السياق، مخططات تدفق المعطيات أو مخططات حالات الاستخدام، إلا أنه ومن أجل توصيف المتطلبات فإنه ينبغي استخدام نماذج بيانية ونماذج صورية أخرى بالإضافة إلى هذه النماذج
- توفر لغة النمذجة الموحدة UML لمحلل النظم مجموعة متكاملة من تقنيات النمذجة التي تساعده في أداء مهمته، وتشكل أساساً لنجاح عملية النمذجة والتطوير
- سنقوم من خلال الشرائح التالية بالحديث عن أهم تلك التقنيات المستخدمة في توصيف المعطيات مع العلم أننا سنخصص جلسات للحديث عن كيفية استخدام مخططات لغة النمذجة الموحدة في بناء وتطوير المنتج البرمجي.

نماذج توصيف المتطلبات

- ينتج عن توصيف المتطلبات ثلاث فئات من النماذج: نماذج الحالة ونماذج السلوك ونماذج تغير الحالة
- تتعامل كل فئة من نماذج توصيف المتطلبات مع عدد قليل من تقنيات النمذجة، والتي تقدم لغة النمذجة الموحدة UML دعماً لها
- سنقوم فيما يلي بالحديث عن كل نموذج من هذه النماذج، إلاّ أنه تجدر الإشارة إلى أن تلك النماذج ليست مستقلة عن بعضها البعض، أو بأسلوب آخر، يمكن أن يجري تطوير العديد من النماذج على التوازي بحيث يغذي كل منها النماذج الأخرى وليس من

المنارة للاستشارات
www.manaraa.com

الضروري إنشاء أحدها قبل الآخر

- يصف نموذج توصيف الحالة عالم نظم المعلومات من منظور سكوني للصفوف وصفاتها والعلاقات القائمة فيما بينها، ويمكن التعبير عنه من خلال مخططات الصفوف الذي سنمر على ذكره بالتفصيل في الجلسات القادمة. وهناك العديد من الطرق لاكتشاف الصفوف، بحيث لا توجد قاعدة مطلقة لذلك، إذ يتم تحديد الصفوف اعتماداً على معرفة المحلل وخبرته بشكل رئيسي.

- توصَّف الصفوف في لغة النمذجة الموحدة بواسطة مخططات الصفوف، وتمثل هذه المخططات الصفوف مع ثلاثة أنواع من العلاقات بينها: الاقتران والتجميع والتعميم

- يصف توصيف السلوك عالم نظم المعلومات من منظور عملياتي، وتعتبر حالات الاستخدام القوة المحركة للتوصيف السلوكي وبالتالي لتحليل المتطلبات وتصميم النظام عموماً

- تعطي مخططات حالات الاستخدام رؤية بسيطة للنظام لكن تكمن قوتها الحقيقية في توصيفها السردي. وتشتق من نماذج حالات الاستخدام مخططات سلوكية أخرى مثل مخططات النشاط ومخططات التفاعل وغيرها (سنقوم بشرح مخططات حالات الاستخدام بالتفصيل في الجلسات القادمة)

- يصف مخطط تغير الحالات عالم نظم المعلومات من منظور ديناميكي، إذ تتعرض الأغراض لأحداث قد يؤدي بعضها إلى تغير حالة الغرض، وتجري نمذجة تغيرات الحالة بواسطة مخططات الحالات التي تمثل بياناً من الحالات والانتقالات.

المنارة للاستشارات
www.manaraa.com

القسم السادس والسابع والثامن

بناء مخططات حالات الاستخدام

الكلمات المفتاحية:

حالات الاستخدام، مخطط حالات الاستخدام، فاعل، حاويات، رزم، حدود النظام، علاقة تجميعية، علاقات التبعية، علاقات التعميم، العامل البشري، الفاعل الأولي، الفاعل الثانوي، المُنشِئ، المُخدِّم الخارجي، المُتلقِّي، الوسيط، الفاعل المجرَّد، تعميم، الشروط المسبقة، الشروط اللاحقة، علاقة الاحتواء، علاقة التمديد، حالة استخدام بسيطة، حالة استخدام متقدمة.

ملخص:

سنناقش خلال هذه الشرائح مخططات حالات الاستخدام، بحيث سنقوم بإجراء دراسة تفصيلية عن هذه المخططات وعن الدور الهام الذي تلعبه في مرحلة تحليل نظم المعلومات.

أهداف تعليمية:

سيتعرف الطالب في هذا الفصل على المهارات التالية:

- خطوات العملية التقليدية والموسّعة المُستخدمة في بناء حالات الاستخدام
- تصميم مخططات حالات الاستخدام
- ما هي حالة الاستخدام؟
- من هو الفاعل؟
- ما هي الحاويات والرزم؟
- العلاقات
  - العلاقات التجميعية
  - علاقات التبعية
  - العلاقة <<include>>
  - العلاقة <<extend>>
  - علاقات التعميم.
- الفائدة من تحديد الفاعلين في النظام
- كيف يتم تحديد الفاعلين
- أنواع الفاعلين:
  - الفاعل الأولي
  - الفاعل الثانوي.
- التصنيف الوظيفي للفاعلين
- الفاعل المجرد
- تمثيل الفاعل باستخدام لغة النمذجة الموحدة
- حالات الاستخدام
- وصف حالات الاستخدام وتدوين حالات الاستخدام البسيطة والمتقدمة

المنارة للاستشارات
www.manaraa.com

- علاقات التمديد والاحتواء والتعميم
- حالة الاستخدام المجردة
- أمثلة تطبيقية.

المنارة للاستشارات
www.manaraa.com

مقدمة

- سنقوم في هذه الجلسة بدراسة مخططات حالات الاستخدام ابتداءً من التعريفات الأساسية، مروراً ببناء حالات الاستخدام البسيطة، وصولاً إلى توصيف حالات الاستخدام المتقدمة

- سنناقش من خلال هذه الجلسة كافة حيثيّات حالات الاستخدام وخصائصها ومكوّناتها، بالإضافة إلى الهدف من استخدامها والدور الهام الذي تلعبه في مرحلة تحليل نظم المعطيات

- يمكن أن يَعتبر البعض أن استنتاج حالات الاستخدام واستخراجها بالأمر البسيط، كما يمكن أن يعتبرها البعض الآخر بالعملية الصعبة أو المعقّدة نسبياً، إلاّ أننا سنقوم من خلال دراستنا هذه بتبسيط هذا المفهوم وشرح أهميّته وضرورة استخدامه.

تمهيد

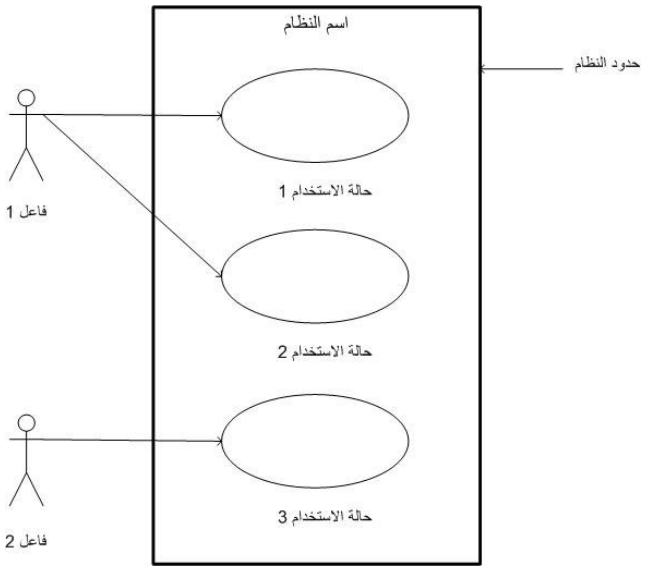
- يمكن التعبير عن العملية التقليدية المُستخدمة في بناء حالات الاستخدام من خلال ثلاث خطوات رئيسية متتالية، وهي:
  1. تحديد الفاعلين
  2. تحديد حالات الاستخدام
  3. وصف كل حالة استخدام على حده.

- يجري عادةً "إهمال" أو "تضمين" عدّة خطوات أخرى في الخطوات الثلاث السابقة –وخاصةً عندما يتعلق الأمر بالمستخدمين الجدد على أسلوب النمذجة باستخدام حالات الاستخدام– بحيث يمكننا التعبير عن خطوات بناء نموذج حالات الاستخدام من خلال تسلسل النقاط التالية:
  1. تعريف حدود النظام
  2. تحديد الفاعلين
  3. تحديد حالات الاستخدام
  4. وصف كل حالة استخدام على حده.
  5. إعادة تحليل نموذج حالات الاستخدام
  6. تحديد الأولويات فيما بين حالات الاستخدام المختلفة، أو بطريقة أخرى، تثقيل حالات الاستخدام
  7. إضافة المتطلبات المستقبلية
  8. إنشاء نموذج حالة الاستخدام.

تشكل مراحل العملية الموسّعة في بناء حالات الاستخدام تلك، الأساس الذي يتم الاعتماد عليه في بناء حالات الاستخدام المتقدّمة، والتي سنتحدث عنها بالتفصيل في الجلسات القادمة، في حين يعتبر اختصار مراحل العملية الموسّعة تلك، كافياً من أجل بناء العديد من الأنظمة البسيطة.

المنارة للاستشارات
www.manaraa.com

- يُعتبر بناء مخطط حالات الاستخدام مرحلة هامة وأساسية في عملية نمذجة حالات الاستخدام، بحيث يُستخدم للتعبير عن حالات الاستخدام المُضمّنة في النظام الذي نقوم بتحليله، وذلك بشكل مبسّط وباستخدام أشكال تعبيرية للدلالة على حالات الاستخدام دون التطرّق إلى تفاصيلها، كما يوضّح الشكل التالي:



- يمكن تعريف مخططات حالات الاستخدام من خلال لغة النمذجة الموحدة UML على أنها عبارة عن بيان، تعبّر عقدُه عن حالات الاستخدام والفاعلين، في حين تعبّر الوصلات فيه عن العلاقات التي تربط بين حالات الاستخدام، وعن العلاقات التي تربط بين الفاعلين، وعن العلاقات التي تربط بين كل من حالات الاستخدام والفاعلين

- تعتبر الخطوة الأولى من العمليّة الموسّعة في بناء حالات الاستخدام –كما مرّ معنا– هي تعريف حدود النظام نقوم عادةً بتعريف حدود النظام مع العلم أنه ليس من الضروري تحديد ذلك، إذ غالباً ما تتم المباشرة بنمذجة حالات الاستخدام قبل الإجماع على ما ينبغي بناؤه
- ويجري عادةً وصف حدود النظام في مخططات حالات الاستخدام من خلال مستطيل يحتوي على اسم النظام في الزاوية العلوية منه، بحيث تتوضع العناصر التي تكوِّن النظام بحد ذاته في داخل ذلك المستطيل، في حين تتوضع العناصر الأُخرى –والتي لا تعتبر من النظام– خارج ذلك المستطيل

- عادةً ما تُختتم الخطوة الأولى من عملية بناء حالات الاستخدام من خلال إطلاق اسم مناسب على المخطط الذي نقوم ببنائه، بحيث ينبغي على ذلك الاسم أن يعبر عن النظام، خاصةً بوجود الفاعلين الذين يعبّرون بشكل مناسب عمّا يريدونه من النظام وعن كيفية بداية ونهاية النظام

المنارة للاستشارات
www.manaraa.com

تعاريف

- تُعبِّر حالة الاستخدام عن تسلسل لأفعال محددة في النظام، بحيث تعرّف مخططاً وظيفياً يتم تنفيذه في ذلك النظام. ويتم التعبير عنها في مخطط حالات الاستخدام على شكل قطع ناقص بحيث يظهر ضمنه أو تحته اسم حالة الاستخدام تلك

- تقوم حالات الاستخدام بتأمين قيمة قابلة للقياس أو للمراقبة لفاعل أو أكثر، بحيث يمكن أن يمثل الفاعل أي شيء خارج النظام يقوم بتبادل المعلومات معه، ويتضمن ذلك مستخدمي النظم الأخرى. يتم تمثيل الفاعلين في مخططات حالات الاستخدام من خلال رمز "شخص" وتحته اسم الدور الذي يلعبه ذلك الفاعل

- يمكن أن تحتوي الحاويات على حالات استخدام، بحيث تمثل تلك الحاويات حدود النظام. يمكن للرزم –والتي تعتبر أحد أنواع الحاويات– أن تحتوي على حالات استخدام أو فاعلين أو حالات استخدام وفاعلين، مع العلم أنه يمكن للرزم أيضاً أن تحتوي على حاويات أخرى

- تربط العلاقات عنصرين من مخطط حالات الاستخدام بعضهما ببعض، سواء كانت تلك العناصر عبارة عن حالات استخدام أو فاعلين أو رزم، وتختلف أنواع العلاقات المُستخدمة بحسب الغرض من استخدامها، إذ تدل العلاقات التجميعية عن كيفية التفاعل ما بين حالات الاستخدام والفاعلين، كما يمكن أن تدل علاقات التبعية القائمة بين رزمتين على وجود عنصر من الرزمة الأولى يرتبط بعنصر آخر من الرزمة الثانية، بحيث يمكن لذلك العنصر أن يكون عبارة عن حالة استخدام أو فاعل ما، مع العلم أنه يوجد أنواع أخرى من العلاقات، كالعلاقة <<include>> والعلاقة <<extend>> التي تربط حالات الاستخدام، والتي سنتحدث عنها بالتفصيل في الشرائح القادمة، بالإضافة إلى علاقات التعميم التي تعبر عن وراثة السلوك بين حالات الاستخدام أو الفاعلين.

سلوك النظام

- يُقصد بسلوك النظام كيفية استجابته وتفاعله مع الأحداث الخارجية، ويعبر عن سلوك النظام المرئي من الخارج والقابل للاختبار والتطبيق من خلال حالات الاستخدام، بحيث تنجز كل حالة استخدام وظيفة محددة ومستقلة يمكن أن يراها الفاعل الخارجي وأن يختبرها باستقلالية

- يمثل الفاعل أي كائن (شخص، آلة...) يمكن أن يتفاعل مع حالة الاستخدام، ويتوقع الفاعل أن يحصل من حالة الاستخدام على نتيجة مفيدة قابلة للقياس

- يُعرف مخطط حالات الاستخدام بأنه التمثيل المرئي للفاعلين وحالات الاستخدام بالإضافة إلى أية تعاريف أو توصيفات إضافية، وهو ليس مجرد مخطط بل هو أيضاً نموذج موثق بالكامل لسلوك النظام المطبَّق.

الفاعلون

- يسعى التوجّه الحديث في مجال هندسة البرمجيات إلى توصيف ما يجري حولنا في العالم الحقيقي وبناء النماذج المناسبة التي تعبّر عنه، خاصةً وأنه من المعروف أن قيمة النظم البرمجية تعتمد وبشكل رئيسي على سلوك تلك النظم وأدائها، والذي يرتبط بدوره

المنارة للاستشارات
www.manaraa.com

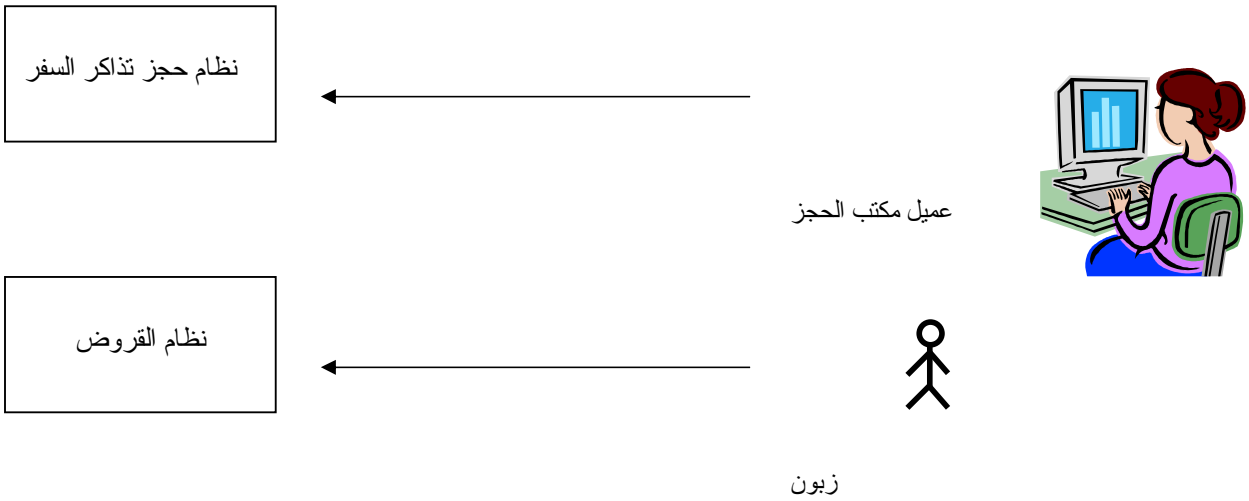
بعلاقات مختلفة مع كيانات خارج النظام، كالمستخدمين أو البرمجيات الأخرى أو غيرها

- يُطلق عادةً على العمليات والتفاعلات التي تنشأ بين النظام والكيانات المرتبطة به، اسم الأحداث، بحيث تعمل بعض تلك الكيانات على إنشاء الأحداث وإطلاقها في حين تتفاعل كيانات أخرى مع النظام كنتيجة لحدث معين

- يُطلق على تلك الكيانات التي نتحدث عنها اسم: "الفاعلون".

- يساهم الفاعلون في استنتاج حالات الاستخدام وذلك من خلال تأمين صورة واضحة للتفاعلات التي تجري في النظام

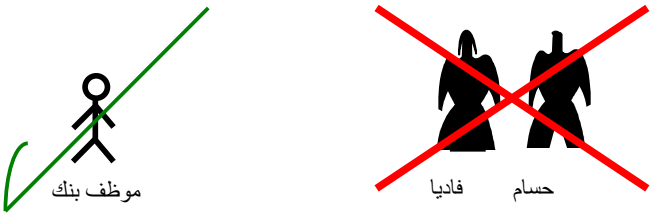
- يمكننا أن نعرّف الفاعل على أنه ذلك الكيان الذي يتفاعل مع النظام بغرض إتمام حدث معيّن، كالزبون الذي يطلب قرض ما من بنك محدد، أو عميل حجز بطاقات السفر من خلال شبكة الوِب



- ليس من الضروري أن يكون الفاعلون في نظام ما أشخاصاً حقيقيين، إذ يمكن أن يكونوا عبارة عن نظم أُخرى أو مؤسسات خارجية أو تجهيزات خارجية أو أية كيانات خارجية أخرى تتفاعل مع النظام، فقد يكون الوقت مثلاً أحد الفاعلين في النظام

- عندما يكون الفاعلون عبارة عن أشخاص حقيقيين، يمكن تشبيه ذلك بالدور الذي يلعبه مستخدم ما يتفاعل مع النظام، مع العلم أن الفاعل لا يمثل شخص أو فرد أو كيان محدد، ففي المثال السابق، قمنا باعتبار عميل حجز بطاقات السفر أحد فاعلي النظام، ولم نعتبر "باسم سعيد" موظف مكتب الحجز بأنه فاعل في النظام، أي بطريقة أُخرى، يعبر الفاعل عن نموذج لمفهوم موظف الحجز

المنارة للاستشارات
www.manaraa.com



- الفاعل بالتعريف هو الدور وليس الشخص، وبالتالي يمكن أن يقوم الفاعل بنمذجة العديد من الأشخاص أو الأمثَال، مع العلم أنه يمكن كذلك لفرد محدد أن يلعب أكثر من دور، أي أن يمثِّل أكثر من فاعل، فمدير البنك مثلاً يمكن أن يكون زبون البنك في الوقت نفسه

- ينبغي أن نفرّق ما بين الفاعلين الذين يمثلون دوراً محدداً يمكن استنتاج عدّة أمثال منه، وبين الفاعلين الذين يمثلون كيانات فيزيائية، كنظام معرّف مسبقاً أو ميزان حرارة يقوم بتبادل المعلومات مع النظام، ويمكن اعتبار هذا النوع من الكيانات فاعلين دون أدنى شك.

ما هي الفائدة من تحديد الفاعلين في النظام؟

- يزودنا الدور الذي يلعبه الفاعلون بمنظور عام حول أهمية حالات الاستخدام والفائدة المرجوّة منها

- تؤمن كل من الأدوار التي يلعبها الفاعلون في البيئة المحيطة، والسلوك الذي يتبعونه، ومسؤولياتهم تجاه النظام، عاملاً أساسياً ومؤثراً على حالات الاستخدام، فبالتركيز على الفاعلين، يمكننا أن نركز اهتمامنا على الكيفية التي سيتم من خلالها استخدام النظام بدلاً من الاهتمام بكيفية بناء النظام وتطبيقه في هذه المرحلة المبكرة

- يساعدنا التركيز على الفاعلين في النظام على مراجعة وتأكيد حدود النظام وتعريفها بالشكل الأنسب، هذا بالإضافة إلى الدور الذي يلعبه الفاعلون في تحديد كافة متطلبات النظام

كيف يتم تحديد الفاعلين في النظام؟

- ينبغي أثناء نمذجة حالات الاستخدام أن ننتبه لمجموعة من النقاط الهامة التي تساهم في تحديد فاعلي النظام المُحتملين واستنتاجهم، منها:
  - ينبغي البحث عن كافة الكيانات الخارجية التي تتفاعل مع النظام، ويمكننا لتحقيق ذلك أن نقوم بالاعتماد على مخططات مختلفة تصف سياق الأحداث التي تجري على النظام، أو على النماذج المختلفة التي تصف حدود النظام مع البيئة المحيطة
  - تحليل العامل البشري: يُقصد بالعامل البشري كل الأشخاص المعنيين بالمشروع البرمجي بطريقة أو بأخرى، سواء كانوا سيتأثرون بالنظام أو سيؤثرون على تطويره، ويُقصد بالأشخاص المعنيين بالمشروع، كل من الزبائن (أي المستخدمين ومالكي النظام) والمطورين (من محللين ومصممين ومبرمجين...).

المنارة للاستشارات
www.manaraa.com

- تُساهم التوصيفات المكتوبة وكافة الوثائق المرتبطة بالمشروع، كالملاحظات المُسجّلة، واللقاءات المسجلة مع المستخدمين، في تحديد وتعريف الفاعلين المحتملين في النظام
- قد يساعد دليل المستخدم أو الكتيّبات المخصصة للنظام الحالي أو للعملية قيد الاستخدام، على استنتاج فاعلي النظام الجديد المُحتملين.

- عادةً ما يُنصح بطرح الأسئلة التالية عند البحث عن فاعلي النظام أو استنتاجهم:
  - مَن أو ما الذي يقوم بإطلاق الأحداث في النظام؟
  - مَن أو ما الذي يتفاعل مع النظام من أجل الاستجابة لحدث معين؟
  - هل توجد أية تقارير خرج؟
  - هل توجد أية واجهات مخصصة لإدارة النظام؟
  - هل يحتاج النظام حالياً أو مستقبلياً للتعاطي أو تبادل المعلومات مع نظام آخر موجود مسبقاً؟
  - هل يوجد فاعلون مُعرّفون مسبقاً في النظام؟
  - هل توجد أية تجهيزات عتادية أو برمجية تقوم بالتفاعل مع النظام الذي ينبغي أن تتم نمذجته أثناء مرحلة التحليل؟
  - إذا حصل حدث معين في النظام، هل من الضروري إبلاغ أي كيان خارجي بهذا الحدث؟
  - هل يحتاج النظام إلى أن يسأل كيان خارجي معين أي سؤال بهدف إتمام مهمة معينة؟

أنواع الفاعلين

- لا تتحدد حالة الاستخدام بفاعل وحيد فقط، إذ من الممكن أن يرتبط بحالة الاستخدام عدد غير محدد من الفاعلين، فحالة الاستخدام موجودة لإعادة قيمة معينة لفاعل واحد على الأقل

- يمتلك الفاعلون، المرتبطون بحالة استخدام معينة، بحدّ ذاتهم، أدواراً مختلفة ومسؤوليات متعددة، فبعضهم يستقبل قيمة محددة، وبعضهم الآخر يُؤدي خدمات معينة، في حين يساهم غيرهم بتنفيذ أو إطلاق الأحداث أو تشغيل حالات الاستخدام

- يُقسم عادةً الفاعلون إلى نوعين أساسيين، هما:
  - الفاعل الأولي
  - الفاعل الثانوي.

سنستعرض فيما يلي من شرائح خصائص وصفات كل من هذين النوعين.

المنارة للاستشارات
www.manaraa.com

أنواع الفاعلين

الفاعل الأولي

- يُطلق اسم الفاعل الأولي على المستخدم الذي يقوم باستخراج قيمة محددة من النظام، بحيث تؤثر حاجات الفاعل الأساسي على سلوك وأداء حالة الاستخدام بحد ذاتها، فلو تغيرت تلك الاحتياجات، كان لابد من إجراء تعديلات هامة وأساسية على النظام

مثال:

لنفترض وجود نظام معالجة طلبات قروض لمصرف محدد، بحيث يتم إجراء تلك الطلبات وتحريرها من قبل موظفي المصرف

ولنفترض أنه تم تطوير عمل المصرف من خلال تبني أسلوب جديد يسمح بإجراء طلبات القروض من خلال نظام خاص يعمل على شبكة الوِب

على الرغم من أن الإضافة الجديدة على النظام تبدو للوهلة الأولى بأنها عبارة عن نفس الخدمة الأولى المتاحة، إلاّ أننا لا نستطيع ببساطة اعتبار الأمر على أنه مجرد إضافة واجهة وِب جديدة إلى النظام لتقوم بإنشاء طلبات القروض، فموظفو المصرف يمتلكون سماحيات ولوج في النظام الذي يعملون عليه تختلف بالتأكيد عن السماحيات التي يمكن إعطاؤها لمستخدمي واجهة الوِب الذين يتقدمون بطلب القرض، بحيث يمكن لموظفي المصرف أن يقوموا بالولوج إلى معلومات مختلفة حول القرض الذي يقومون بتحريره، في حين لا يمكن اعتبار هذه العملية متاحة للمتقدمين من خلال الوِب

على الرغم مما سبق، فإنه ينبغي الانتباه إلى أن متطلبات الاستخدام تختلف أيضاً في ظل هذه التغيرات الجديدة، إذ يستطيع موظفو المصرف أن يتحكموا باستخدام النظام وفق سماحياتهم المتاحة وذلك بعد تدريبهم بالشكل المناسب، في حين تنحصر سماحيات مستخدمي الوِب بتصفح الصفحات المتاحة فقط مع الأخذ بعين الاعتبار عدم إمكانية تدريب هذا النوع من المستخدمين على استخدام الواجهات اللازمة.

- عادةً ما يُنصح بطرح الأسئلة التالية عند البحث عن الفاعل الأولي في النظام:
  - ما هو الإجراء الرئيسي الذي يقوم به الفاعل في المؤسسة بشكل عام؟
  - ما هي القيمة القابلة للقياس من بين الأدوار والمسؤوليات المنجزة من قبل الفاعل؟ (ستعبر الإجابة عن هذا السؤال – حتماً– عن السلوك والمتطلبات المحددة في حالة الاستخدام بحيث يؤثر تغيير القيمة الناتجة بالضرورة على ذلك السلوك وتلك المتطلبات)
  - ما هو السلوك الذي ينبغي على النظام أن يُؤمنه للوصول إلى تلك القيمة؟
  - ما هي المتطلبات الوظيفية وقواعد العمل المرتبطة بهذه القيمة؟
  - هل ترتبط بهذه القيمة أيٌّ من متطلبات الأداء أو الزمن أو متطلبات واجهات العرض أو غيرها؟

المنارة للاستشارات
www.manaraa.com

أنواع الفاعلين

الفاعل الثانوي

- الفاعل الثانوي هو الذي يقوم بتنفيذ مهمة ما في حالة الاستخدام بحيث لا يمكن أن يوجد من دون وُجود فاعل أولي يرتبط به
- عادةً ما يُشارك الفاعل الثانوي في حالة الاستخدام بغرض تأمين الدعم اللازم لإنشاء قيم محددة لفاعل أوّلي أو أكثر

مثال:

يمكن اعتبار الفاعل المسؤول عن تنصيب التطبيقات أو إجراء عمليات الخزن الاحتياطي في نظام معالجة طلبات القروض في مصرف ما، أنَّه يمثِّل فاعلاً ثانوياً في ذلك النظام

- عادةً ما يُنصح بطرح الأسئلة التالية عند البحث عن الفاعل الثانوي في النظام:
  - ما هي القيمة التي يدعم هذا الفاعل عملية حسابها في حالة الاستخدام؟ هل ستؤثر التغيرات التي تطرأ على تلك القيمة أثناء تطوير النظام، على ذلك الفاعل؟
  - هل هناك أيٌّ من متطلبات الأداء أو الزمن أو متطلبات واجهات العرض أو غيرها ترتبط بهذه الخدمة؟

التصنيف الوظيفي للفاعلين

- يمكن تصنيف أنواع الفاعلين وظيفياً ضمن أربعة أنواع رئيسية، بحيث يمكننا الاعتماد على هذا التصنيف من أجل اكتشاف وتعريف الفاعلين المشاركين في حالة الاستخدام

- ينبغي قبل أن نتحدث عن التصنيف الوظيفي للفاعلين أن نُشير إلى إمكانية أن يساهم الفاعل في أكثر من وظيفة محددة، كما سيمر معنا من خلال شرح هذه الخصائص الوظيفية:

  - **المُنشِئ:** يُطلق على فاعل ما صفة المُنشئ عندما يكون له دور ما في تأسيس سلوك في النظام بغرض إتمام حدث معين. يمكن أن يقوم المُنشِئ بطلب خدمة معينة، أو أن يقوم بإطلاق حدث ما.

مثال: يُصنف الزبون الذي يتقدم بطلب قرض من المصرف بأنه مُنشئ الحدث "طلب قرض".

ليس من الضروري أن يكون المُنشئ فاعلاً أولياً، وعادةً ما يُنصح بطرح الأسئلة التالية عند البحث عن الفاعل المُنشئ لحدث ما في النظام:

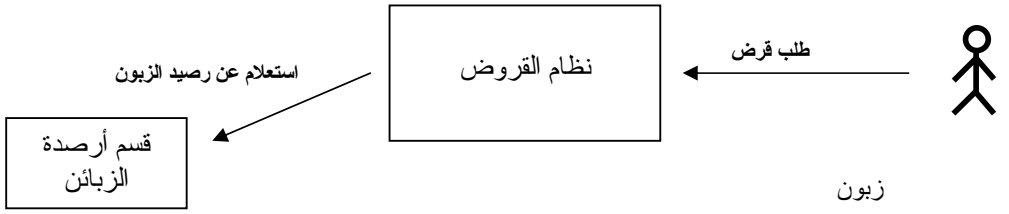
- ما هو الحدث الذي يقوم هذا الفاعل بإطلاقه؟
- هل يقوم هذا الحدث بإنشاء حالة الاستخدام؟
- هل يعتمد هذا الحدث على الزمن ليتم إطلاقَه؟
- ما هي المتطلبات التي ترتبط بأسلوب تعاطي هذا الفاعل مع النظام؟

  - **المُخدِّم الخارجي:** يُطلق عادةً على الكيان الخارجي –(الذي يمكن أن يكون شخص أو مؤسسة أو نظام خارجي)– والذي يستجيب لطلب ما من طلبات النظام، اسم المُخدِّم الخارجي، ويساهم هذا النوع من الفاعلين بتأمين خدمات مناسبة لطلبات

المنارة للاستشارات
www.manaraa.com

النظام.

مثال: يقوم نظام معالجة القروض، في حالة الاستخدام "طلب قرض" السابقة، بالاستعلام عن الرصيد المتاح لعملية الإقراض من خلال طلب تلك المعلومات من القسم المناسب، يُطلق على ذلك القسم اسم المُخدِّم الخارجي، خاصةً وأنه يستجيب لطلب يقوم النظام بإجرائه من أجل تنفيذ خدمه محددة.



عادةً ما يكون المُخدِّم الخارجي فاعلاً ثانوياً، ويُنصح بطرح الأسئلة التالية عند البحث عن المُخدِّم الخارجي لحدث ما في النظام:

- ما هي الخدمة التي يؤمنها هذا الفاعل؟ وكيف ترتبط بالقيمة التي تقوم حالة الاستخدام بتأمينها لفاعل آخر؟
- ما هي المتطلبات التي ترتبط بهذا الفاعل من معلومات أو توقيت أو غير ذلك؟
- إذا كان المُخدِّم الخارجي عبارة عن نظام آخر، فهل ينبغي تعديل أو تطوير سلوك ذلك النظام ليتوافق مع الدور الذي سيلعبه في حالة الاستخدام؟

- **المُتلقِّي الخارجي:** يُطلق على فاعل ما اسم المُتلقّي عندما يقوم باستلام معلومات من النظام

يمكن اعتبار مخزن المعطيات متلقِّي خارجي، خاصةً وأنه يقع خارج حدود النظام ويستقبل معلومات منه

يُنصح بطرح الأسئلة التالية عند البحث عن المُتلقِّي الخارجي لحدث ما في النظام:

ما هي المعلومات التي يستقبلها هذا الفاعل؟ ولماذا يحتاج لهذه المعلومات؟ (يحتاج مخزن المعطيات على سبيل المثال لهذه المعلومات من أجل إجراء عمليات تحليل مختلفة)

ما هو نوع المعطيات التي يحتاجها هذا الفاعل؟

ما هي المتطلبات التي ترتبط بهذا الفاعل من معلومات أو توقيت أو غير ذلك؟

- **الوسيط:** يُطلق على فاعل ما اسم الوسيط عندما يلعب دور الوساطة لتأمين خدمة محددة يقدمها النظام لفاعل أولي لا يملك سماحيات الولوج لتلك الخدمة

يمثل موظف الإدخال في حالة الاستخدام المسؤولة عن طلب قرض من المصرف، مثالاً على الفاعل الوسيط الذي يقوم بإدخال طلب الزبون إلى النظام

يُنصح بطرح الأسئلة التالية عند البحث عن الفاعل الوسيط لحدث ما في النظام:

- ما هي الخدمات التي يؤديها الوسيط؟
- ما هي القيود التي يفرضها الوسيط على طريقة تعاطي فاعل أولي مع النظام؟
- هل ينبغي وجود واجهات تخاطبية مخصصة لخدمة الفاعل الوسيط؟

المنارة للاستشارات
www.manaraa.com

- هل هناك احتمالية لاستبدال الفاعل الوسيط بواجهات تخاطبية مؤتمتة تقوم بتأدية مهماته مستقبلياً؟

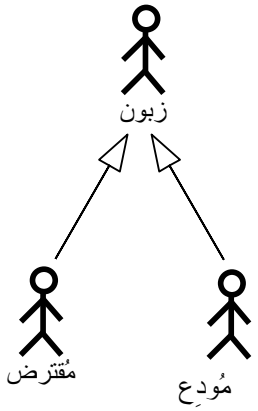
الفاعل المجرّد

- تلعب بعض أنواع الفاعلين أدواراً واقعية محددة، في حين تلعب أنواعٌ أُخرى أدواراً مفاهيمية مجرَّدة، فعلى سبيل المثال يمكن أن يحتوي نظام "المصرف" دوراً مجرّداً يُطلق عليه اسم "الزبون" الذي يستفيد من الخدمات المختلفة التي يقدّمها المصرف، إلا أنه ومن جهة أُخرى، يمكن أن يحتوي على أدوار أكثر تحديداً كالمُودِع أو المُقترِض، اللّذين يمثلان بالضرورة أنواعاً من زبائن النظام، إذ لا يُعتبر دور الزبون كافياً لتحديد كافة العناصر الضرورية لوصف مُستخدِم نظام الإقراض

- يعتبر دور "الزبون" في المثال السابق دوراً رئيسياً في حين يعتبر دور المُودِع والمُقترِض دوراً فرعياً، وتربط بينهما علاقة تعميم

- يمثِّل الفاعل المجرَّد السلوك المشترك ما بين فاعلين أو أكثر، وهذا المفهوم شبيه بمفهوم الصف المجرَّد في لغات البرمجة غرضية التوجّه

- ليس من السهل اكتشاف الفاعلين المجرَّدين مبكراً، إذ يتم ذلك بعد التقدّم في عملية تحليل النظام ودراسة تفاعله مع الفاعلين الذين يؤثرون ويتأثرون به.



تمثيل الفاعل باستخدام لغة النمذجة الموحدة

- يُفضل عادةً استخدام التدوين القياسي المحدد من خلال لغة النمذجة الموحدة UML للتعبير عن فاعلي النظام ولتوثيق المعلومات عنهم

- تستخدم لغة UML رمز الشخص للتعبير عن فاعل ما في النظام وتحته اسم الدور الذي يلعبه:

المنارة للاستشارات
www.manaraa.com

- يمثل الجدول التالي الجدول المستخدم لتدوين معلومات الفاعل، وفيه نجد حقلاً يعبر عن اسم الفاعل، وحقلاً ثانياً لإضافة وصف حول وظيفة أو دور هذا الفاعل، وحقلاً ثالثاً للدلالة عما إذا كان فاعلاً مجرّداً أم لا:

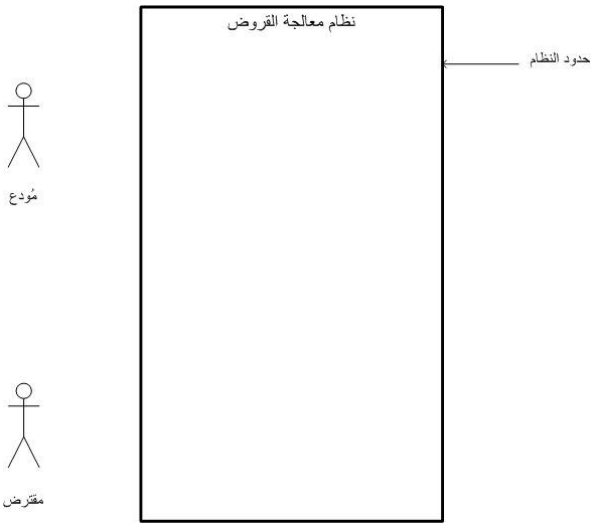
اسم الفاعل	<الاسم>
مجرّد	<نعم/لا>
الوصف	<وصف الدور الذي يلعبه الفاعل>

- سنقوم في الشرائح التالية بدراسة حالات الاستخدام وكيفية البحث عنها واستنتاجها، بالإضافة إلى مناقشة كيفية استخدام لغة النمذجة الموحدة للتعبير عن حالات استخدام النظام وتوثيقها وتدوينها

حالات الاستخدام

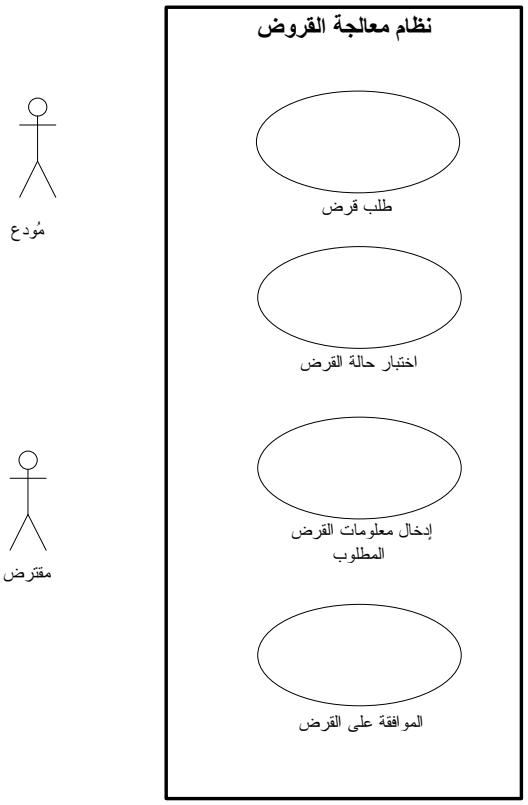
- تصف حالات الاستخدام كيفية توظيف النظام من قبل فاعليه من أجل تحقيق أهدافهم
- وتعبّر حالة الاستخدام عن تتالي أحداث يقوم النظام بتنفيذها، والتي تقوم بدورها بإعادة نتيجة ذات قيمة محددة لفاعل ما
- لكي نفهم كيفية البحث عن حالات الاستخدام في نظام ما سنقوم بدراسة المثال الذي كنّا قد تحدثنا عنه في شرائح سابقة، والذي كان يعالج مسألة طلب قرض من مصرف. لقد بدأنا عملية التحليل بتعريف حدود النظام ثم بتحديد الفاعلين ورسمهم خارج حدود النظام، لنبدأ بعد ذلك بالبحث عن حالات الاستخدام وتضمينها في المخطط:

المنارة للاستشارات
www.manaraa.com



- لكي نبحث عن حالات الاستخدام في نظامٍ ما، لابد لنا أولاً من أن نحدد أهداف ذلك النظام. يمكن أن تتضمن أهداف نظام معالجة القروض مثلاً ما يلي:
  - "طلب قرض": يتقدم المُقترض بطلب يحتوي على المعلومات الضرورية للحصول على القرض، ويتم بعد ذلك اختبار تلك المعلومات من أجل ضمان توافقها مع متطلبات النظام
  - "اختبار حالة القرض": يقوم المُقترض بالاستعلام عن أية تبدلات في حالة القرض الذي يطلبه قبل المتابعة في إجراءات عملية الاقتراض
  - "إدخال معلومات إضافية عن القرض المطلوب": يتطلب الحصول على قرض من النظام إدخال معلومات إضافية من أجل إتمام الإجراء، كاستفسارات محددة عن بعض المشاكل التي يمكن أن تواجه المُقترض من حيث التأمينات المتوفرة أو معلومات حول سجلّ الزبون...الخ
  - "الموافقة على القرض": ينبغي على المُقترض –عند المُصادقة على طلب القرض– أن يوافق على كافة الشروط والإجراءات المتعلقة بهذه العملية.
- تساعدنا النقاط السابقة على استنتاج حالات الاستخدام، فكل هدف يعيد قيمةً ما كنتيجة قابلة للقياس لفاعلٍ محدد، يمكن اعتباره حالة استخدام مستقلّة في النظام

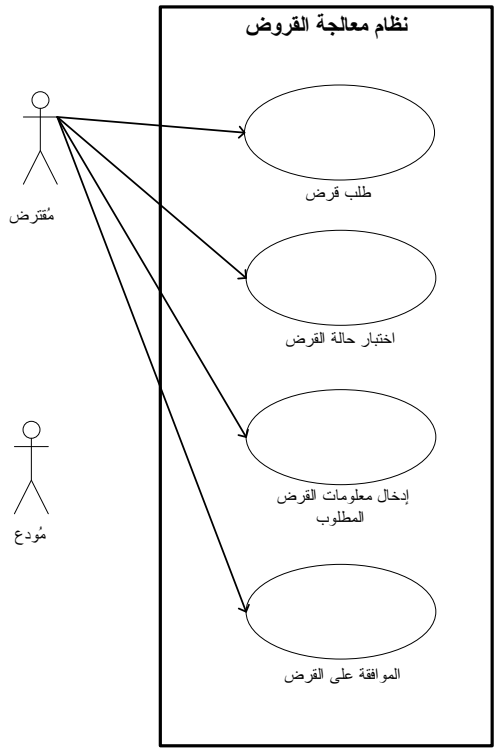
المنارة للاستشارات
www.manaraa.com



العلاقات التجميعية

- تصف حالات الاستخدام العلاقات الناشئة ما بين النظام والفاعلين فيه، ويُطلق على العلاقات التي تربط الفاعلين بحالات الاستخدام اسم العلاقات التجميعية

المنارة للاستشارات
www.manaraa.com



- يمكن أن تكون العلاقات التجميعية موجّهة أو ثنائية الاتّجاه

- يتم تمثيل العلاقة التجميعية الموجَّهة من خلال سهم موجّه من المُنشئ إلى حالة الاستخدام عندما يلعب الفاعل دور المُنشئ، في حين تكون عبارة عن سهم موجَّه من حالة الاستخدام إلى الفاعل عندما يلعب الفاعل دور المتلقِّي

وصف حالات الاستخدام وتدوين حالات الاستخدام البسيطة

- بعد أن نقوم بتحديد أهداف النظام ورسمها ضمن مخطط حالات الاستخدام داخل المستطيل المعبِّر عن حدود النظام، يصبح من الضروري وصف كيفية تحقيق الأهداف تلك، إذ ينبغي عدم الاكتفاء برسم مخطط يدل على حالات الاستخدام وكيفية ارتباطها مع فاعلي النظام، إنما ينبغي توصيف تسلسل الأحداث التي تم التعبير عنها في المخطط من خلال أسلوب تدوين خاص يطلق عليه أحياناً اسم هيكل أو جسم حالة الاستخدام

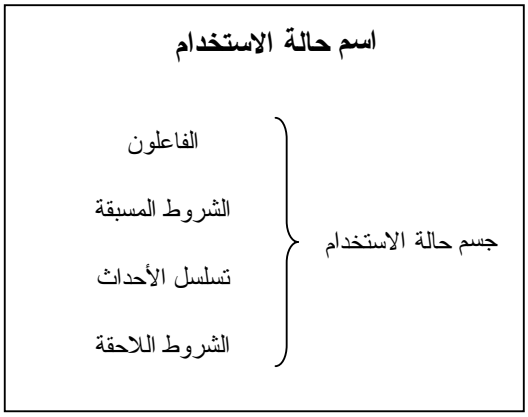
- رأينا سابقاً أن لكل حالة استخدام اسم دلالي يعبِّر عن الهدف الذي بُنيت من أجله، فما أن نقوم بتسمية حالة الاستخدام وتحديد هدفها، يصبح ضرورياً وصف ذلك الهدف من خلال نص مقتضب مكتوب بأسلوب سردي بسيط، ومضمَّن في جسم حالة الاستخدام

- يُقسم جسم حالة الاستخدام إلى عدّة بُنى منطقية تُساهم في جعل حالة الاستخدام متّسقة في كافة أجزاء النظام، ويساعد هذا التدوين على تسهيل أسلوب قراءة وفهم الغرض من حالة الاستخدام والهدف الذي تسعى لتحقيقه

- لا يوجد فعليّاً طريقة قياسية للتعبير عن البنى المنطقية المكوِّنة لجسم حالة الاستخدام، إلا أنه يمكننا أن نحدد مجموعة بنود معلومات ينبغي أن تتوافر بشكل دائم ضمن هذا التدوين، وهي:

المنارة للاستشارات
www.manaraa.com

- الفاعلين: حيّز خاص لوصف الفاعل أو الفاعلين الذين يؤثرون ويتأثرون بحالة الاستخدام
- الشروط المسبقة: حيّز خاص للتعبير عن الشروط التي ينبغي تحقيقها قبل إطلاق حالة الاستخدام
- تسلسل الأحداث: حيِّز خاص للتعبير عن الأحداث التي يتم تنفيذها أثناء محاولة الفاعلين والنظام الوصول بحالة الاستخدام إلى الهدف المرجو تحقيقه، بحيث يمكن أن تتضمن تلك الأحداث تفاعلات النظام مع المستخدم أو المناقلات الضمنية التي يتم تنفيذها
- الشروط اللاحقة: حيّز خاص للتعبير عن الشروط التي تتحقق بعد الانتهاء من تنفيذ حالة الاستخدام بنجاح.



- يمكننا التمييز بين نوعين أساسيين من أساليب تدوين حالات الاستخدام، وهما: تدوين حالات الاستخدام البسيطة، وتدوين حالات الاستخدام المتقدّمة بحيث يتضمن التدوين الأول كل من اسم حالة الاستخدام ورمزها وفاعليها ووصف أولي بسيط عن أهدافها والغرض منها، في حين يتضمن التدوين الآخر معلومات أكثر تفصيلاً عن حالة الاستخدام، كالفاعلين الأوليين والفاعلين الثانويين وتسلسل الأحداث والشروط المسبقة واللاحقة، بالإضافة إلى حلول بديلة عند الحاجة وغيرها ... سنقوم بكتابة تدوين لحالة استخدام متقدّمة في الشرائح التالية.

تحليل نموذج حالات الاستخدام

- يمكن تعريف النظام من خلال مجموع حالات الاستخدام التي يتضمنها
- فعندما نعمل على توصيف جزء من النظام من خلال حالة استخدام محددة، من المحتمل أن تتوسع حالة الاستخدام وتزداد تعقيداً، ويبدو ذلك جلياً في الحالات التي تتطلب تكراراً للأحداث في النظام، وبالتالي يمكننا التعبير عن نفس تسلسل الأحداث في النظام من خلال حالة استخدام وحيدة أو من خلال حالة استخدام مجزأة إلى عدة أقسام، وذلك من منطلق التقسيم الوظيفي الذي يُعتبر واحداً من المبادئ الشهيرة في هندسة البرمجيات
- تُستخدم علاقات الاحتواء <<include>> والتمديد <<extend>> كأدوات لإعادة تحليل أو بناء نموذج حالات الاستخدام بغرض التعامل مع مفهوم التكرارية وعزل العمليات المشتركة بين حالات الاستخدام
- تسمح علاقة الاحتواء باستخراج السلوك المشترك في حالة الاستخدام المحتواة، وتعطي علاقة التمديد صيغة مضبوطة لتمديد سلوك

المنارة للاستشارات
www.manaraa.com

حالة استخدام معينة بتنشيط حالة استخدام أخرى. وتختلف علاقة الاحتواء عن علاقة التمديد من حيث أن حالة الاستخدام "المحتواة" ضرورية لإكمال حالة الاستخدام التي جرى تنشيطها.

- ومن الناحية العملية تؤدي الحاجة إلى الكثير من الجهود لاكتشاف العلاقات بين حالات الاستخدام وتحديد أنواع العلاقات الموجودة بين أزواج الحالات، إلى ظهور مشكلات هامة في وجه تطوير المشاريع. بالإضافة لذلك تترابط حالات الاستخدام في المستوى الأعلى بعلاقات قد تسيطر على المخطط وتوجه الاهتمام نحو العلاقات بين حالات الاستخدام بدلاً من توجيهه نحو تحديد حالات الاستخدام.

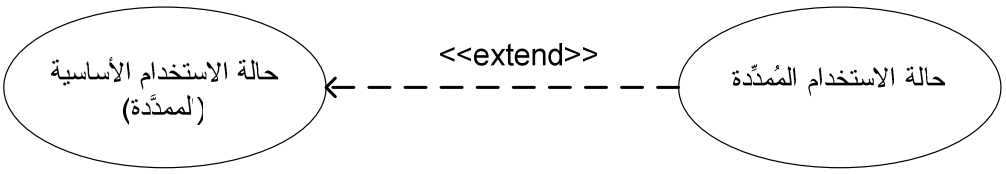
- سنقوم في الشرائح التالية بشرح كل من العلاقتين السابقتين بالتفصيل، مع التطرق إلى كيفية تضمين هاتين العلاقتين في مخططات حالات الاستخدام.

علاقة التمديد

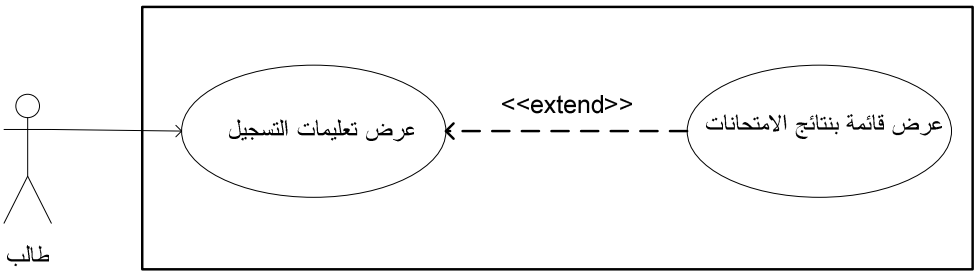
- تسمح علاقة التمديد بتوسيع حالة الاستخدام بسلوك أو تبدلات إضافية

- تظهر أهمية علاقة التمديد عندما تمتلك حالة استخدام محددة سلوك حالة استخدام أخرى بشكل كامل، إلا أنها تتميز بأنه ليس من الضروري أن يتم تنفيذها لكي تستكمل حالة الاستخدام الأساسية تنفيذها، لنفترض على سبيل المثال، أننا نتحدث عن نظام تسجيل جامعي، وأنه يمكن للفاعل "طالب" أن يقوم بالتسجيل على مجموعة من مواد الفصل الدراسي الجديد بحيث تظهر قائمة بنتائج الامتحانات السابقة قبل أن يصبح بالإمكان اختيار المواد المرغوبة، وبعد ذلك تظهر تعليمات التسجيل. وبافتراض وجود كل من حالتي الاستخدام "عرض تعليمات التسجيل" و"عرض قائمة بنتائج الامتحانات"، فإنه قد تمدد حالة الاستخدام "عرض قائمة بنتائج الامتحانات" حالة الاستخدام "عرض تعليمات التسجيل"، لكن ليس من الضروري أن يحصل هذا التمديد دوماً، فبالنسبة للطلاب الجدد مثلاً لا توجد نتائج امتحانات
- يمكن أن تمتلك حالة الاستخدام عدّة علاقات تمديد مختلفة، كما أنه يمكن لحالة الاستخدام الممدَّدة أن تختار الوقت المناسب لتنفيذ الأنشطة المتاحة من خلال حالة الاستخدام الفرعية التي أجرت عملية التمديد

- يُشار إلى علاقة التمديد بين حالتي استخدام من خلال خط منقّط وموجَّه من حالة الاستخدام التي تزوِّد عملية التمديد إلى حالة الاستخدام الأساسية الممدَّدة

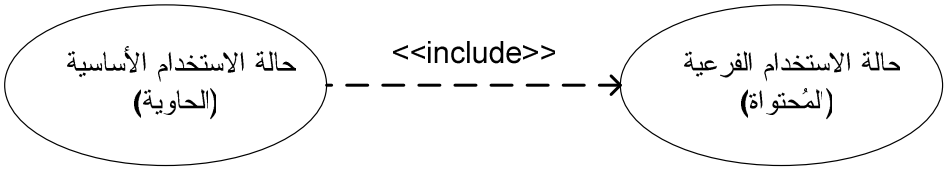


المنارة للاستشارات
www.manaraa.com

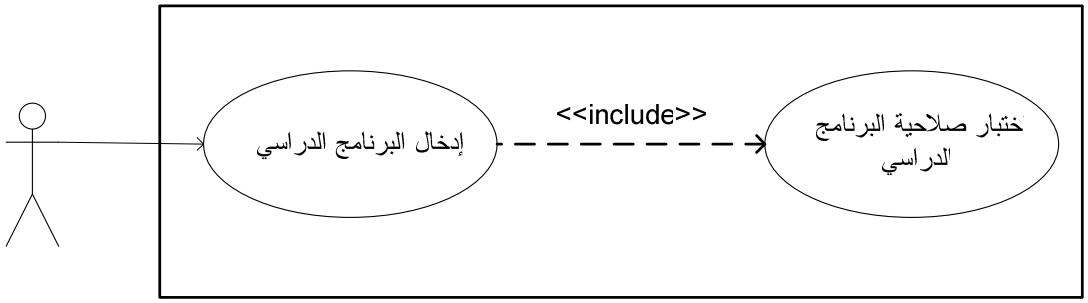


علاقة الاحتواء

- تسمح علاقة الاحتواء لحالة الاستخدام بالولوج إلى مجموعة من العمليات التي تشكل إجراءً مستقلاً ضمن حالة استخدام مستقلّة
- عندما توجد علاقة احتواء بين حالتي استخدام، تصبح عملية تنفيذ حالة الاستخدام الرئيسية مرتبطة مباشرة بتنفيذ حالة الاستخدام الفرعيّة
- مثال: تنطبق صفة الاحتواء <<include>> على العلاقة بين حالة الاستخدام "إدخال البرنامج الدراسي" وحالة الاستخدام "اختبار صلاحية البرنامج الدراسي". وتعني هذه العلاقة أن الحالة الأولى تحوي دوماً الحالة الثانية، فكلما تم إدخال برنامج دراسي يجري التحقق من صلاحيته من حيث توافقه مع قيود جدول الحصص الزمني وغيرها من القيود الأخرى
- يُشار إلى علاقة الاحتواء بين حالتي استخدام من خلال خط منقّط وموجَّه من حالة الاستخدام الأساسية الحاوية، إلى حالة الاستخدام الفرعية المُحتواة



مثال:



المنارة للاستشارات
www.manaraa.com

علاقة التعميم

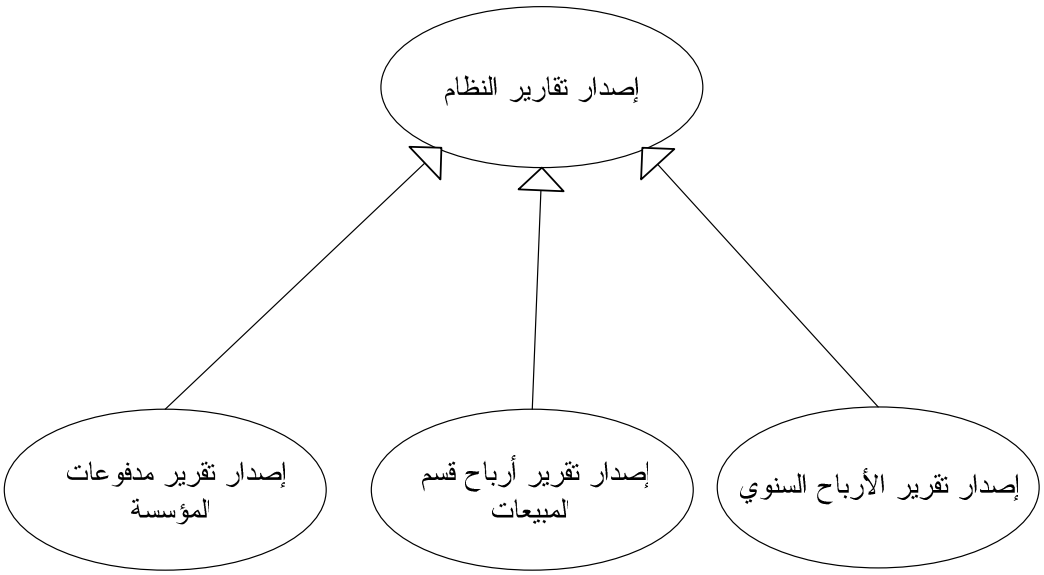
- تعرِّف لغة النمذجة الموحّدة UML علاقة التعميم بأنها علاقة بين حالتي استخدام أب وابن، بحيث تتضمن حالة الاستخدام الابن كافة واصفات الأب وخصائص سلوكه، كما تشترك في كافة ارتباطات وعلاقات حالة الاستخدام الأب

- تعتبر حالة الاستخدام الابن في علاقة التعميم، أكثر تخصيصاً من حالة الاستخدام الأب، إذ ترث هذه الحالة كافة خصائص الأب وواصفاته، وتقوم بتعديل بعضها أو إضافة البعض الآخر

- يُطلق على حالة الاستخدام الأب اسم حالة الاستخدام المُجرَّدة، كما ويتم الإشارة إلى ذلك ضمن التدوين الخاص بوصفها

- يُشار إلى علاقة التعميم بين حالتي استخدام من خلال خط موجّه من حالة الاستخدام الابن إلى حالة الاستخدام الأب بحيث يكون رأس السهم بشكل مثلث مغلق

- مثال: لنفترض أنه لدينا نظام محاسبة يقوم بإصدار تقارير، وأنه لدينا كل من حالات الاستخدام التالية: "إصدار تقرير الأرباح السنوي"، "إصدار تقرير أرباح قسم المبيعات" "إصدار تقرير مدفوعات المؤسسة"، يمكن أن ترث كافة حالات الاستخدام المذكورة سلوك وواصفات حالة استخدام أخرى معممة يُطلق عليها اسم "إصدار تقارير النظام":



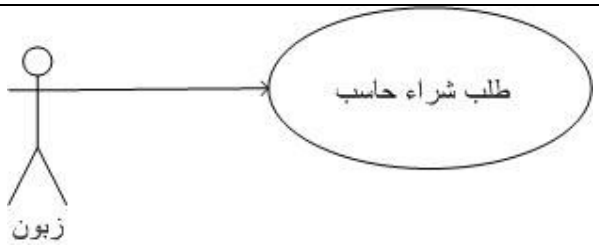
تدوين حالات الاستخدام المتقدمة

- مرّ معنا سابقاً أن تدوين حالة الاستخدام المتقدمة هو طريقة أكثر تفصيلاً في عملية تدوين وتوثيق حالات الاستخدام مقارنة مع الطريقة المتبعة مع حالات الاستخدام البسيطة

- سنتعرض مباشرةً فيما يلي لمثال عملي يصف طريقة توصيف حالة استخدام متقدّمة يتم من خلالها إجراء عملية شراء إلكتروني

المنارة للاستشارات
www.manaraa.com

لحاسب:



اسم حالة الاستخدام	طلب شراء حاسب		
رقم معرِّف حالة الاستخدام	UC-01		
وصف موجز	تسمح هذه الحالة للزبون بإدخال طلب الشراء. ويتضمن ذلك تزويد النظام بعنوان الشحن والفاتورة بالإضافة إلى تفاصيل طريقة الدفع.		
الفاعل (أو الفاعلون) الأولي	الزبون		
الفاعل (الفاعلون) الثانوي	---		
الشروط السابقة	يوجه الزبون أحد برامج تصفح الإنترنت إلى صفحة الويب الخاصة بالمصنع. تعرض الصفحة معلومات تفصيلية عن مكونات الحاسوب إلى جانب سعره.		
التدفق الرئيسي		**المستخدم**	**النظام**
	.1	تبدأ حالة الاستخدام هذه عندما يقرر الزبون أن يطلب شراء الحاسوب بانتقاء الوظيفة عند ظهور تفاصيل الطلب على الشاشة.	
	.2		يطلب النظام من الزبون أن يدخل معلومات تفصيلية تتضمن: اسم مندوب المبيعات (إذا كان معروفاً)، تفاصيل الشحن (اسم الزبون وعنوانه). تفاصيل الفاتورة (إذا كانت مختلفة عن تفاصيل الشحن). طريقة الدفع (شيك أو بطاقة ائتمان) وأية تعليقات أخرى.
	.3	يختار الزبون الوظيفة "شراء" لإرسال الطلب إلى المصنّع، وذلك من خلال النقر على الزر المناسب في الواجهة التخاطبية.	
	.4		يسند النظام لطلب الشراء رقماً وحيداً مميزاً ورقم حاسب الزبون ويخزن معلومات الطلب في قاعدة المعطيات.
	.5		يرسل النظام رقم الطلب ورقم الزبون إلى الزبون بواسطة البريد الإلكتروني، كتأكيد على قبول طلب الشراء.

المنارة للاستشارات
www.manaraa.com

التدفقات البديلة	– ينشط الزبون وظيفة الشراء Purchase قبل إدخال كل المعلومات الضرورية. فيعرض النظام عندئذ رسالة خطأ ويطلب من الزبون إتمام المعلومات الناقصة. – ينتقي الزبون الوظيفة Reset (أو وظيفة باسم مشابه) للعودة إلى طلب شراء فارغ، فيسمح النظام للزبون بإدخال المعلومات من جديد.
الشروط اللاحقة	إذا اكتملت حالة الاستخدام بنجاح يسجل طلب الشراء في قاعدة معطيات النظام، وإلا فتبقى حالة النظام كما هي دون تغيير.

مثال تطبيقي

- ليكن لدينا نظام مكتبة تابع للجامعة ويستخدم لإدارة الكتب وإعارتها للقرّاء
- يدير نظام المكتبة شخص يطلق عليه اسم أمين الكتبة
- يمكن أن يكون المُستعير طالباً من طلاب الكلّية أو موظفاً في الجامعة
- لا يمكن السماح لأي مُستعير بأن يأخذ أي كتاب ما لم يقم أمين المكتبة بمسح رمز الكتاب بواسطة جهاز القارئ الآلي المتوفر في المكتبة وذلك بغرض تسجيل طلب الإعارة للكتاب المحدد
- يختبر النظام إمكانية إعارة الكتاب المُختار من خلال اختبار فيما إذا تجاوز المستعير الحد المتاح له، أي عدد الكتب التي يُسمح بإعارتها للقارئ. فإذا ما تخطى عدد الكتب الحد المسموح به، يقوم النظام بعرض رسالة إعلام مناسبة ولا يسمح بإعارة المزيد من الكتب لذلك المستعير
- يمثل السيناريو التالي عرضاً لأحداث عملية الاستعارة:
  - بعد أن يقوم المستعير باختيار الكتب التي يرغب باستعارتها، يقوم بتحديد أرقام (أو رموز) تلك الكتب وأخذها إلى مكتب أمين المكتبة في صالة الإعارة
  - يقوم أمين المكتبة بمسح أرقام الكتب باستخدام جهاز قارئ الباركود
  - يختبر النظام إمكانية الإعارة من خلال اختبار عدد الكتب التي استعارها القارئ، كما يختبر فيما إذا كان المستعير قد أودع في حسابه مبلغ 500 ل.س. كتأمين عن الكتب التي يقوم باستعارتها
- لتسهيل عملية البحث عن كتاب محدد، يستطيع المستعير أن يقوم باستخدام نظام إدارة المكتبة للبحث عن الكتب من خلال اسم الكتاب أو اسم المؤلف أو رقم الكتاب.

حل المثال التطبيقي

–البحث عن الفاعلين–

سنقوم من خلال هذه الشريحة بإعادة قراءة نص التمرين واستنتاج الفاعلين منه:

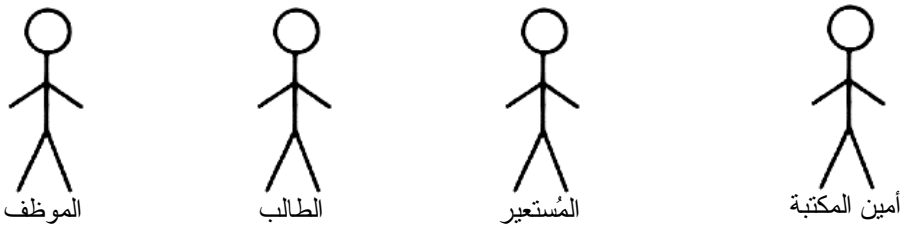
- ليكن لدينا نظام مكتبة تابع للجامعة ويستخدم لإدارة الكتب وإعارتها للقرّاء
- يدير نظام المكتبة شخص يطلق عليه اسم **أمين الكتبة**
- يمكن أن يكون المُستعير **طالباً** من طلاب الكلّية أو **موظفاً** في الجامعة
- لا يمكن السماح لأي **مُستعير** بأن يأخذ أي كتاب ما لم يقم أمين المكتبة بمسح رمز الكتاب بواسطة جهاز القارئ الآلي المتوفر في المكتبة وذلك بغرض تسجيل طلب الإعارة للكتاب المحدد
- يختبر النظام إمكانية إعارة الكتاب المُختار من خلال اختبار فيما إذا تجاوز **المستعير** الحد المتاح له، أي عدد الكتب التي يُسمح

المنارة للاستشارات
www.manaraa.com

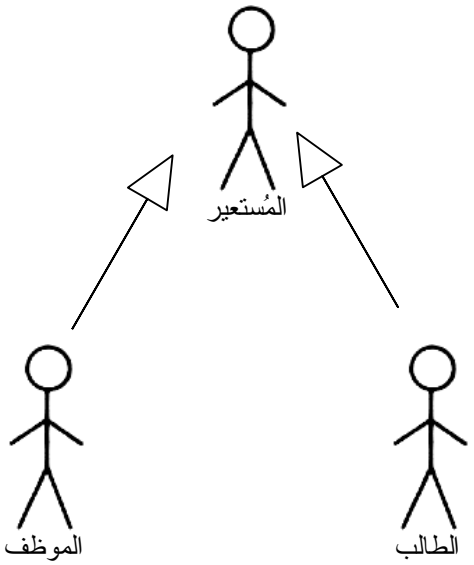
بإعارتها **للقارئ**. فإذا ما تخطى عدد الكتب الحد المسموح به، يقوم النظام بعرض رسالة إعلام مناسبة ولا يسمح بإعارة المزيد من الكتب لذلك المستعير

- يمثل السيناريو التالي عرضاً لأحداث عملية الاستعارة:
  - بعد أن يقوم **المستعير** باختيار الكتب التي يرغب باستعارتها، يقوم بتحديد أرقام (أو رموز) تلك الكتب وأخذها إلى مكتب **أمين المكتبة** في صالة الإعارة
  - يقوم أمين المكتبة بمسح أرقام الكتب باستخدام جهاز قارئ الباركود
  - يختبر النظام إمكانية الإعارة من خلال اختبار عدد الكتب التي استعارها **القارئ**، كما يختبر فيما إذا كان **المستعير** قد أودع في حسابه مبلغ 500 ل.س. كتأمين عن الكتب التي يقوم باستعارتها
- لتسهيل عملية البحث عن كتاب محدد، يستطيع **المستعير** أن يقوم باستخدام نظام إدارة المكتبة للبحث عن الكتب من خلال اسم الكتاب أو اسم المؤلف أو رقم الكتاب.

يمكننا بعد تفحص النص السابق أن نستنتج فاعلي النظام، وهم:



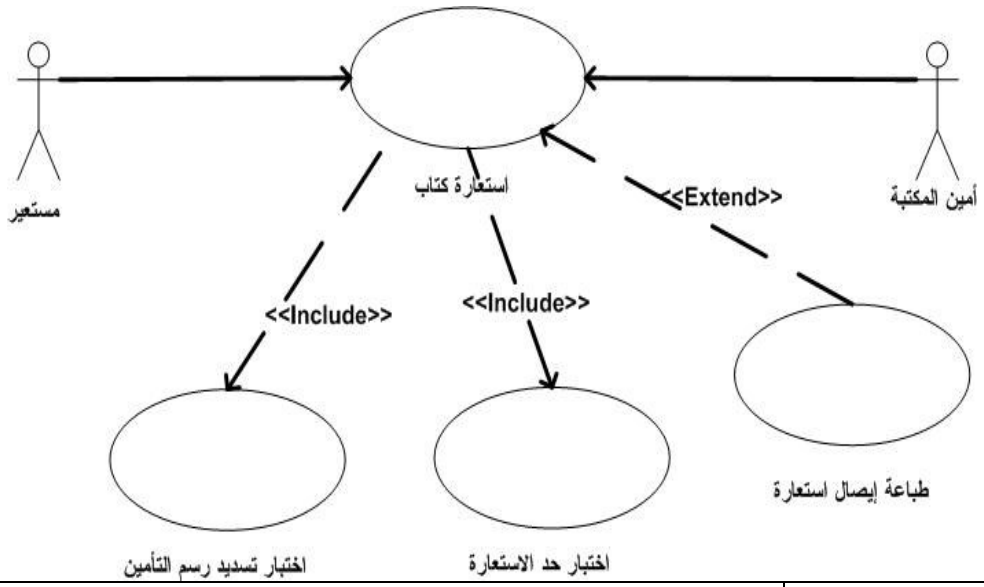
كما ويمكننا أن نستنتج وجود علاقة تعميم بين الفاعلين المستنتجين:



المنارة للاستشارات
www.manaraa.com

حل المثال التطبيقي

-حالة الاستخدام استعارة كتاب-



اسم حالة الاستخدام	استعارة كتاب		
رقم معرِّف حالة الاستخدام	UC-01		
وصف موجز	يذهب المستعير إلى المكتبة لاستعارة كتاب، ويقوم بجلب الكتب إلى أمين المكتبة يطلب أمين المكتبة رقم بطاقة المستعير يقوم المستعير بتقديم بطاقته الخاصة (بطاقة الطالب أو بطاقة الموظف) بالإضافة إلى الكتب التي يرغب باستعارتها، يقوم أمين المكتبة بمسح أرقام الكتب المختارة بواسطة القارئ الآلي لإنشاء طلب استعارة للمستعير.		
الفاعل (أو الفاعلون) الأولي	المستعير (المُنشئ)، أمين المكتبة.		
الفاعل (الفاعلون) الثانوي	---		
الشروط السابقة	أن يمتلك المستعير بطاقة ورقم يخول له إجراء عملية الاستعارة أن يكون للمستعير سجلّ سابق في النظام.		
التدفق الرئيسي	**المستخدم**		**النظام**
	يجلب المستعير الكتب إلى أمين المكتبة، ويقدم بطاقته الخاصة	.1	
	يمسح أمين المكتبة رقم بطاقة المستعير	.2	
			يختبر النظام فيما إذا كان للمستعير سجل معرّف، وفيما إذا كان هذا المستعير قد أودع

المنارة للاستشارات
www.manaraa.com

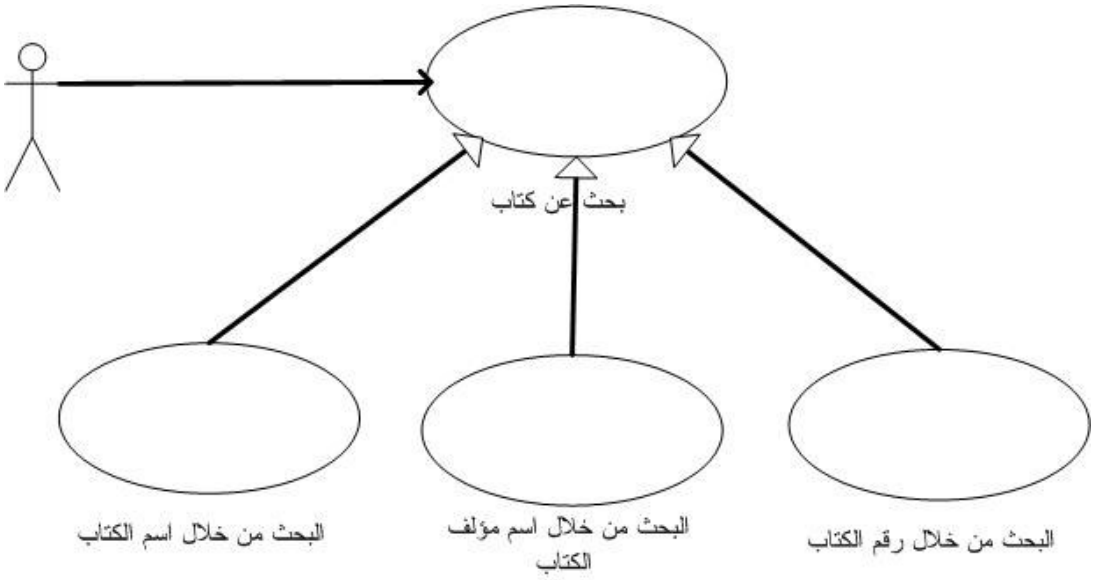
	.3		رسم التأمين الذي تطلبه المكتبة من قرائها
	.4		يختبر النظام فيما إذا كان المستعير قد تجاوز الحد المسموح له بالاستعارة، أي عدد الكتب الأعظمي التي يمكنه استعارتها في نفس الوقت
	.5	يقوم أمين المكتبة بمسح أرقام الكتب الجديدة التي يرغب المستعير باستعارتها	
	.6		يختبر النظام فيما إذا تجاوز المستعير حد الكتب المتاحة له
	.7		يجعل النظام حالة الكتب التي تمت قراءة أرقامها على أنها "مُعارة" للقارئ المحدد
	.8		يطبع النظام إيصالاً مناسباً بتاريخ الاستعارة والتاريخ الذي ينبغي فيه إعادة الكتب إلى المكتبة.
التدفقات البديلة	1) في حال عدم وجود رسم تأمين، يطلب أمين المكتبة من المستعير أن يسدد رسم التأمين المطلوب، ويتم إلغاء حالة الاستخدام. 2) يقوم النظام بإعلام أمين المكتبة بأن الحد المسموح به من عدد الكتب التي ينبغي إعارتها لهذا المستعير قد تم تجاوزه، ويتم إلغاء حالة الاستخدام.		
الشروط اللاحقة	---		

حل المثال التطبيقي

-حالة الاستخدام البحث عن كتاب-

نلاحظ في النظام وجود حالة استخدام مجرّدة وهي حالة الاستخدام "بحث عن كتاب"، والتي تعتبر أباً لثلاثة حالات استخدام وهي: "البحث من خلال اسم الكتاب"، "البحث من خلال اسم مؤلف الكتاب"، "البحث من خلال رقم الكتاب":

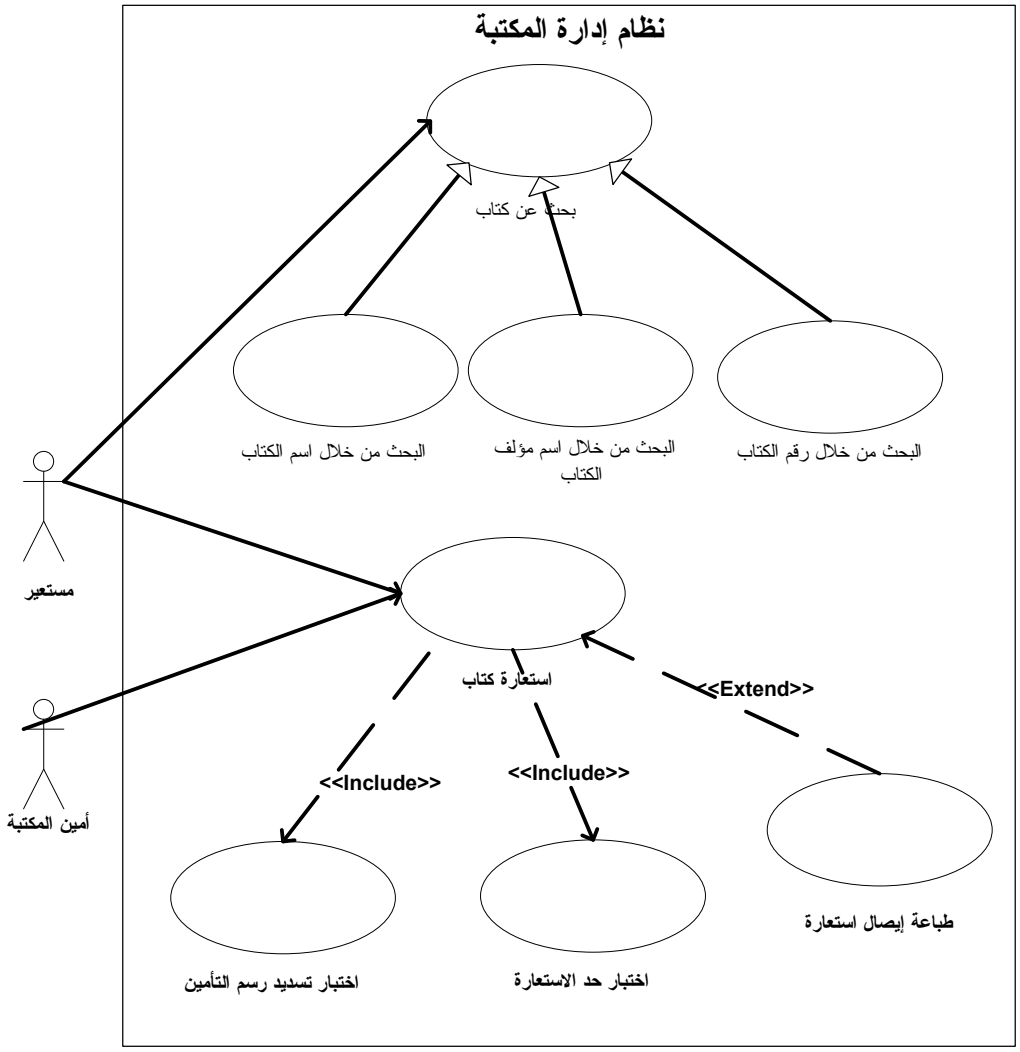
المنارة للاستشارات
www.manaraa.com



حل المثال التطبيقي

فيما يلي عرض لمخطط حالات الاستخدام الكلي الناتج عن المثال التطبيقي، ونترك للطالب مهمة استنتاج جداول حالات الاستخدام المتقدمة لبقية حالات الاستخدام الرئيسية والمحتواة و الممددة:

www.manaraa.com



ملاحظات

- لا يمكن وضع فاعل لا يقوم بتفعيل أي حالة استخدام
- يجب عدم الإكثار من حالات الاستخدام بحيث يبقى المخطط مقروءاً مع الانتباه إلى شمل جميع الحالات، مع ضرورة الأخذ بعين الاعتبار لمبدأ التقسيم الوظيفي الذي تحدثنا عنه في شرائح سابقة
- تجنب تسمية الفاعلين بأسماء عادية
- تسمية حالات الاستخدام بجمل فعلية
- يجب عدم إظهار أي تسلسل زمني عند رسم المخطط.

المنارة للاستشارات
www.manaraa.com

تمرين

- ترغب إدارة مستشفى بتطوير نظام محاسبة يقوم على شقين أساسيين بحيث يرتبط الأول بتكاليف المرضى في حين يتعلق الثاني برواتب الموظفين بما تشمله من مكافآت أو عقوبات
- يمكن أن تشمل تكاليف المرضى ما يلي:
  - رسوم دخول
  - رسوم إقامة
  - أسعار أدوية
  - رسوم أطباء
- وتشمل رواتب الموظفين ما يلي:
  - راتب أساسي
  - مكافآت

– ويقتطع منه:

  - ضريبة دخل
  - أسعار وجبات
- يقوم موظف معتمد بإدخال معلومات المرضى، في حين يقوم كل موظف بإدخال خياراته عند طلب الوجبة التي يرغب بها
- يُصدر النظام تقارير مفصلة بالرواتب والفواتير

المطلوب: رسم مخطط حالات الاستخدام المناسب، مع توصيف حالات الاستخدام المتقدمة وفق التدوين الأنسب لكل منها.

المنارة للاستشارات
www.manaraa.com

القسم التاسع

مبادئ الهندسة غرضية التوجه

الكلمات المفتاحية:

غرضية التوجه، الغرض، الصف، مميز هوية الغرض، الطرائق، الوراثة، الكبسلة، هرمية الصفوف، تعددية الأشكال، إعادة تعريف الطرائق.

ملخص:

يُركِّز هذا الفصل على المفاهيم الأساسية للتقانة غرضية التوجه.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- التعرف على الأغراض واصفاتها وطرائقها
- تعريف الصفوف والعلاقات بينها
- الوراثة الأحادية والمتعددة
- إعادة التعريف وتعددية الأشكال
- العلاقات بين الأغراض

المنارة للاستشارات
www.manaraa.com

مقدمة

إن البرمجة غرضية التوجه هي طريقة بديلة عن البرمجة التقليدية، وهي مُعتمِدة على أغراض. حيث يُمثِّل كل غرض كيان من العالم الحقيقي مُميَّز الهوية مع واصفات مميزة، وإمكانية العمل بنفسه والتفاعل مع الأغراض الأخرى.

يتم تعريف الغرض من خلال مُميِّز الهوية الخاص به، حيث يُسنَد للغرض لحظة بنائه ولايمكن تغييره أبداً، ويُحذَف لحظة حذف الغرض.

يتم تحديد كل غرض من خلال مجموعة من الواصفات، وتتميز النسخ المختلفة من الغرض عن طريق القيم المختلفة للواصفات، حيث تملك كل نسخة قيم مختلفة لهذه الواصفات.

أساسيات التقانة غرضية التوجه

قد يكون إجراء محاكاة مع أغراض حسية من الحياة الواقعية طريقة جيدة لشرح وتوضيح المفاهيم غرضية التوجه، إذ يتألف العالم من حولنا من أغراض يكون كل منها في حالة محددة تحددها القيم الحالية لصفات الغرض.

فعلى سبيل المثال يتواجد فنجان القهوة على مكتبي في الحالة "مملوء" لأنه مصمم بحيث يستوعب السوائل ومازالت القهوة موجودة فيه، وعندما لا تبقى هناك قهوة في الفنجان يصبح في الحالة "فارغ" وإذا سقط على الأرض وتحطم سيصبح في الحالة "مكسور".

إلا أن فنجان القهوة كائن سلبي، فهو لا يتميز بسلوك خاص (behavior)، بالمقابل لا يمكن قول الأمر نفسه بالنسبة لكلب أو شجرة، فالكلب ينبح، والشجرة تنمو، وللأغراض الحقيقية عادة سلوك.

لكل غرض من الأغراض الحقيقية هوية (identity)، وهي خاصة ثابتة نميز بواسطتها بين غرض وآخر، فإذا كان على مكتبي فنجانا قهوة من المجموعة نفسها يمكنني القول إن الفنجانين متساويان لكنهما غير متطابقين، فهما متساويان لأن لهما قيم الخصائص نفسها و لهما الشكل والحجم نفسه، وكلاهما أسود مثلاً، لكنهما ليسا متطابقين في اللغة غرضية التوجه، لأن هناك اثنان يمكنني أن أستخدم أياً منهما وفقا لرغبتي.

الغرض

الغرض هو مثل من "شيء"، فقد يكون من أمثلة عديدة للشيء نفسه، ففنجان القهوة الموجود لدي هو مثل من مجموعة الفناجين الموجودة.

يتألف النظام غرضي التوجه من مجموعة أغراض متعاونة، فكل شيء في النظام غرضي التوجه هو عبارة عن غرض.

يُدون الغرض في لغة UML كمستطيل من جزأين، يحوي الجزء الأعلى اسم الغرض واسم الصف الذي ينتمي إليه الغرض، ويعبر عن ذلك بالصيغة: (ObjectName: ClassName).

ويحوي الجزء الأدنى قائمة بأسماء وقيم صفات الغرض، كما قد تظهر أيضاً أنماط الخصائص بالصيغة التالية: (AttributeName: type = value).

من المهم أن نشير هنا إلى أن تدوين الغرض لا يحوي جزءاً للعمليات (Operations) التي يمكن أن ينفذها الغرض، ويعود هذا إلى كون العمليات متطابقة في كل الأمثال ولن يكون تكرار ذكرها في كل مثل مناسباً. لذلك يجري تخزين العمليات على مستوى الصف.

واصفات الغرض

يمكن أن تُشيِر واصفات الغرض إلى غرض أو عدة أغراض أخرى، كما يمكن أن تأخذ قيمة واحدة أو عدة قيم.

يُستخدَم مميز هوية الغرض من أجل الربط بين غرضين، فمثلاً لدينا الغرض (طالب) وواصفة لهذا الغرض (المواد الدراسية) تشير إلى مجموعة من المواد. عندها تحوي الواصفة (المواد الدراسية) على مميز هوية غرض يحوي على قائمة من أغراض المواد، وبذلك يتم

المنارة للاستشارات
www.manaraa.com

الربط بين الغرضين.

طرائق الغرض

الطريقة: هي عبارة عن رماز يستخدم لتنفيذ عملية معينة على واصفات الغرض، حيث يتم تحقيق كل العمليات التي يمكن تنفيذها على الغرض من خلال طرائق. فالطريقة هي التي تحمي واصفات الغرض من الوصول إليها.

يتم طلب تنفيذ طريقة معينة من غرض معين من خلال إرسال رسالة إلى الغرض تحوي اسم الطريقة والمعاملات المطلوبة لتنفيذها، حيث يقوم الغرض المعني بتنفيذ الطريقة وإعادة النتيجة في حال وجودها.

من الواضح أن أي غرض لا يمكن أن يصل للبنية الداخلية لغرض آخر، إنما يتم التخاطب بينهما عن طريق رسائل تتضمن طلب تنفيذ طرائق معينة. وهذا ما يدعى بمفهوم الكبسلة أي إمكانية إخفاء البنية الداخلية للغرض (الواصفات والطرائق) عن الأغراض الأخرى.

الصفوف

يتم تجميع الأغراض التي تمتلك معطيات متشابهة في صفوف، فالصف هو عبارة عن مجموعة من الأغراض المتشابهة مع بنية متشابهة (واصفات) وسلوك متشابه (طرائق). فالصف هو الواصف لمجموعة من الأغراض لها الصفات نفسها والعمليات نفسها، وهو يلعب بذلك دور قالب لإنشاء الأغراض.

يحتوي كل غرض يجري إنشاؤه بهذا القالب، على قيم الصفات التي تتوافق مع الأنماط المعرفة ضمن تعريف الصف، كما يستطيع أي غرض أن يستدعي العمليات المعرفة في صفه.

يحتوي الصف على وصف لبنية المعطيات وتفصيل تحقيق الطرائق للأغراض في ذلك الصف.

يُدعى كل غرض من الصف بـــ(نسخة الصف)، وكل نسخة تملك مميز هوية فريد، كما أنها تعرف الصف الذي تنتمي إليه.

هنالك نوعان لطرائق الصف، طرائق عامة وطرائق خاصة، حيث يمكن لأغراض أخرى أن تطلب تنفيذ الطرائق العامة فقط، بينما لا يمكنها أبداً طلب تنفيذ الطرائق الخاصة.

مثال 1

- نستعرض في هذا المثال الصف Point الذي يحتوي على واصفتين X و Y تمثلان إحداثيات النقطة، والطريقة Distance لحساب البعد بين نقطتين، والطريقة Equals لمقارنة نقطتين (متساويتين أم لا).

```
CLASS Point {
//variables
        ATTRIBUTE Real X;
                                ATTRIBUTE Real Y;
//methods
        Float Distance (IN Point aPoint);
        //computes the distance between two points
        Float Equals (IN Point aPoint);
        //determines if two points have the same coordinates
}
```


المنارة للاستشارات
www.manaraa.com

- سنقوم باستخدام الصف Point في بناء الصف Rectangle الذي يحتوي على واصفتين من النمط Point (UpperLeftCorner و LowerRightCorner) بالإضافة إلى الطرائق (Area و Length و Height).

```
CLASS Rectangle {
//variables
        ATTRIBUTE Point UpperLeftCorner;
        ATTRIBUTE Point LowerRightCorner;
//methods
        Float Area ();
        //computes the area of the rectangle
        Float Length ();
        //compute the length
        Float Height ();
        //compute the height
}
```

علاقات الصفوف

يتم تنظيم الصفوف في بنية هرمية، حيث يكون لكل صف أب وحيد.

والصف الأعلى: هو عبارة عن تصنيف أعم للصفوف الجزئية منه.

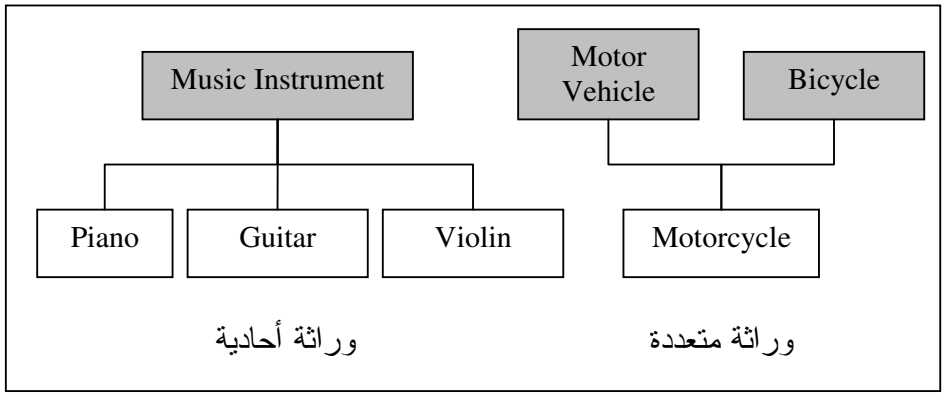
أما الصفوف الجزئية: فتحتوي على مركبات مخصصة من التصنيف الأعم للصف الأعلى.

مثال: الصف (أداة موسيقية) هو صف أعلى للصفوف (بيانو، غيتار، فيولون)، وبالتالي فإن الصفوف الأخيرة هي صفوف جزئية من صف الأداة الموسيقية.

جميع الصفوف في الهرمية موروثة من الصف الجذر للهرمية.

هنالك نوعان من الوراثة:

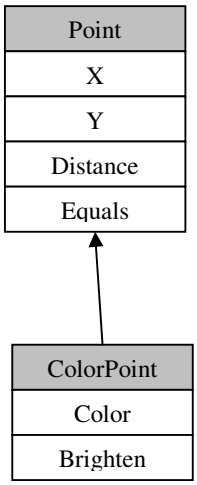
- وراثة أحادية: وهي موجودة عندما يكون للصف أب واحد فقط (صف أعلى)، وعندما يُرسِل النظام طلب تنفيذ طريقة معينة إلى غرض معين يتم البحث أولاً عن هذه الطريقة في الصف الذي ينتمي إليه الغرض ومن ثم في حال عدم وجودها يتم البحث في الصفوف الأعلى في الهرمية.
- وراثة متعددة: وهي موجودة عندما يكون للصف أكثر من أب واحد.



المنارة للاستشارات
www.manaraa.com

مثال 2

- نعود في هذا المثال إلى الصف Point وسنقوم بتطبيق مفهوم الوراثة لبناء صف جديد هو ColorPoint حيث يتضمن هذا الصف الجديد نفس واصفات وطرائق الصف Point، ولكن إضافة إليها الواصفة Color والطريقة Brighten.



- تعريف الصف ColorPoint

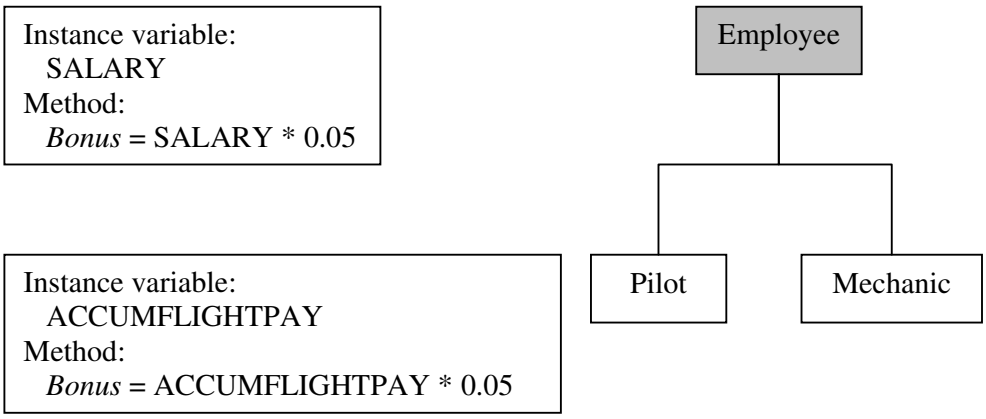
```
CLASS ColorPoint EXTENDS Point{
//variables
        ATTRIBUTE INTEGER Color;
        ATTRIBUTE Point LowerRightCorner;
//methods
        Integer Brighten ();
        //computes a new color that is brighter
}
```

إعادة تعريف الطرائق

يعد تعريف طريقة معرفة في الصف الأب، ضمن الصفوف الجزئية منه، وهذا ما يدعى إعادة التعريف.

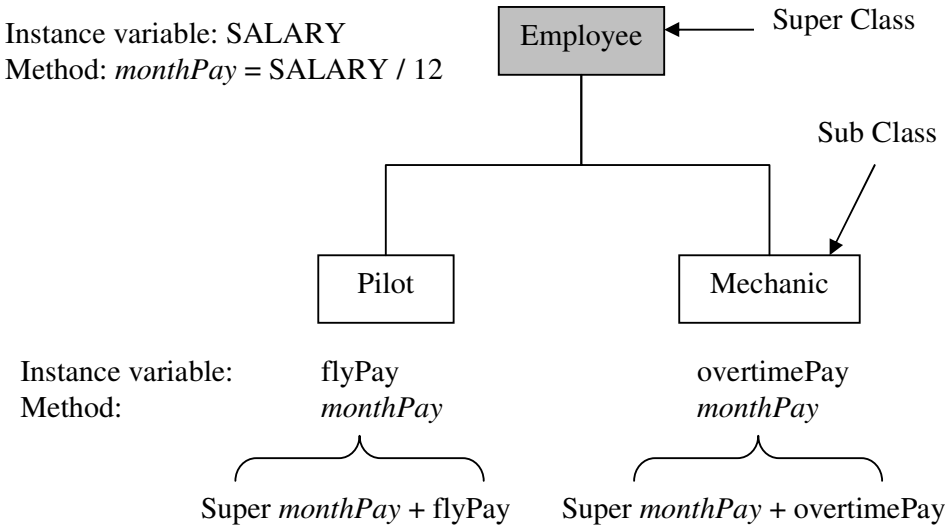
في مثال إعادة التعريف لدينا صف أب (موظف)، وصفوف جزئية منه (طيار وميكانيكي)، نلاحظ أنه أعدنا تعريف الطريقة (*Bonus*) في الصف طيار ولم نقم بإعادة تعريفها في الصف ميكانيكي، فجميع الموظفين لديهم طريقة واحدة في حساب المكافآت ماعدا الطيار لذلك قمنا بإعادة تعريفها في الصف الخاص به.

المنارة للاستشارات
www.manaraa.com



تعددية الأشكال

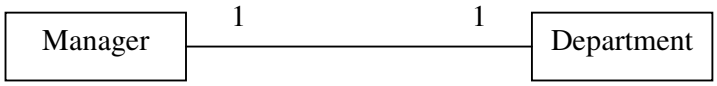
- تمكن تعددية الأشكال الغرض من السلوك بحسب معطياته الخاصة.
- مثال: بالعودة إلى نفس المثال السابق فإن حساب الراتب الشهري من خلال طلب نفس الطريقة (*monthPay*) من الصف *ميكانيكي* والصف *طيار* ولكن سيتم حسابها بطريقة مختلفة بكل صف وسيتم إعادة النتيجة الصحيحة.



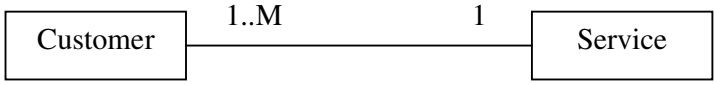
العلاقات بين الصفوف

- علاقة واحد لواحد (1:1): علاقة غرض لغرض. مثال كل مدير يرأس قسم واحد وكل قسم يرأسه مدير واحد.

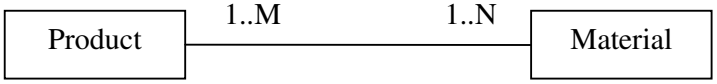
www.manaraa.com



- علاقة واحد لكثير (1:M): كل غرض من الصف الأول يرتبط بـــM غرض من الصف الثاني. مثال الموظف والخدمة، يعمل الموظف على خدمة واحدة، بينما تحوي الخدمة أكثر من موظف (علاقة 1:M).



- علاقة كثير لكثير (M:N): يرتبط كل غرض من الصف الأول بـــM غرض من الصف الثاني، وكذلك يرتبط كل غرض من الصف الثاني بـــN غرض من الصف الأول. مثال كل منتج يحتاج إلى مجموعة مواد أولية، وكل مادة أولية تُستخدَم في إنتاج أكثر من منتج.



www.manaraa.com

القسم العاشر والحادي عشر

مخططات الصفوف

الكلمات المفتاحية:

نماذج الحالة، نماذج السلوك، نماذج تغير الحالة، توصيف الحالة، حالة الغرض، صفوف الكيانات، صفوف التحكم، صفوف الحدود، نموذج الصفوف، منهج العبارات الاسمية. منهج عينات الصفوف المشتركة، المنهج المساق بحالات الاستخدام، منهج الصف – المسؤولية– المشاركون (CRC)، المنهج المختلط، صفة، عملية، اسم الصف، علاقات الاقتران، علاقة تركيب، علاقة تجميع، علاقة التعميم، الأنماط المحيطية، القيود، اللاحة، الملاحظات، الأمارات، نطاق الرؤية، التغليف، الوراثة، الخصائص المحمية، الصف الصديق، صف الاقتران، الصف المصنع، التعقيد، الوصلة، الحزمة.

ملخص:

تركز هذه الوحدة على التعرف على كيفية رسم مخططات الصفوف، حيث تلقي الضوء على مفاهيم توصيف المتطلبات الأساسية، ومفاهيم تعريف الصفوف، ونمذجتها وتوصيفها، بالإضافة إلى توصيف العلاقات المختلفة بين الصفوف ونمذجتها بلغة UML، كما تلقي هذه الوحدة الضوء على المفاهيم الأساسية في عملية نمذجة الصفوف المتقدمة.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- توصيف المتطلبات.
- توصيف الحالة
  - نمذجة الصفوف
    - مناهج اكتشاف الصفوف
    - توجيهات عامة لاكتشاف الصفوف
    - أمثلة عن اكتشاف الصفوف
  - توصيف الصفوف
    - تسمية الصفوف
    - اكتشاف وتوصيف صفات الصف
    - أمثلة لتوصيف الصفوف
  - نمذجة علاقات الاقتران
    - اكتشاف علاقات الاقتران
    - توصيف علاقات الاقتران
  - نمذجة علاقات التركيب والتجميع
    - اكتشاف علاقات التجميع والتركيب
    - توصيف علاقات التجميع والتركيب
    - أمثلة لتوصيف علاقات التجميع والتركيب
  - نمذجة علاقات التعميم
    - اكتشاف وتوصيف علاقات التعميم

المنارة للاستشارات
www.manaraa.com

- نمذجة الصفوف المتقدمة
  - الأنماط المحيطية – القيود
  - الملاحظات والأمارات
  - نطاق الرؤية والتغليف
    - رؤية الخصائص المحمية
    - رؤية خصائص الصف الموروثة
    - رؤية الخصائص الصديقة
  - صف الاقتران والصف المصنع
    - مثال
- طبقات الصفوف
  - تعقيد الشبكات
  - تعقيد البنى الهرمية
  - الحزمة (package)
- مثال – التسويق عبر الهاتف

المنارة للاستشارات
www.manaraa.com

توصيف المتطلبات

- تأخذ مرحلة توصيف المتطلبات كدخل لها متطلبات الزبون وتعطي بالنتيجة نماذج التوصيف، وتعطي هذه النماذج تعريفاً أكثر صورية لمناظر النظام المختلفة، وتأخذ هذه المرحلة بالحسبان الفئتين الرئيستين لمتطلبات الزبون وهما:
  1. متطلبات الوظائف
  2. متطلبات المعطيات.

- يمكن تصنيف نماذج التوصيف في ثلاث فئات:

1. نماذج الحالة: تمثل نماذج الحالة متطلبات المعطيات
2. نماذج السلوك: تغطي نماذج السلوك توصيفاً تفصيلياً لمتطلبات الوظائف
3. نماذج تغير الحالة: تغطي نماذج تغير الحالة نوعي المتطلبات المذكورين فهي توضح كيف تسبب الوظائف تغير المعطيات.

- تمثل النماذج بمخططات مدونة بلغة النمذجة المرئية- وهي في حالتنا هذه لغة UML، ويفيد كل مخطط عادة واحدة من غايات النمذجة الثلاث- الحالة، السلوك، أو تغير الحالة
  - الاستثناء الملحوظ هنا هو مخطط الصفوف الذي يوصف المفاهيم الثلاثة- أي حالات الأغراض وسلوكها وبشكل غير مباشر تغيرات حالات الأغراض.

- يركز كل مخطط على أحد مظاهر النظام، وتسمح هذه المخططات مجتمعة للمطورين والمستخدمين برؤية الحل المقترح من وجهات نظر مختلفة تركز كل منها على جوانب معينة وتتجاهل الجوانب الأخرى، ولا يمكن أن يعطي مخطط واحد تعريفاً كاملاً للنظام، ولا يمكن فهم النظام إلا من خلال تكامل مجموعة المخططات هذه.
- لا يجري بناء هذه المخططات وفق إجرائية تسلسلية برسم مخطط ثم الانتقال لرسم المخطط الآخر، بل يجري بناء هذه المخططات على التوازي وتُضاف التفاصيل في عمليات تكرارية متتالية. إن تحديد النماذج التي تشكل القوة المحركة للتطوير هو قرار شخصي يعتمد كثيراً على تفضيلات المحللين إلا إذا فُرضت على المطورين إجرائية معينة. ويفضل عادة تطوير أهم نموذجين وهما مخططات حالات الاستخدام ومخططات الصفوف على التوازي، حيث يعطي كل منهما أفكاراً تساعد في بناء الآخر.

توصيف الحالة

- **حالة الغرض:** تتحدد حالة غرض ما بقيم صفاته وعلاقات الاقتران مع الأغراض الأخرى
  - بما أن بنى المعطيات هي التي تحدد حالات الغرض ندعو نماذج بنى المعطيات بتوصيف الحالة

- **توصيف الحالة:** تصور توصيفات الحالة النظام من وجهة نظر سكونية (ولذلك تدعى نمذجة الحالات أحياناً النمذجة الساكنة):
  - تكون المهمة الأساسية هنا هي تعريف صفوف حقل التطبيق وصفاتها وعلاقاتها بالصفوف الأخرى
  - تترك عادة في هذه المرحلة عمليات الصفوف ليتم اشتقاقها لاحقاً من نماذج توصيف السلوك

- **أنواع الصفوف:**
  - صفوف الكيانات: هي الصفوف التي تعرف حقل التطبيق والتي سيكون لها وجود دائم في قاعدة معطيات النظام، وتدعى هذه الصفوف أحياناً "أغراض العمل"
  - صفوف التحكم: الصفوف التي تخدم أحداث النظام
  - صفوف الحدود : الصفوف التي تمثل واجهة الاستخدام البيانية

المنارة للاستشارات
www.manaraa.com

- **تعريف الصفوف:**
  - نبدأ، في الحالات النمطية، بتحديد صفوف الكيانات
  - لا يجري تعريف صفوف التحكم والحدود إلا بعد معرفة مميزات النظام السلوكية

توصيف الحالة

1 – نمذجة الصفوف

- **نموذج الصفوف:** يشكل نموذج الصفوف حجر الزاوية في تطوير النظام غرضي التوجه، إذ تشكل الصفوف الأساس اللازم لدراسة ومعاينة حالة النظام وسلوكه
  - من الصعب العثور على الصفوف بتسلسل زمني مباشر كما أن خصائص الصفوف ليست واضحة دوماً، فإذا لم يكن التطبيق تقليدياً لن يصل محللان مختلفان إلى المجموعة نفسها من الصفوف والخصائص

- **شروط نجاح نمذجة الصفوف:** إن نمذجة الصفوف ليست إجرائية دقيقة، ولا توجد وصفة جاهزة لكيفية العثور على صفوف جيدة وتعريفها، فالإجرائية تبقى تكرارية وتزايدية إلى حد بعيد. ويعتمد نجاح تصميم الصفوف إلى حد بعيد على مهارة المحلل في:
  - معرفة آليات نمذجة الصفوف
  - فهم حقل التطبيق
  - الخبرة في تصاميم مشابهة ناجحة
  - القدرة على التفكير والتنبؤ بالنتائج
  - الرغبة بمراجعة النموذج والتخلص من عيوبه

توصيف الحالة

1 – نمذجة الصفوف

(1) – مناهج اكتشاف الصفوف

- توجد هناك أربع مناهج شائعة الاستخدام لتحديد الصفوف وهي:
  1. العبارات الاسمية
  2. عينات الصفوف المشتركة
  3. المنهج المساق بحالات الاستخدام
  4. منهج الصف – المسؤولية– المشاركون أو CRC (Class- Responsibility- Collaborators)

المنارة للاستشارات
www.manaraa.com

توصيف الحالة

1 – نمذجة الصفوف

(1) – مناهج اكتشاف الصفوف

منهج العبارات الاسمية

- ينصح هذا المنهجُ المحللَ بقراءة عبارات وثيقة المتطلبات بحثاً عن العبارات الاسمية التي تظهر فيها، ويعتبر كل اسم صفاً محتملاً.

- في المرحلة الثانية تقسم قائمة الصفوف المحتملة إلى ثلاث مجموعات:

1. **صفوف ذات صلة بالموضوع:** هي تلك التي تنتمي بوضوح إلى حقل المسألة، ويتكرر ظهور الأسماء المعبرة عن هذه الصفوف بكثرة في وثيقة المتطلبات، كما يمكننا إدراك معاني هذه الصفوف وغاياتها من معرفتنا العامة بحقل التطبيق ومن تفحص أنظمة مشابهة وكتب ووثائق أخرى وحزم برمجية موجودة
2. **صفوف زائفة:** هي تلك الصفوف التي لا نستطيع أن نُجمِع على تصنيفها كصفوف ذات صلة بالموضوع ونحتاج لتحليلها ودراستها بعناية قبل أن ندرجها ضمن الصفوف ذات الصلة أو نستبعدها، والقرارات التي تتخذ بشأن هذه الصفوف هي التي تميز في النهاية بين نموذج صفوف جيد وآخر سيء
3. **صفوف غير ذات صلة بالموضوع:** هي تلك التي تقع خارج حقل المسألة، وغالباً ما يتجنب المصمم البارع تضمين هذه الصفوف في نماذجه

توصيف الحالة

1 – نمذجة الصفوف

(1) – مناهج اكتشاف الصفوف

منهج عينات الصفوف المشتركة

- يشتق هذا المنهج الصفوف المشاركة من نظرية تصنيف الأغراض العامة، وهي نظرية تعنى بتقسيم عالم الأغراض إلى مجموعات مفيدة بحيث نستطيع تبرير وجودها بطريقة أفضل.

- تساعد المجموعات (العينات) التالية المحلل في العثور على الصفوف:

1. صف المفهوم (Concept Class): المفهوم هو فكرة يتفق عليها عدد كبير من الناس، وبدونها لن يتمكن الناس من التواصل فيما بينهم بفاعلية، أو حتى من التواصل بدرجة مقنعة. فعلى سبيل المثال يعتبر الحجز Reservation صفاً مفهومياً في نظام حجز الخطوط الجوية.
2. صف الأحداث (Events Class): الحدث هو شيء ما لا يستغرق وقتاً بالنسبة لقياسنا الزمني، فالوصول (Arrival) مثلاً هو صف حدث في نظام حجز الخطوط الجوية.
3. صف التنظيم (Organization Class): التنظيم هو أي شكل من أشكال تجميع الأشياء لغاية ما، فوكالة السفر TravelAgency مثلاً هي صف في نظام حجز الخطوط الجوية.
4. صف الأشخاص (People Class): يجب فهم كلمة الشخص هنا على أنها الدور الذي يلعبه شخص ما في النظام، وليس

المنارة للاستشارات
www.manaraa.com

كشخص فيزيائي، فالمسافر Passenger هو صف في نظام حجز الخطوط الجوية.

5. صف الأمكنة (Places Class): يقصد بالأمكنة المواقع الفيزيائية ذات الصلة بنظام المعلومات، وعليه يكون مكتب السفر TravelOffice صفاً في نظام حجز الخطوط الجوية.

- هناك أيضاً طريقة مختلفة لتصنيف الصفوف:
  1. الصفوف الفيزيائية (مثلاً: الطائرة Airplane)
  2. صفوف العمل (مثلاً: الحجز Reservation)
  3. صفوف منطقية (مثلاً: جدول مواعيد الرحلات FlightTimetable)
  4. صفوف التطبيق (مثلاً: مناقلة الحجز ReservationTransaction)
  5. الصفوف الحاسوبية (مثلاً: الدليل، Index)
  6. صفوف السلوك (مثلاً: إلغاء الحجز، ReservationCancellation)

- يعطينا هذا المنهج توجيهات هامة ومفيدة في اكتشاف الصفوف لكنه لا يقدم إجرائية منظمة يمكن الاعتماد عليها لاكتشاف مجموعة كاملة وموثوقة من الصفوف، وقد يكون استخدام هذا المنهج مفيداً جداً لتحديد مجموعة الصفوف البدائية أو للتحقق من ضرورة أو عدم ضرورة وجود بعض الصفوف (التي اشتقت بوسائل أخرى). ويبقى ارتباط هذا المنهج بمتطلبات المستخدم ضعيفاً وقد لا يسمح بالوصول إلى حل مفهوم.

توصيف الحالة

1 – نمذجة الصفوف

(1) – مناهج اكتشاف الصفوف

المنهج المساق بحالات الاستخدام

- هو المنهج الذي ترتكز إليه لغة UML. يكمّل النموذج البياني لحالات الاستخدام بوصف سردي وبمخططات تسلسل أو مخططات تعاون لكل حالة من حالات الاستخدام.
  - تعرف هذه التوصيفات والمخططات الإضافية الخطوات (والأغراض) اللازمة لتحقق كل حالة استخدام، ويمكن أن نكتشف الصفوف انطلاقاً من هذه المعلومات.

- ينطوي هذا المنهج من الناحية العملية على بعض التشابه مع منهج الجمل الاسمية:
  - يستند هذا المنهج إلى حقيقة أن حالات الاستخدام تحدد المتطلبات، وعليه يعتمد كل من المنهجين على دراسة عبارات وثيقة المتطلبات لاكتشاف الصفوف المشاركة، وتبقى طريقة تمثيل هذه العبارات، سردية أم بيانية، مسألة ثانوية
  - في كل الأحوال، توصف معظم حالات الاستخدام في هذه المرحلة من دورة التطوير وصفاً نصياً بدون مخططات تفاعل
  - يعاني هذا المنهج عيوب مشابهة لتلك التي يعاني منها منهج الجمل الاسمية:
    - نظراً لكونه منهجاً تصاعدياً تعتمد دقته على صحة واكتمال نماذج حالات الاستخدام
    - قد يخل أيضاً بتوازن طريقة التطوير التكرارية التزايدية، إذ يتطلب نجاحه إكمال نماذج حالات الاستخدام قبل بناء نماذج الصفوف، مما قد يعيده في نهاية المطاف إلى منهج موجه بالوظائف

المنارة للاستشارات
www.manaraa.com

توصيف الحالة

1 – نمذجة الصفوف

(1) – مناهج اكتشاف الصفوف

المنهج CRC

- يتطلب تطبيق هذا المنهج إجراء جلسات تفكير تستخدم بطاقات مجهزة خصيصاً لهذه الغاية.
  - تضم البطاقة ثلاثة أجزاء:
    1. اسم الصف ويكتب في الجزء العلوي من البطاقة
    2. مسؤوليات الصف وتعدد في الجزء الأيسر من البطاقة
    3. المشاركون في الجزء الأيمن منها
  - المسؤوليات هي الخدمات (العمليات) التي يفترض أن ينجزها الصف لمصلحة صفوف أخرى
    - يتطلب إنجاز العديد من المسؤوليات تعاوناً (أو خدمات) من صفوف أخرى تدعى الصفوف المشاركة

- المنهج CRC هو إجرائية حركية، "يلعب" فيها المطورون بالبطاقات – فيكتبون فيها أسماء الصفوف ويربطون بها المسؤوليات والصفوف المشاركة أثناء تمثيل سيناريوهات المعالجة، (أي سيناريو لحالة استخدام).
  - كلما ظهرت الحاجة لخدمات جديدة لا توفرها الصفوف الموجودة يضاف صف جديد وتتسدل المسؤوليات وتحدد الصفوف المشاركة
  - إذا أصبح أحد الصفوف "محملاً" أكثر من اللازم يقسم إلى عدد من الصفوف الأصغر.

- خلافاً للطرق الأخرى يعتمد المنهج CRC في تحديد الصفوف على تحليل الرسائل المتبادلة بين الأغراض لإنجاز مهمات المعالجة. يجري التركيز في هذا المنهج على توزيع أعباء النظام توزيعاً متجانساً ومنتظماً وقد تشتق بعض الصفوف من حاجة تقنية كهذه بدلاً من اكتشافها كأغراض عمل،
  - بهذا المعنى قد يكون المنهج CRC مناسباً للتحقق من الصفوف التي تم اكتشافها بطرق أخرى
  - يعتبر هذا المنهج مفيداً لتحديد خصائص الصف (كما تمليها مسؤوليات الصف والصفوف المشاركة).

توصيف الحالة

1 – نمذجة الصفوف

(1) – مناهج اكتشاف الصفوف

المنهج المختلط

- تقاد إجرائية اكتشاف الصفوف عملياً بمناهج مختلفة في أوقات مختلفة، ويجري الاعتماد عادة على ملاحظات من المناهج الأربعة المذكورة سابقاً كلها، وتبقى معارف المحلل وخبرته وحدسه من العوامل الأساسية، والإجرائية ككل ليست تصاعدية وليست تنازلية، إنها شيء من الاثنين، وهذا ما ندعوه المنهج المختلط.

المنارة للاستشارات
www.manaraa.com

- مثال عن المنهج المختلط:
  - يمكن أن توضع مجموعة الصفوف الأولية بالاعتماد على معارف المحلل وخبرته
  - ويمكن أن يزودنا منهج عينات الصفوف المشتركة بتوجيهات إضافية
  - ويمكن أن تضاف لاحقاً صفوف أخرى بالاعتماد على منهج الجمل الاسمية بدراسة وتحليل توصيف حقل المسألة ذي المستوى الأعلى.
  - وإذا كانت مخططات حالات الاستخدام موجودة يمكن الاعتماد عليها لإضافة صفوف جديدة والتحقق من الصفوف الموجودة.
  - ويسمح المنهج CRC أخيراً بالتفكير بقائمة الصفوف المكتشفة سابقاً والتحقق منها.

توصيف الحالة

1 – نمذجة الصفوف

(2) – توجيهات عامة لاكتشاف الصفوف

- نعرض فيما يلي قائمة من التوجيهات التي قد تساعد المحلل في اكتشاف الصفوف ذات الصلة، لكن هذه القائمة ليست كاملة، ونذكر أننا نهتم هنا بصفوف الكيانات فقط:

1. يجب أن يكون لكل صف هدف واضح في النظام.
2. كل صف هو قالب يصف مجموعة من الأغراض، أما الصفوف التي لا يمكن أن نتصور وجود أكثر من غرض واحد منها فتنتمي على الأغلب إلى "أغراض العمل". تشكل هذه الصفوف عادة أساساً معرفياً للتطبيق ويجري ترميزها بصيغة ثابتة في البرامج التطبيقية. فإذا كان النظام مصمماً لمؤسسة واحدة مثلاً لن يعود وجود الصف Organization مبرراً.
3. يجب أن يستضيف كل صف (أي صف كيان) مجموعة من الصفات، ومن المستحسن أن نحدد الصفات المميزة (المفاتيح) لتساعدنا في معرفة تعداد الصف (أي العدد المتوقع وجوده في قاعدة المعطيات من أغراض هذا الصف). وليس من الضروري أن يتضمن الصف مفتاحاً يعرفه المستخدم، فمحددات الأغراض تميز أغراض الصفوف.
4. يجب أن يكون الصف متميزاً عن الصفة، ويكون المفهوم صفاً أو صفة تبعاً لحقل التطبيق. فالمفهوم "لون" السيارة مثلاً (Color) هو صفة للصف Car (سيارة) في مصنع للسيارات، أما في معمل للدهانات فيصبح المفهوم Color صفاً له صفاته الخاصة (درجة اللمعان، الإشباع، الشفافية، الخ).
5. يجب أن يستضيف كل صف مجموعة من العمليات، لكننا لا نهتم في هذه المرحلة بتحديد العمليات. وتستنتج عادة العمليات المكونة لواجهة الصف (أي الخدمات التي يوفرها للنظام) من بيان الهدف من الصف.

المنارة للاستشارات
www.manaraa.com

توصيف الحالة

1 – نمذجة الصفوف

(3) – أمثلة عن اكتشاف الصفوف

مثال (1) – تسجيل الطلاب في الجامعة (مثال محلول)

- **نص المسألة:**

تمنح جامعة متوسطة الحجم عدداً من الشهادات لطلاب من المرحلة الجامعية أو ما بعد الجامعية، يتبعون دروساً بدوام كامل أو بدوام جزئي. تتألف من أقسام تتضمن عدة فروع. ويشرف على كل شهادة قسم محدد لكن قد تعتمد الشهادة على دروس من أقسام أخرى. وتفتخر هذه الجامعة بالحرية التي تمنحها للطلاب في انتقاء موادهم الدراسية.

تفرض مرونة الانتقاء هذه أعباء قاسية على نظام التسجيل، إذ يجب ألا يتناقض برنامج الدراسة الذي ينتقيه الطالب مع القواعد الناظمة للحصول على الشهادة، كبنية المواد الأساسية التي تعتبر بالنسبة للشهادة. ويخضع انتقاء المواد لقيود معينة كالجدولة الزمنية للدروس والسعات العظمى للصفوف وغيرها.

تعتبر هذه المرونة أيضاً سبباً أساسياً في تزايد عدد طلاب الجامعة، وللحفاظ على تميزها فقد قررت هذه الجامعة استبدال نظام التسجيل الحالي– الذي مازال جزء منه يدوياً– بنظام برمجي جديد، لكنها لم تتمكن من الحصول على نظام برمجي جاهز يناسب نظامها، لذا فقد قررت تطوير نظامها الخاص.

يفترض أن يساعد النظام في الأنشطة السابقة للتسجيل وأن يحقق إجراءات التسجيل. وتتضمن الأنشطة السابقة للتسجيل إرسال علامات امتحانات الفصل الأخير للطلاب إلى جانب أية تعليقات متعلقة بالتسجيل، أما أثناء التسجيل فيجب أن يتقبل النظام برامج الدراسة التي يرغب بها الطلاب ويتحقق من صلاحيتها من حيث التزامها بالمواد الأساسية وعدم تعارضها مع الجدولة الزمنية للحصص وسعات الصفوف وغيرها. ويمكن حل بعض المشكلات التي تظهر باستشارة مدرسي الجامعة.

- **المطلوب:** حدد الصفوف ذات الصلة بعد أخذ المتطلبات التالية لنظام تسجيل الطلاب في الجامعة:

1. يحتاج الحصول على أي شهادة لعدد من المواد الإلزامية وعدد من المواد الخيارية
2. تقع كل مادة في مستوى معين ولها معدل علامات
3. يمكن أن تشكل المادة جزءاً من مواد أي عدد من الشهادات
4. يتطلب الحصول على أي شهادة بلوغ حد أدنى من النقاط (يتطلب فرع نظم المعلومات مثلاً الحصول على 68 نقطة بما فيها المواد الإلزامية)
5. يمكن للطالب أن يجمع عروض المواد في برامج دراسية تناسب رغباته وتؤدي لحصوله على الشهادة التي سجل بها

المنارة للاستشارات
www.manaraa.com

توصيف الحالة

1 – نمذجة الصفوف

(3) – أمثلة عن اكتشاف الصفوف

مثال (1) – تسجيل الطلاب في الجامعة (مثال محلول)

الحل

- يبين الجدول التالي (الشكل 1) مجموعة الصفوف ذات الصلة بهذه المسألة.

(الشكل 1):

الصف ذو الصلة	علاقات الاقتران
– المادة Course	مع الشهادة Degree
– الشهادة Degree	مع المادة Course
– الطالب Student	مع عرض المادة CourseOffering
– CourseOffering عرض المادة	مع الطالب Student

- يبين الجدول التالي (الشكل 2) مجموعة الصفوف الزائفة وفق معطيات المسألة.

(الشكل 2):

الصف الزائف	الشرح
CompulsoryCourse – مادة إلزامية	من الواضح أن المادة قد تكون إلزامية أو اختيارية تبعاً للشهادة، وقد نستطيع أن نميز بين المواد الإلزامية والمواد الاختيارية من خلال علاقة اقتران أو حتى بواسطة صفة للصف، ولذلك نعتبر هذين الصفين من الصفوف الزائفة.
– ElectiveCourse مادة اختيارية	
– StudyProgram البرنامج الدراسي	قد يظهر الصف StudyProgram كعلامة اقتران بين الصفين Student و CourseOffering، ولذلك سنصنف هذا الصف كصف زائف.

توصيف الحالة

1 – نمذجة الصفوف

(3) – أمثلة عن اكتشاف الصفوف

مثال (2) – مخزن أشرطة الفيديو (مثال غير محلول)

- **نص المسألة:**

ينوي أحد محلات الفيديو الجديدة تقديم خدمة إعارة أشرطة وأقراص الفيديو لقطاع واسع من الجمهور، وقد قرر أن يبدأ عمله

المنارة للاستشارات
www.manaraa.com

معتمداً على دعم نظام حاسوبي، ولذلك فقد استدعى محلل أعمال لتحديد احتياجاته وتوصيفها.
يحتفظ المخزن مبدئياً بحوالي 1000 شريط ممغنط (tape) و500 قرص (disk)، وقد طلب كل هذه الأفلام من مزود وحيد لكنه سيعتمد في الطلبات اللاحقة على عدة مزودين. وعلى كل شريط أو قرص رماز قضباني (bar Code) بحيث يعتمد النظام في دعم عمليات الإعارة والإعادة على آلة مسح. كما ترمز أيضاً معلومات الزبائن الأعضاء في رماز قضباني يظهر على بطاقات العضوية.
يمكن أن يحجز الزبون فيلماً بتاريخ معين، ويجب أن يعتمد النظام على محرك بحث مرن للإجابة على استفسارات الزبائن، بما فيها استفساراتهم عن الأفلام التي لا يملكها المخزن (لكن يمكن أن يطلبها بناء على طلبات الزبائن).

- **المطلوب:** حدد الصفوف المحتملة لنظام مخزن أشرطة الفيديو بعد أخذ المتطلبات التالية بالحسبان:
  - يحتفظ مخزن الفيديو في مستودعه بمكتبة واسعة من الأفلام الحالية والشائعة، ويمكن أن يكون فيلم ما مسجلاً على أشرطة فيديو أو على أقراص
  - تتواجد أشرطة الفيديو بشكلين 'Beta' أو 'VHS'، أما الأقراص فهي من الشكل DVD
  - لكل فيلم فترة استئجار محددة (بالأيام) مع أجرة محددة لتلك الفترة
  - يجب أن يستطيع المخزن الإجابة مباشرة على أي استفسار يتعلق بوجود فيلم ما وعدد الأشرطة و/أو الأقراص المتوفرة (يجب أن تكون حالة كل شريط أو قرص معروفة ومسجلة)

توصيف الحالة

1 – نمذجة الصفوف

(3) – أمثلة عن اكتشاف الصفوف

مثال (3) – إدارة العلاقات (مثال محلول)

- **نص المسألة:**

تعتمد إحدى شركات أبحاث التسويق على قاعدة من الزبائن هم في الواقع مؤسسات تشتري منها تقارير تحوي خلاصة تحليل الأسواق. ويشتري بعض كبار الزبائن برمجيات تخصصية من الشركة، فتزودهم الشركة بمعلومات خام غير مصاغة بحيث يتمكن أولئك الزبائن من استخدامها لتوليد تقارير خاصة بهم.
تسعى هذه الشركة دوماً لكسب زبائن جدد حتى لو لم يهتموا إلا بنمط واحد من التقارير. وحيث أن أولئك الزبائن هم زبائن محتملون ولم يصبحوا بعد زبائن فعليين، تفضل الشركة أن تسميهم "اتصالات" (Contacts).
تسعى الشركة إلى تطوير نظام إدارة علاقات جديد يمكن لكل موظفي الشركة استخدامه لكن بمستويات مختلفة من حقوق الوصول. يجب أن يسمح النظام بجدولة وإعادة جدولة الأنشطة بحيث يتمكن الموظفون من التعاون فيما بينهم لكسب زبائن جدد وتعزيز العلاقات القائمة حالياً.

- **المطلوب:** حدد الصفوف المحتملة لنظام إدارة العلاقات بعد أخذ المتطلبات التالية بالحسبان:
  - يدعم النظام وظيفة "البقاء على اتصال" مع زبائننا الحاليين والمتوقعين بحيث نتمكن من تلبية احتياجاتهم ومن كسب عقود جديد لشراء منتجاتنا

المنارة للاستشارات
www.manaraa.com

- يخزن النظام أسماء الشركات وأرقام هواتفها وعناوينها البريدية وعناوين المراسلة، بالإضافة إلى الأشخاص المسؤولين عن تنسيق الاتصال في تلك الشركات
- يسمح النظام لموظفينا بجدولة المهام والأحداث التي يجب إنجازها مع منسقي الاتصال. ويجدول الموظفون المهام والأحداث الخاصة بهم أو بغيرهم من الموظفين
- المهمة هي مجموعة أحداث يؤدي وقوعها إلى بلوغ نتيجة، وقد تكون النتيجة تحويل زبون متوقع إلى زبون فعلي أو تنظيم عملية تسليم منتج أو حل مشكلة أحد الزبائن. وأنماط الأحداث النمطية هي: مكالمة هاتفية، زيارة، إرسال فاكس، ترتيب عملية تدريب.. الخ.

توصيف الحالة

2 – توصيف الصفوف

(1) – تسمية الصفوف

- **اسم الصف:** يجب أن يعطى لكل صف اسماً يميزه، وفي بعض الأدوات المساندة قد يعطى أيضاً لكل صف رمزاً إلى جانب الاسم ويختلف عنه، ويجب أن ينسجم الرمز المعطى مع قواعد التسمية التي تتطلبها لغة البرمجة المستهدفة أو نظام قواعد المعطيات. ويستخدم الرمز، وليس الاسم، لتوليد الرماز البرمجي انطلاقاً من نماذج التصميم.

- **طريقة تسمية الصفوف:**
  1. يبدأ اسم الصف بحرف كبير
  2. عندما يكون الاسم مركباً من عدة كلمات نبدأ كل كلمة منها بحرف كبير
  3. يجب أن يكون اسم الصف اسماً مفرداً (مثلاً Course) أو صفة مع اسم مفرد (مثلاً: CompulsoryCourse).
  4. يجب أن يكون اسم الصف معبراً بحيث يعكس الطبيعة الحقيقية للصف، ويجب أن يؤخذ من مفردات المستخدمين (وليس من المفردات التي يستخدمها المطورون)

توصيف الحالة

2 – توصيف الصفوف

(2) – اكتشاف وتوصيف صفات الصف

- تتألف الأيقونة البيانية التي تمثل الصف من ثلاثة أجزاء (كما هو موضح في الشكل التالي):
  1. اسم الصف
  2. صفاته
  3. عملياته

- يجري اكتشاف الصفات على التوازي مع عملية اكتشاف الصفوف، فالتعرف على الصفات هو في الواقع من الآثار الجانبية لعملية تحديد الصفوف، لكن هذا لا يعني أن اكتشاف الصفات هو نشاط مباشر يجري وفق خط مستمر، بل على العكس إنها إجرائية تكرارية.

المنارة للاستشارات
www.manaraa.com

- نعرف في نماذج التوصيف الأولية فقط الصفات الأساسية اللازمة لفهم الحالات التي يمكن أن تتواجد فيها أغراض الصف، وقد نتجاهل مؤقتاً الصفات الأخرى (لكن يجب أن يضمن المحلل ألا يفقد هذه المعلومات في المستقبل). من النادر أن تذكر وثيقة المتطلبات كل صفات الصفوف لكن من المهم هنا ألا ندرج الصفات التي لا تشير إليها المتطلبات، ويمكن أن نضيف المزيد من الصفات في التكرارات التالية.

الشكل التالي (الشكل 3)

اسم الصف
واصفات الصف
عمليات الصف

توصيف الحالة

2- توصيف الصفوف

(3) – أمثلة لتوصيف الصفوف

مثال 4 – تسجيل طلاب في الجامعة (مثال محلول)

- **نص المسألة:** عد إلى المثال السابق (مثال 1) وخذ بعين الاعتبار المتطلبات الإضافية التالية من وثيقة المتطلبات:

1. يمكن أن يخضع اختيار الطالب من المواد للقيود التي يفرضها الجدول الزمني للحصص، وللقيود التي تحدد عدد الطلاب الذين يمكن تسجيلهم في عرض المادة الحالي
2. يُدخَل البرنامج الدراسي الذي يقترحه الطالب إلى نظام التسجيل آنياً. فيتحقق النظام من انسجام البرنامج ويشير إلى أية مشكلات في حال وجودها، والتي يجب حلها بمساعدة خبير أكاديمي. يجب أن يصدق مفوض رئيس القسم على البرنامج الدراسي النهائي الذي يوجه بعدئذ إلى المسجِّل

- **المطلوب:** تحديد واصفات كل صف من الصفوف التي تم تحديدها في المثال

توصيف الحالة

2- توصيف الصفوف

(3) – أمثلة لتوصيف الصفوف

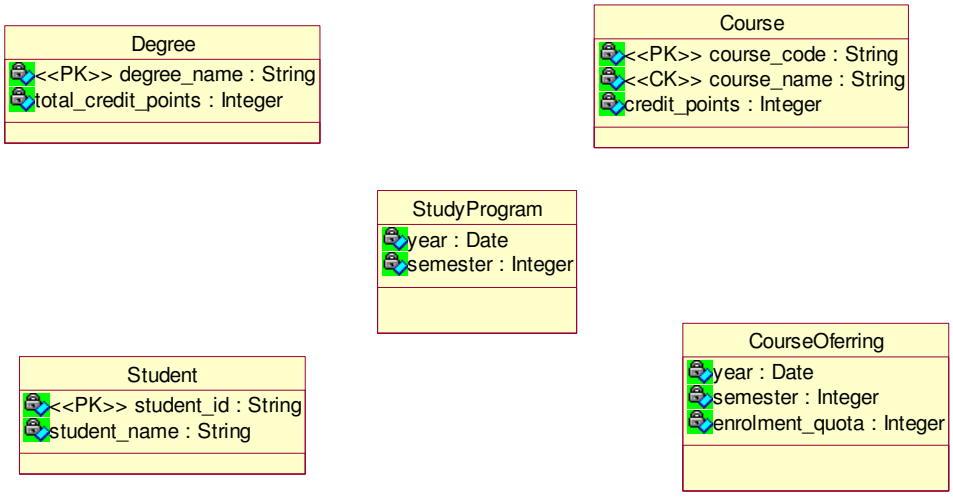
مثال 4 – تسجيل طلاب في الجامعة (مثال محلول)

الحل

- **الحل:** يبين الشكل التالي (الشكل 4) الصفوف التي تم تحديدها سابقاً (في المثال 1) مع واصفات كل منها

المنارة للاستشارات
www.manaraa.com

(الشكل 4):



توصيف الحالة

2- توصيف الصفوف

(3) – أمثلة لتوصيف الصفوف

مثال 5 – مخزن أشرطة الفيديو (مثال غير محلول)

- **نص المسألة:** بالعودة إلى المثال السابق (مثال 2) سنفترض أننا وضحنا المتطلب الثالث المتعلق بشروط التأخير من خلال المتطلبات الإضافية التالية:

1. تختلف أجرة الفيلم تبعاً لنوع الوسيط المخزن عليه: شريط أم قرص (لكن الأجرة هي نفسها بالنسبة لنوعي الأشرطة Beta و VHS)
2. مع أن الأقراص DVD هي الصيغة الوحيدة للأقراص الممغنطة حالياً في المخزن يريد المستخدمون أن يتسع النظام بسهولة في المستقبل ليستوعب صيغ أخرى للأقراص
3. ينحو موظفو المخزن إلى تذكر رموز معظم الأفلام الشائعة فيميزون الفيلم برمزه بدلاً من عنوانه، وهذه عادة مفيدة فقد يتواجد لعنوان الفيلم نفسه أكثر من إصدار لمخرجين مختلفين

- **المطلوب:** تحديد واصفات كل صف من الصفوف التي تم تحديدها في المثال

المنارة للاستشارات
www.manaraa.com

توصيف الحالة

2- توصيف الصفوف

(3) – أمثلة لتوصيف الصفوف

مثال 6 – إدارة العلاقات (مثال غير محلول)

- **نص المسألة:** عد إلى المثال السابق (مثال 3) وخذ بالحسبان المعلومات الإضافية التالية:

1. يعتبر الزبون حالياً في حال وجود عقد معه يقضي بتسليمه منتجاتنا أو خدماتنا. وتعتبر عملية إدارة التعاقد خارج نطاق نظامنا
2. يسمح النظام بتوليد تقارير متنوعة عن العقود بالاعتماد على العناوين البريدية وعناوين المراسلات. (مثلاً: البحث عن الزبائن حسب الرمز البريدي)
3. يسجل النظام تاريخ ووقت إنشاء المهمة، كما يمكن أن يخزن أيضاً "القيمة الحالية" المتوقعة من إنجاز المهمة
4. ستعرض الأحداث الخاصة بالموظف على شاشة في صفحات تشبه صفحات التقويم (يوم واحد في كل صفحة). يجب أن تتميز أولويات الأحداث (منخفضة، متوسطة، عالية) بصيغة بصرية مرئية على الشاشة
5. ليس من الضروري أن يقترن بكل حدث "وقت استحقاق"، فقد تكون بعض الأحداث غير موقوتة (أي يمكن إنجازها في أي وقت من اليوم المجدولة فيه)
6. لا يمكن تغيير وقت إنشاء الحدث لكن يمكن تغيير وقت استحقاقه
7. عند إنجاز الحدث يسجل النظام تاريخ ووقت الإنجاز
8. يسجل النظام أيضاً معلومات تحدد هوية الموظف الذي أنشأ المهمة أو الحدث، ومن هو المسؤول عن القيام به ("الموظف المسؤول") ومن أنجزه

- **المطلوب:** تحديد واصفات كل صف من الصفوف التي تم تحديدها في المثال.

توصيف الحالة

3- نمذجة علاقات الاقتران

(1) – اكتشاف علاقات الاقتران

- تربط علاقات الاقتران بين أغراض في النظام فتسهل التعاون والتشارك بينها، وبدونها لا يمكن أن تترابط الأغراض إلا في وقت التشغيل إذا كانت تشترك في الصفات نفسها أو تعرف قيم محددات هوية الأغراض الأخرى (عبر وسائل أخرى كالمتحولات مثلاً).
  - تعتبر علاقات الاقتران أهم نوع من العلاقات التي تظهر في النموذج، وخصوصاً في نموذج "أغراض العمل" الدائمة
  - تدعم علاقات الاقتران تنفيذ حالات الاستخدام وتربط بالتالي بين توصيف الحالة وتوصيف السلوك

- **اكتشاف علاقات الاقتران:** تظهر علاقات الاقتران على هامش عملية اكتشاف الصفوف:
  - عند تعريف الصفوف يتخذ المحلل قراراً بصدد صفات الصف التي يجسد بعضها علاقات اقتران بصفوف أخرى، فقد تكون أنماط معطيات الصفات أنماطاً أولية وقد تكون منمطة بصفوف أخرى فتجسد بذلك العلاقات مع تلك الصفوف

المنارة للاستشارات
www.manaraa.com

- يجب، من حيث المبدأ، أن تُنمذَج أية صفة نمط معطياتها غير أولي كعلاقة اقتران (أو تجميع) بصف يمثل نمط المعطيات الذي تنتمي إليه

- لا يحتاج التعبير الكامل عن دلالات النموذج لإغلاق حلقة الاقترانات، إذ يوجد علاقة واحدة على الأقل يمكن اشتقاقها من العلاقات الأخرى، بل يجب بالأحرى حذف هذا النوع من الاقتران تجنباً لتكرار تمثيل المفهوم نفسه. ومن الممكن، بل من المرجح أيضاً، أن تظهر العديد من علاقات الاقتران المشتقة على نموذج التصميم (لأسباب تتعلق بالفاعلية).

توصيف الحالة

3- نمذجة علاقات الاقتران

(2) - توصيف علاقات الاقتران

- يتطلب توصيف علاقة الاقتران:
  1. تسميتها
  2. تسمية أدوارها
  3. تحديد تعدادها

- تسمية علاقات الاقتران:
  - تخضع تسمية علاقات الاقتران لقواعد واصطلاحات تسمية الصفات – أي بأحرف صغيرة، واستخدام الشرطة التحتية للفصل بين الكلمات
  - عندما تربط بين صفين علاقة اقتران واحدة فقط يمكن إغفال اسم العلاقة وأسماء أدورها. وتميز الأداة المساندة في هندسة البرمجيات بين علاقات الاقتران بواسطة أسماء يعطيها النظام لهذه العلاقات

- تسمية أدوار علاقات الاقتران:
  - يمكن أن تستخدم أسماء الأدوار لتوضيح علاقات الاقتران الأكثر تعقيداً، وعلى وجه الخصوص علاقات الاقتران الذاتي (أو الاقتران العددي الذي يربط بين أغراض من الصف نفسه)
  - وفي كل الأحوال يجب عند تسمية الأدوار الأخذ بالحسبان أن هذه الأسماء ستتحول في نماذج التصميم إلى صفات للصفوف على الطرف المقابل من علاقة الاقتران

- يجب تحديد تعداد علاقة الاقتران على كل من طرفيها (أي أدوارها)، وإذا لم يكن تعداد الاقتران واضحاً في هذه المرحلة يمكن تجاهله وتأجيل تحديده إلى مرحلة لاحقة.

المنارة للاستشارات
www.manaraa.com

4- نمذجة علاقات التركيب والتجميع

- **عمليتا التركيب والتجميع:** تجسد علاقة التجميع والصيغة الأقوى منها، وهي علاقة التركيب، مفهوم "الكل- الجزء" بين صف مركب (أعلى) وصف مكوِّن (أدنى).
  - تُعامل علاقة التجميع في UML كصيغة مقيدة من علاقة الاقتران، ويقلل هذا الأمر في الواقع من أهمية ما تعنيه علاقة التجميع في النمذجة. ويكفي القول إن علاقة التجميع هي، إلى جانب علاقة التعميم، أهم تقنية لإعادة استخدام الوظائف في الأنظمة غرضية التوجه.

- **دلالات عملية التجميع:** لقد كان من الممكن أن تزداد مقدرة UML على النمذجة إلى حد بعيد لو أنها دعمت تمثيل الدلالات الأربع الممكنة لعلاقة التجميع:

1. **تجميع "الملكية الحصرية":**
   - يكون وجود الصفوف المكونة مرهوناً بوجود الصف المركب (ولذلك يؤدي حذف الغرض المركب إلى حذف كل الأغراض المكونة له)
   - علاقة التجميع هي علاقة متعدية
   - علاقة التجميع غير متناظرة
   - علاقة التجميع ثابتة

2. **تجميع "الملكية":** تدعم علاقة تجميع الملكية الخصائص الثلاث الأولى لعلاقة تجميع الملكية الحصرية. أي:
   - ارتباط الوجود
   - التعدي
   - عدم التناظر

3. **تجميع "الحيازة":** لعلاقة تجميع الحيازة معنى أضعف من علاقة تجميع الملكية فهي:
   - علاقة متعدية
   - علاقة غير متناظرة

4. . **تجميع "العضوية":** تشير إلى تجميع أغراض مستقلة لهدف معين – أي هي علاقة تجميع لا تفترض شيئاً بالنسبة لخصائص ارتباط الوجود أو التعدي أو عدم التناظر أو الثبات. إنها عملية تجريد تعتبر أن مجموعة الأعضاء المكونة تشكل غرضاً مركباً من مستوى أعلى، ويمكن حسب هذه العلاقة أن ينتمي الغرض المكوِّن إلى أكثر من غرض مركب (وعليه فقد يكون تعداد علاقة الاقتران هذه من الشكل "عدة لعدة").

المنارة للاستشارات
www.manaraa.com

توصيف الحالة

4- نمذجة علاقات التركيب والتجميع

(1) - اكتشاف علاقات التجميع والتركيب

- تُكتشَف علاقات التجميع على التوازي مع عملية اكتشاف علاقات الاقتران:
  - عندما تبدي علاقة الاقتران خاصية أو أكثر من الخصائص الدلالية الأربع لعلاقة التجميع يمكن نمذجتها كعلاقة تجميع

- يكمن الاختبار الأساسي في استخدام العبارات "له" و"هو جزء من" في شرح معاني العلاقات
  - عند الشرح التنازلي تستخدم العبارة "له" (مثلاً: للكتاب فصل أو Book 'has' Chapter)
  - في التفسير التصاعدي تستخدم العبارة "هو جزء من" (مثلاً: الفصل هو جزء من كتاب أو Chapter 'is_part_of' Book) فإذا لم يضيف وجود هذه العبارات معنى إضافياً لا تكون العلاقة علاقة تجميع

- تربط علاقة التجميع، من وجهة نظر بنيوية، بين عدد كبير نسبياً من الصفوف حيث لا تعطي علاقة الاقتران بعد الدرجة الثانية معنى إضافياً

- قد تكون علاقة تجميع العضوية هي الحل الأفضل عندما نحتاج لربط أكثر من صفين معاً

توصيف الحالة

4- نمذجة علاقات التركيب والتجميع

(2) - توصيف علاقات التجميع والتركيب

- تدعم لغة UML التجميع دعماً محدوداً، وتوفر صيغة أقوى هي علاقة التركيب التي يمكن أن يحوي فيها الغرض المركب الأغراض الأجزاء احتواء فيزيائياً (أي بمعنى الاحتواء بالقيمة by value)، ولا يمكن أن ينتمي الغرض الجزء لأكثر من غرض مركب واحد. وعليه تقابل علاقة التركيب في UML إلى حد ما علاقات تجميع "الملكية" وتجميع "الملكية الحصرية".

- أما الصيغة الأضعف من علاقة التجميع في UML فهي المدعوة ببساطة علاقة التجميع، ولها دلالة الاحتواء "بالمرجع" (by reference)، فالغرض المركب لا يحوي الأغراض الأجزاء فيزيائياً، ويمكن أن يكون للغرض الجزء نفسه عدة علاقات تجميع أو اقتران في النموذج، وتقابل هذه العلاقة علاقات تجميع "الحيازة" و"العضوية".

- **التمثيل في لغة UML:**
  - تمثل علاقة التركيب في UML بمعين مطموس
  - تمثل علاقة التجميع بمعين فارغ

يماثل ما تبقى من عناصر توصيف علاقة التجميع عناصر توصيف علاقة الاقتران

المنارة للاستشارات
www.manaraa.com

توصيف الحالة

4- نمذجة علاقات التركيب والتجميع

(3) – أمثلة لتوصيف علاقات التجميع والتركيب

مثال 7 – تسجيل الطلاب في الجامعة (مثال محلول)

- **نص المسألة:** عد إلى المثالين (مثال 1) و(مثال4) وخذ بالحسبان المتطلبات الإضافية التالية:

1. يجب أن يكون السجل الأكاديمي للطالب متوفراً عند الطلب، يجب أن يتضمن هذا السجل العلاقات التي حصل عليها الطالب في المواد التي سجل بها (ولم يفصل منها كعقوبة، أي في الأسابيع الثلاثة الأولى من بداية الفصل)
2. لكل مادة شخص أكاديمي واحد مسؤول عنها، لكن يمكن أن يشارك في تدريسها أكثر من أكاديمي. وقد يختلف هؤلاء الأشخاص بالنسبة للمادة نفسها من فصل لآخر

- **المطلوب:** توصيف علاقات التجميع والتركيب في هذا المثال.

توصيف الحالة

4- نمذجة علاقات التركيب والتجميع

(3) – أمثلة لتوصيف علاقات التجميع والتركيب

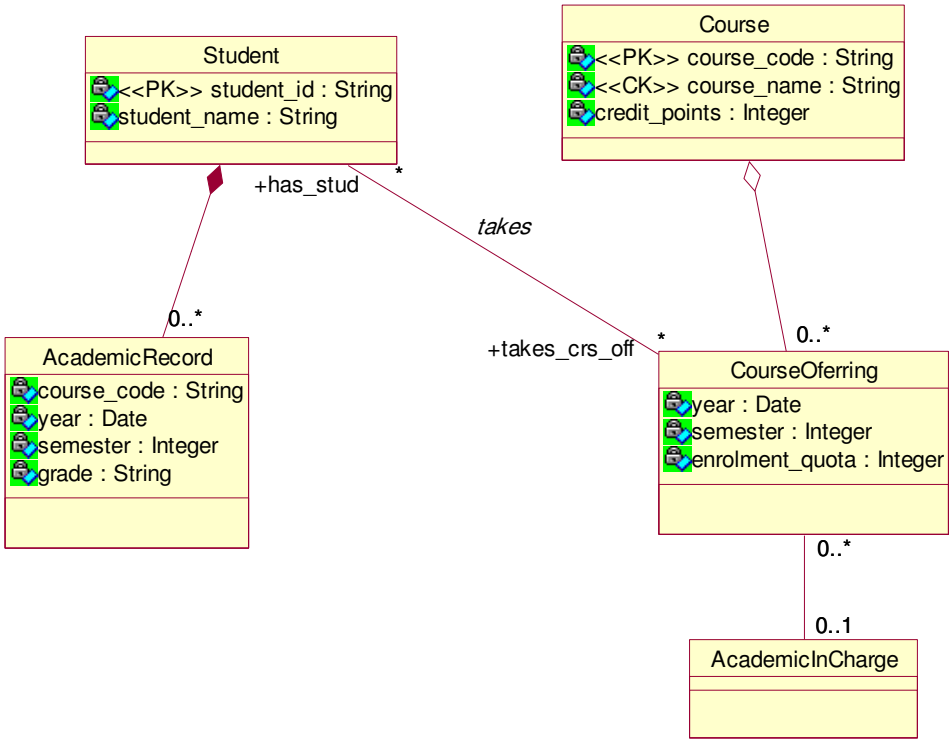
مثال 7 – تسجيل الطلاب في الجامعة (مثال محلول)

الحل

الحل: يبين الشكل التالي (الشكل 5) نموذج الصفوف الذي يركز على علاقات التجميع

المنارة للاستشارات
www.manaraa.com

(الشكل 5):



توصيف الحالة

4- نمذجة علاقات التعميم

- **التعميم:** هو عملية تجريد الملامح المشتركة (من طبقات وعمليات) بين عدد من الصفوف ووضعها في صف أكثر عمومية.

- **علاقة التعميم:** تربط علاقة التعميم بين صف عام (صف أعلى) وبين عدد من الصفوف المحددة (صفوف أدنى)
  - يسمح التعميم للصفوف الأدنى بوراثة (إعادة استخدام) ملامح الصف الأعلى، تطبق الوراثة في النظام غرضي التوجه التقليدي على الصفوف وليس على الأغراض (تورث الأنماط وليس القيم).

- **أهداف عملية التعميم:** لعملية التعميم هدفان، إلى جانب الوراثة:

1. **قابلية التبديل:** ويُقصد بمبدأ قابلية التبديل أن غرضاً من الصف الأدنى يمكن أن يكون قيمة مقبولة لمتحول من نمط الصف الأعلى، فإذا صرحنا مثلاً عن متحول ليخزن أغراضاً من الصف Fruits، يكون الغرض Apple قيمة مقبولة لهذا المتحول.
2. **تعدد الأشكال:** وفقاً لمبدأ تعدد الأشكال يمكن تحقيق العملية نفسها بطرق مختلفة في صفوف مختلفة، فيستطيع بذلك غرض ما أن يستدعي العملية دون أن يعرف أي تحقيق من بين تحقيقاتها المختلفة سينفذ، إذ يعرف الغرض المستدعى إلى أي صف ينتمي وينفذ تحقيق العملية الخاص بهذا الصف.
   - تظهر فائدة مبدأ تعدد الأشكال جلية عند استخدامه إلى جانب مبدأ الوراثة، فمن الشائع أن نصرح عن عملية متعددة

المنارة للاستشارات
www.manaraa.com

الأشكال في الصف الأعلى دون تحقيقها، أي يعطي توقيع العملية (اسمها وقائمة الوسطاء الصورية)، على أن يجري تحقيقها في كل صف من الصفوف الأدنى، وبذلك تصبح العملية مجردة.

- يجب عدم الخلط هنا بين العملية المجردة والصف المجرد، فالصف المجرد هو صف ليس له أية أغراض أمثال منه (لكن يمكن أن يكون لصفوفه الأدنى أغراض أمثال) على سبيل المثال قد لا نجد أي مثل من الصف Vegetable (خضار)، والأمثال الوحيدة الموجودة هي أغراض من الصفوف Potato، Carrot، وغيرها. وفي الواقع، يكون الصف الذي يحوي عملية مجردة، صفاً مجرداً، ولا يمكن أن يحوي الصف الملموس، مثل الصف Apple، عمليات مجردة. ينعكس وجود العمليات المجردة على توصيف السلوك بينما تظهر الصفوف المجردة في توصيف الحالة.

توصيف الحالة

4- نمذجة علاقات التعميم

(1) – اكتشاف وتوصيف علاقات التعميم

- **اكتشاف علاقات التعميم:** قد يلاحظ المحلل وجود العديد من الصفوف الأعلى/ الصفوف الأدنى عند إنشاء قائمة الصفوف الأولية، بينما تظهر العديد من علاقات التعميم الأخرى أثناء تعريف علاقات الاقتران. فقد تحتاج علاقات الاقتران المختلفة (حتى لو كانت من الصف نفسه) للاتصال بصف في مستوى مختلف من التعميم/ التخصيص. يمكن على سبيل المثال أن يقترن الصف Course بالصف Student (حسب علاقة الاقتران **طالب يأخذ مادة** Student Takes Course).

- تظهر علاقات التعميم عند استخدام عبارات مثل "يمكن أن يكون" و"هو نوع من" في شرح العلاقات:
  - تستخدم العبارة "يمكن أن يكون" عند الشرح التنازلي (يمكن أن يكون الطالب مدرساً مساعداً Student Can be TeachingAssistant)
  - تستخدم العبارة "هو نوع من" في التفسير التصاعدي (مثلاً: المدرس المساعد هو نوع من الطلاب: TeachingAssistant 'is- a- kind- of' Student). لاحظ أنه قد يكون المدرس المساعد TeachingAssistant نوع من المدرسين Teacher، ليظهر عندئذ مفهوم الوراثة المتعددة

- **توصيف علاقات التعميم:** تُظهر علاقة التعميم بين الصفوف أن صفاً ما يتمتع بالبنية والسلوك المعرفين في صف أو أكثر من الصفوف الأخرى
  - تمثل علاقة التعميم في UML بخط مستقيم يصل بين الصفين مع سهم يشير إلى الصف الأعلى

نمذجة الصفوف المتقدمة

1 – الأنماط المحيطية – القيود

- **النمط المحيطي:** يحدد النمط المحيطي عنصر نمذجة موجود في UML لكنه لا يشكل عنصراً جديداً بحد ذاته، كما أنه لا يغير بنية UML بل يغني فقط معنى تدوين موجود، ويسمح بتحديد وتخصيص الطريقة، ويمكن استخدامه بأساليب مختلفة.

المنارة للاستشارات
www.manaraa.com

- تكتب الأنماط المحيطية عادة في النموذج كاسم تحيط به قوسان صغيرتان مثل: <<include>>, <<PK>> <<global>> كما يمكن تمثيلها بأيقونات أيضاً.

- **الأنماط المحيطية في لغة UML:** تتضمن لغة UML تعريفاً مسبقاً مضمناً فيها لتمثيل بعض الأنماط المحيطية الشائعة، وغالباً ما نجد أيقونات تمثل هذه الأنماط في الأدوات المساندة في هندسة البرمجيات والتي يسمح معظمها بإنشاء أيقونات جديدة حسب رغبة المحلل.

- **اللاحة (Profile):** يطلق على مجموعة الأنماط المحيطية المعرفة بهدف موضوع من مواضيع نمذجة التصميم تسمية اللاحة (Profile)، وقد تتضمن مقاييس UML في المستقبل لاحات لأهداف تصميمية شائعة كقواعد المعطيات مثلاً.

- يجري غالباً الخلط بين مفهومي الأنماط المحيطية والقيود إذ لا يكون الفارق بينهما في بعض الأحيان واضحاً، يستخدم النمط المحيطي عادة لوضع قيد جديد على النموذج–وهو أمر ذو معنى بالنسبة للمنمذج لكن UML لا تدعمه كقيد مباشرة.

- يمكن أن يقترن أي عنصر من عناصر النمذجة بقيد ما أو أن يعطى نمطاً محيطياً:
  - لا تُعرض على مخطط النموذج إلا القيود البسيطة، فيظهر القيد كعبارة نصية بين قوسين كبيرتين، ويربط بيانياً ومعنوياً بعنصر النمذجة المعني
  - أما القيود الأكثر تعقيداً (والتي تكون طويلة بحيث يتعذر إظهارها على النموذج البياني) فتخزن في مخزن معطيات الأداة المساندة المستخدمة وبصيغة نصية عادية.

نمذجة الصفوف المتقدمة

2 – الملاحظات والأمارات

- يمكن استخدام مفهوم الملاحظة في UML عندما:
  - يكون نص القيد طويلاً
  - أو عندما يتعلق القيد بثلاثة رموز بيانية أو أكثر

- تمثل الملاحظة بيانياً بمستطيل زاويته العليا اليمنى مطوية

- يمكن استخدام رمز الملاحظة لاحتواء نص يعبر عن قيد. لكن يمكن عموماً أن يحوي هذا الرمز أية معلومات أخرى
  - للتأكيد على أن الملاحظة تمثل قيداً يفضل وصفها بالنمط المحيطي <<Constraint>>

- تستخدم الأمارات، مثل الملاحظات، لتمثيل أية معلومات أخرى بصيغة نصية في النموذج، وقد تمثل الأمارة قيداً:
  - تكتب الأمارات مثل القيود بين قوسين كبيرتين وتأخذ الصيغة التالية: Tag=value
  - مثلاً: {analyst=Les, Status=2nd iteration}

المنارة للاستشارات
www.manaraa.com

نمذجة الصفوف المتقدمة
3- نطاق الرؤية والتغليف
(1) – رؤية الخصائص المحمية

- يتم تدوين نطاقات الرؤية في لغة UML باستخدام إشارات توضع قبل أسماء الصفات.

- **رؤية الخصائص المحمية:** ليس من المناسب دوماً أن نحصر حق الوصول إلى صفات وعمليات الصف الأساسي الخاصة بأغراض الصف نفسه فقط، إذ يفضل في العديد من الحالات أن يسمح لأغراض صف مشتق (أي صف فرعي من الصف الأساسي) بالوصول إلى خصائص الصف الأساسي الخاصة. لنأخذ على سبيل المثال هرمية الصفوف التي يظهر فيها الصف Person كصف أساسي والصف Employee كصف مشتق، فإذا كان Joe غرضاً من الصف Employee يجب وفقاً لتعريف علاقة التعميم أن يكون له حق الوصول إلى خصائص الصف Person أو بعضها على الأقل (مثلاً إلى الصفة date-of-birth).
  - لكي نسمح لصف مشتق بالوصول إلى خصائص صف الأساس (الخاصة طبعاً) يجب أن نعرف هذه الخصائص في الصف الأساسي كخصائص محمية (Protected)
  - نجد هذا السيناريو في معظم بيئات البرمجية غرضية التوجه الشائعة مثل C++ وغيرها، ولا تتميز لغة UML بالنسبة لهذا الموضوع عن غيرها فهي تؤكد: "إن إمكانية الرؤية هي جزء من العلاقة بين العنصر وحاويته، التي قد تكون حزمة أو صفاً أو أي فضاء تسمية آخر."

نمذجة الصفوف المتقدمة
3- نطاق الرؤية والتغليف
(2) – رؤية خصائص الصف الموروثة

- تطبق قواعد الرؤية في UML على أغراض بمستويات مختلفة، فهي تطبق كما نعلم على أغراض أولية- الصفات والعمليات. لكن يمكن أيضاً تحديد نطاقات الرؤية بأخذ حاويات أخرى بالحسبان، ويؤدي هذا إلى ظهور قواعد إبطال متشابكة فيما بينها.

- تعرف نطاقات الرؤية في هرمية الوراثة على مستويين هما:
  - مستوى الصف الأساس
  - مستوى خصائص هذا الصف

- لنقل مثلاً إن B صف مشتق من الصف A الذي يحوي خليطاً من الصفات والعمليات- بعضها عمومي وبعضها خاص وبعضها أيضاً محمي، ولنطرح السؤال التالي: ما هي نطاقات رؤية الخصائص الموروثة في الصف B؟ تتعلق الإجابة على هذا السؤال بمستوى الرؤية المعطى للصف الأساس A عند التصريح عنه في الصف B. إذ يمكن أن يعرف الصف الأساس أثناء الاشتقاق:
  - كصف عمومي
  - أو كصف محمي
  - أو كصف خاص

- تخضع نطاقات الرؤية في السيناريو السابق للقواعد التالية:

المنارة للاستشارات
www.manaraa.com

- تكون خصائص (صفات وعمليات) الصف الأساس A الخاصة غير مرئية لأغراض الصف B بغض النظر عن كيفية تعريف الصف الأساس A في الصف المشتق B
- إذا عُرف الصف الأساس A كصف عمومي لا تتغير نطاقات رؤية الخصائص الموروثة في الصف B (العمومية تبقى عمومية والمحمية تبقى محمية)
- إذا عُرف الصف الأساس A كصف محمي تصبح الخصائص العمومية الموروثة محمية في الصف المشتق B.
- إذا عُرف الصف الأساس A كصف خاص تصبح الخصائص العمومية والمحمية الموروثة من A خاصة في الصف المشتق B

نمذجة الصفوف المتقدمة

3- نطاق الرؤية والتغليف

(2) – رؤية الخصائص الصديقة

- **الخصائص الصديقة:** قد نصادف حالات يحتاج فيها تابع ما (أو حتى صف بأكمله) للوصول إلى خصائص صف آخر بغض النظر عن مستويات رؤية هذه الخصائص، وتظهر هذه الحالات عند وجود صفين توأمين تعتمد عمليات أحدهما على خصائص الآخر.

- نذكر كمثال نمطي على هذه الحالة الصفتين Book و Book Shelf مع العملية Puton Book Shelf في الصف Book. يمكن مواجهة هذه الحالة بالتصريح عن العملية Puton Book Shelf كصديق (Friend) في الصف Book Shelf.

- قد يكون الصديق صفاً آخر أو عملية من صف آخر:
  - لكن علاقة الصداقة ليست متبادلة، فإذا كان صف ما صديقاً لصف آخر ليس من الضروري أن يكون هذا الأخير صديقاً للأول

- يُصرح عن الصديق (عملية أو صف) ضمن الصف الذي يمنح علاقة الصداقة، لكن لا تعتبر العملية الصديقة من خصائص هذا الصف وبالتالي لا تطبق عليها قواعد نطاقات الرؤية.
  - وينتج عن هذا أيضاً أننا لا نستطيع ضمن تعريف الصديق أن نشير إلى صفات الصف بذكر أسماء هذه الصفات فقط، بل يجب إرفاقها باسم الصف (كما لو كان الصديق عملية خارجية عادية)

- تمثل علاقة الصداقة في UML كعلاقة ارتباط تظهر بصيغة خط متقطع من الصف الصديق أو العملية الصديقة إلى الصف الذي يمنح الصداقة، ويُلصق بسهم الارتباط النمط المحيطي <<Friend>>. من الواضح أن تدوين UML لا يدعم دلالة هذه العلاقة بشكل كامل.

نمذجة الصفوف المتقدمة

4- صف الاقتران والصف المصنع

- يُستخدم صف الاقتران عادة عند وجود علاقة اقتران تعدادها عدة لعدة بين صفين، ولكل مثل من الاقتران (لكل رابطة) قيم صفاته

المنارة للاستشارات
www.manaraa.com

الخاصة به، فلكي نتمكن من تخزين قيم هذه الصفات نحتاج لصف هو صف الاقتران.

- يبدو مفهوم صف الاقتران بسيطاً لكنه ينطوي في الواقع على قيد مخفي، لنأخذ مثلاً صف الاقتران C بين الصفين A و B. يظهر هنا القيد المذكور إذ لا يمكن أن يوجد أكثر من مثل واحد من C لكل زوج من الأمثال المترابطة من A و B
- إذا لم يكن هذا القيد مقبولاً يجب على المنمذج أن يصنّع الاقتران باستبدال الصف C بصف عادي، D مثلاً. يقترن الصف المصنَّع D اقتراناً ثنائياً مع كل من A و B وهو مستقل عنهما. فلكل مثل من الصف D هويته الخاصة بحيث يمكن أن ننشئ منه عدة أمثال ترتبط بالأمثال نفسها من A و B

- لا يمكن أن تتكرر في صف الاقتران أغراض تحوي مراجع متماثلة للصفوف المقترنة، أما الصف المصنع فهو مستقل عن الصفوف المقترنة ولا يضع قيداً كهذا على أغراضه، إذ لا يستخدم المفتاح الأساسي في الصف المصنع صفات تشير إلى الصفوف المقترنة.

نمذجة الصفوف المتقدمة

4- صف الاقتران والصف المصنع

(1) – مثال (8) – قاعدة معطيات الموظفين

نموذج يستخدم صف الاقتران

- **نص المسألة:**

لكل عامل في المؤسسة رقم مميز وحيد هو emp-id، ويُحفظ اسم الموظف المؤلف من الاسم (first name) والنسبة (last name)، والحرف الأول من الاسم الأوسط (middle initial).

يعين كل موظف في مستوى راتب محدد، ولكل مستوى راتب مجال من القيم، أي راتب أدنى وراتب أعلى، ولا يغير هذا المجال أبداً، فإذا ظهرت الحاجة لتعديل راتب الحد الأدنى أو راتب الحد الأعلى يتم تعريف مستوى راتب جديد ويحفظ تاريخ بداية وتاريخ نهاية المستوى.

يجب الاحتفاظ أيضاً برواتب الموظف السابقة بما فيها تاريخي البداية والنهاية في كل مستوى، كما تسجل أيضاً أية تعديلات تجري على راتب الموظف ضمن المستوى نفسه.

- **نموذج يستخدم صف اقتران**
- **المطلوب:** عد إلى نص مسألة قاعدة معطيات الموظفين المذكور أعلاه وارسم نموذج الصفوف المناسب مستخدماً صف اقتران.

يشير نص المسألة إلى عدد من النقاط الهامة، إذ نعرف أننا نحتاج إلى صف لتخزين تفاصيل معلومات الموظفين (Employee) وإلى صف آخر لتخزين المعلومات المتعلقة بمستويات الرواتب (Salary Level)، وتظهر الصعوبة الأساسية في نمذجة الارتباط بين الموظفين ورواتبهم الحالية والسابقة، وقد يبدو للوهلة الأولى أنه من الطبيعي أن نستخدم صف الاقتران Salary History Association.

المنارة للاستشارات
www.manaraa.com

نمذجة الصفوف المتقدمة

4- صف الاقتران والصف المصنع

(1) – مثال (8) – قاعدة معطيات الموظفين

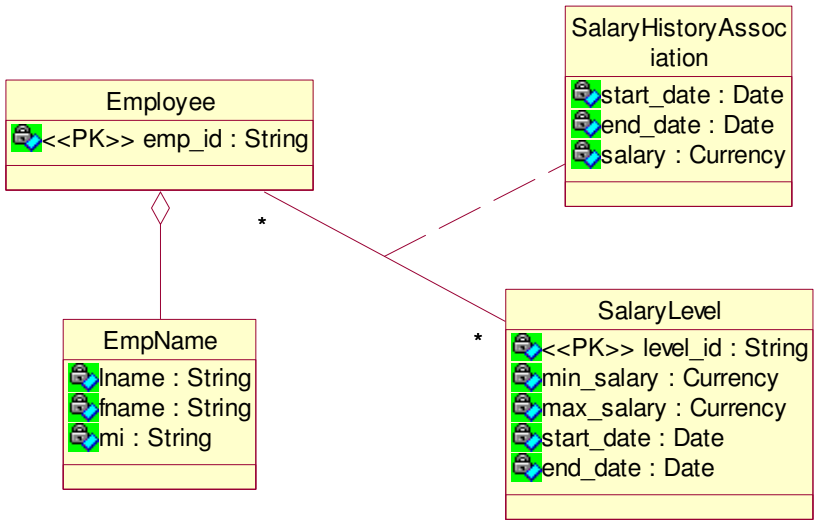
نموذج يستخدم صف الاقتران (الحل)

- **الحل:**

يبين الشكل التالي (الشكل 7) نموذج صفوف يتضمن صف الاقتران Salary History Association، لكن يبدو أن هذا الحل ضعيف وناقص، إذ تشتق هوية غرض الصف Salary History Association من المفتاح المركب المكون من مراجع لمفاتيح رئيسية في الصفين Employee و salary Level (أي emp-id و Level-id).

لا يمكن أن يكون لغرضين من الصف Salary History Association المفتاح المركب نفسه (أي نفس الارتباطات مع Employee و Salary Level)، وهذا يعني أن تصميم الشكل السابق لا يتناسب مع المتطلب الذي ينص على ضرورة تسجيل أية تعديلات تجري على راتب الموظف ضمن المستوى نفسه. ولذلك نحتاج إلى نموذج أفضل.

الشكل 7



www.manaraa.com

نمذجة الصفوف المتقدمة

4- صف الاقتران والصف المصنع

(1) – مثال (8) – قاعدة معطيات الموظفين

نموذج يستخدم صفاً مصنعاً

- **نموذج يستخدم صفاً مصنعاً**
- **المطلوب:** عد إلى نص مسألة قاعدة معطيات الموظفين وارسم نموذج الصفوف المناسب مستخدماً صفاً مصنعاً.

نمذجة الصفوف المتقدمة

4- صف الاقتران والصف المصنع

(1) – مثال (8) – قاعدة معطيات الموظفين

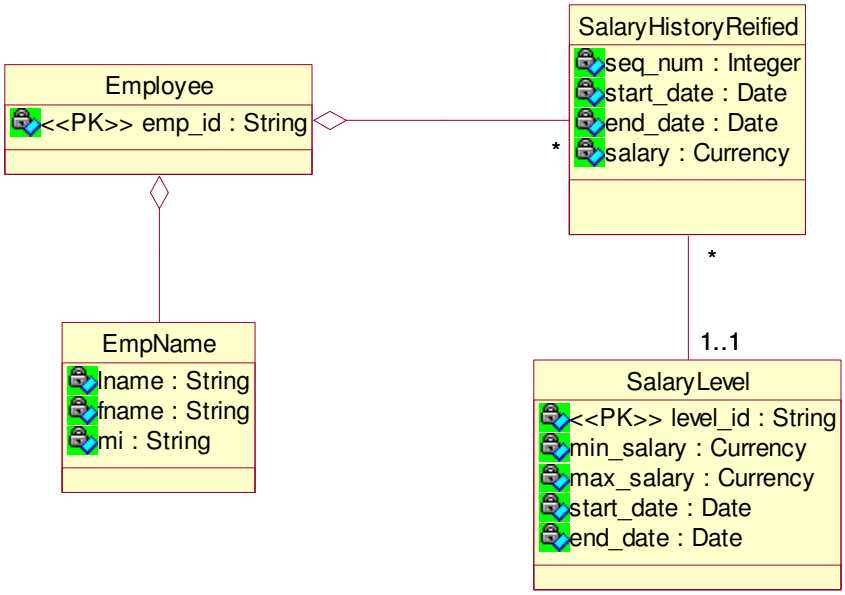
نموذج يستخدم صفاً مصنعاً (الحل)

- **الحل:**

يبين الشكل التالي (الشكل 8) نموذج الصفوف الذي يستخدم الصف المصنع Salary History Reified، ولا يشير نموذج هذا الصف صراحة إلى مفتاحه الأساسي، لكن يمكننا أن نتوقع أن هذا المفتاح يتألف من emp-id و seq-num، وتخزن الصفة seq-num الرقم التسلسلي لتغيرات راتب موظف. ينتمي أي غرض من Salary History Reified إلى غرض Employee وحيد ويرتبط بغرض واحد من Salary Level. وبذلك يستطيع هذا النموذج الجديد أن يعكس تغيرات راتب الموظف ضمن المستوى نفسه، لكن مازال هذا النموذج بحاجة لتحسينات إضافية بالتأكيد.

(الشكل 8):

المنارة للاستشارات
www.manaraa.com



طبقات الصفوف

1 – تعقيد الشبكات

التعقيد: للتعقيد عدة أنواع وأشكال، وقد يكون عدد الوصلات بين الصفوف قياساً بسيطاً لكن في الوقت نفسه معبر جداً.

- نعرف الوصلة بأنها وجود ارتباط دائم أو مؤقت بين صفين
- تسمح كل وصلة عادة بتخاطب ثنائي الاتجاه بين الصفوف، أي من A إلى B ومن B إلى A
- إن عدد الوصلات الممكنة بين m صف هو $m(m-1)/2$، وبتطبيق هذه القاعدة على الصفوف السبعة في الشكل السابق نجد أن عدد الوصلات الممكنة هو 21 (و42 مسار تخاطب)

لاحظ أننا نقيس التعقيد بالنسبة إلى عدد الصفوف وليس لعدد الأغراض:

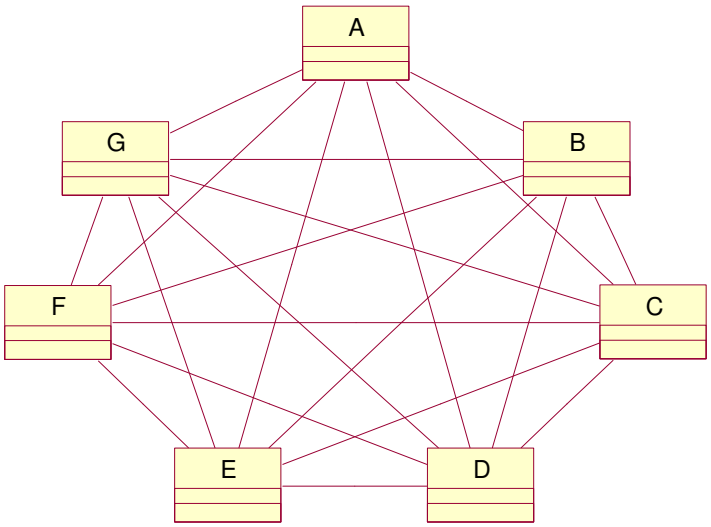
- ترسل الأغراض في البرامج– وليس الصفوف– رسائل لأغراض أخرى من الصف نفسه أو من صفوف أخرى مختلفة، ويضع هذا الأمر صعوبة إضافية أمام المبرمج المسؤول عن وضع منطق التطبيق وإدارة متحولات البرنامج وبنى المعطيات الأخرى
- التحدي الأساسي ليس في قياس تعقيد برنامج بذاته بل في قياس تعقيد نظام كامل من البرنامج

لا يمكن أن يرسل غرض ما رسالة لغرض آخر إلا إذا كان هناك ارتباط دائم أو مؤقت بينهما:

- تظهر الارتباطات المؤقتة في جلسة تشغيل البرنامج وبذلك لا تؤدي إلى زيادة تعقيد النظام الكلي
- تظهر الارتباطات الدائمة فقط عند وجود وصلة (اقتران مثلاً) بين صفوف النموذج. ويمكن أن تتشارك عدة برامج على استخدام الصفوف أو إعادة استخدامها

المنارة للاستشارات
www.manaraa.com

(الشكل 9):



طبقات الصفوف

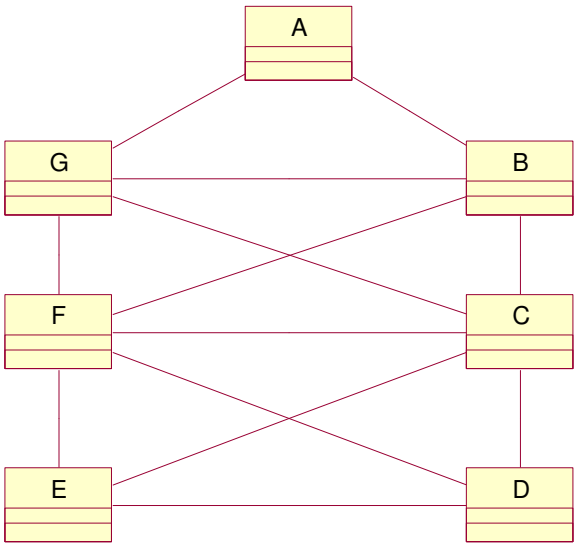
2 – تعقيد البنى الهرمية

- **البنية الهرمية:** تحل مشكلة السيطرة على تعقيد الحل باختصار البنى التحتية عبر تجميع الصفوف في بنى هرمية، وبهذه الطريقة تنظم الصفوف في طبقات مقسمة بصيغة هرمية، وتبقى إمكانية التخاطب بين صفوف الطبقة الواحدة كما هو في حالة البنية الشبكية

- **تعقيد البنية الهرمية:** تخفض البنية الهرمية درجة التعقيد بتحديد عدد مسارات التخاطب الممكنة بين الصفوف، إذ لا تسمح هذه البنية لصفين بالتخاطب مباشرة إلا إذا كانا من الطبقة نفسها أو من طبقتين متجاورتين
- **مثال:** يبين الشكل التالي (الشكل 10) تعقيد بنية هرمية تضم سبعة صفوف موزعة في أربع طبقات، وبالمقارنة مع البنية الشبكية المبينة في الشكل السابق (الشكل 9) انخفض التعقيد من 42 إلى 26 مسار تخاطب

المنارة للاستشارات
www.manaraa.com

(الشكل 10):



طبقات الصفوف

3 – الحزمة (package)

- **الحزمة:** تتضمن لغة UML تدويناً بيانياً للحزمة التي تمثل مجموعة من الصفوف (أو من عناصر نمذجة أخرى كحالات الاستخدام مثلاً). وتستخدم الحزم لتجزئة النموذج المنطقي لبرنامج تطبيقي. والحزمة هي مجموعة صفوف مترابطة بقوة، فهي متلاحمة بعضها ببعض لكنها ضعيفة الارتباط نسبياً بالمجموعات الأخرى.

- **تعشش الحزم:** يمكن أن تعشش الحزم بعضها داخل بعض مما يسمح بتقسيم الأنظمة الكبيرة إلى أنظمة جزئية ومجتزآت، إذ يمكن مثلاً أن تتضمن الحزمة ذات النمط المحيطي <<System>> عدة حزم من النمط المحيطي <<Subsystem>>. وتمتلك الحزمة الخارجية حق الوصول المباشر إلى أي صف في أي من الحزم المعششة داخلها.

- يمكن أن ينتمي الصف إلى حزمة واحدة فقط، لكن هذا لا يمنع الصف من الظهور في حزم أخرى أو من الاتصال مع صفوف في حزم أخرى، ويمكن التحكم بكيفية اتصال الصفوف الموجودة في حزم مختلفة بتعريفها ضمن الحزمة كصفوف عمومية أو خاصة أو محمية.

طبقات الصفوف

3 – الحزمة (package) (تتمة)

- **توصيف الحزمة:** تأخذ الحزمة شكل أيقونة المجلد التقليدية (كما هو موضح في الشكل التالي (الشكل 11):
  - ترسم الحزمة المعششة داخل الحزمة الخارجية

المنارة للاستشارات
www.manaraa.com

- كما يجب أن يكون لكل حزمة مخطط الصفوف الخاص بها الذي يعرف كل الصفوف التي تنتمي إلى الحزمة

- **العلاقات بين الحزم:** يمكن أن ترتبط الحزم بنوعين من العلاقات:

1. **الاعتماد (Dependency):** قد يكون السبب الأساسي والأهم لوجود علاقة الاعتماد بين حزمتين هو وجود غرض في إحداهما يرسل رسالة لغرض في الأخرى
2. **التعميم (Generalization):** ويؤدي وجود علاقة التعميم بين حزمتين إلى وجود علاقة الاعتماد ضمناً من حزمة النمط الأدنى إلى حزمة النمط الأعلى، وتؤثر التعديلات التي تُجرى على حزمة النمط الأعلى على حزمة النمط الأدنى

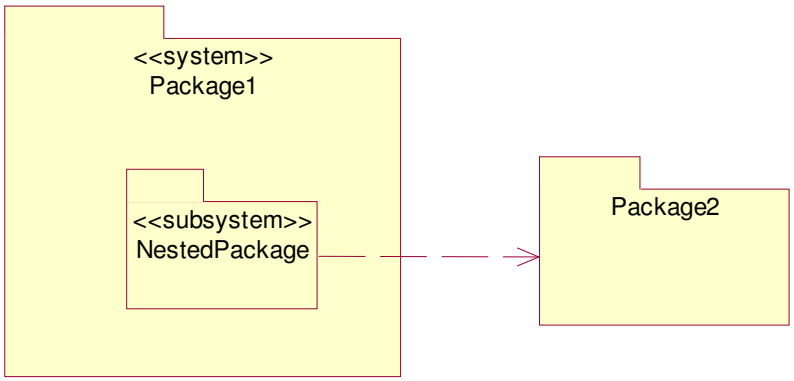
- **مخطط الحزم:** إن مفهوم مخطط الحزم غير موجود بذاته في UML بل يتم إنشاء الحزم بإحدى طريقتين:

1. في مخطط صفوف: ينمذج مخطط الحزم حالة النظام حيث يصف مخطط الحزم في هذه الحالة البنيان الرئيسي لإطار عمل النظام،

2. أما في مخطط حالات الاستخدام: فينمذج النظام من منطلق سلوكي، وتعطي الحزم وصفاً عالي المستوى لبنية النظام الوظيفية

- يفيد مخطط حزم الحالة في إدارة حجم وتعقيد النظام كما يشكل آلية هامة وأساسية للتعاون بين المطورين

(الشكل 11):



مثال – التسويق عبر الهاتف

1- نص المسألة

- **نص المسألة:**

تسعى إحدى الجمعيات الخيرية إلى زيادة رصيدها من خلال بيع بطاقات اليانصيب في حملات خيرية. وتحتفظ الجمعية بقائمة بأسماء من شاركوا سابقاً، بحيث تنتقى مجموعة جزئية من هذه الأسماء عند بدء حملة جديدة لبيعهم البطاقات عبر الهاتف أو الاتصال بهم مباشرة عبر البريد.

تتبع الجمعية أساليب خلاقة لكسب مساهمين جدد، بما في ذلك منح نقاط إضافية للمساهمين الذين يشترون مجموعة بطاقات. ولا تنتقي الجمعية الزبائن المحتملين عشوائياً باستخدام أدلة الهاتف والوسائل المشابهة.

المنارة للاستشارات
www.manaraa.com

ولدعم عملها قررت الجمعية التعاقد على تطوير تطبيق تسويق هاتفي جديد بحيث يدعم النظام الجديد حوالي خمسين مسوقاً يعملون معاً في الوقت نفسه، كما يجب أن يجدول النظام المكالمات الهاتفية تبعاً لأفضليات محددة مسبقاً ولشروط أخرى محددة.
يجب أن يستطيع النظام طلب الاتصالات المجدولة، وعند فشل الاتصال عليه أن يعيد جدولته بحيث يحاول إجراءه من جديد لاحقاً.
كما يجب أن يسمح النظام بتسجيل نتائج المكالمات بما فيها طلبات شراء البطاقات وأية تغيرات أخرى تخص المساهمين.

- **المطلوب:** أنشئ مخطط صفوف العمل للمسألة السابقة. قد تساعدك في ذلك الملاحظتان التاليتان:

1. إن المهمة الأساسية للنظام هي جدولة الاتصالات، وهي بحد ذاتها مسألة إجرائية: أي أن حلها هو حل خوارزمي بطبيعته. يجب أن تخزن أرتال المكالمات المجدولة ونتائج الاتصالات في بنية معطيات من نمط ما
2. كما ذكرنا سابقاً، فقد نحتاج لحفظ معلومات الفاعلين في الصفوف

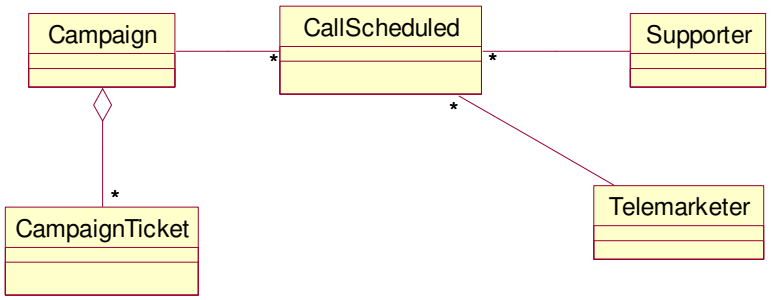
مثال – التسويق عبر الهاتف

2- الحل

- **الحل:**

يبين الشكل التالي (الشكل 13) صورة أولية لنموذج صفوف العمل، ويتضمن المخطط ستة صفوف، اشتُق اثنان منها(Supporter وTelemarketer) من فاعلي نموذج حالات استخدام العمل. تحصل خوارزمية جدولة الاتصالات على رقم هاتف، معلومات أخرى من الصف Supporter وتجدول المكالمة على قائمة اتصالات أحد المسوقين (Telemarketers) المتوفرين حالياً.

(الشكل 13):



مثال – التسويق عبر الهاتف

3 – الطلب 2

- **نص المسألة:** عد إلى نص المسألة وحل الطلب الأول وخذ بالحسبان المعلومات الإضافية التالية:

1. لكل حملة عنوان يستخدم عموماً للإشارة إليها، ورمز وحيد يستخدم كمرجع داخلي، وتستغرق كل حملة فترة محددة من الزمن، ويجري سحب الجوائز بعد إغلاق الحملة مباشرة ويُعلم حاملو البطاقات الرابحة.
2. ترقم كل البطاقات الخاصة بالحملة بحيث تتميز كل بطاقة برقم وحيد، ويجب أن يبقى عدد بطاقات الحملة الكلي معروفاً إلى

المنارة للاستشارات
www.manaraa.com

جانب عدد البطاقات المباعة سابقاً والوضع الحالي لكل بطاقة (متوفرة، مطلوبة للشراء، مدفوع ثمنها، رابحة الجائزة).

3. لتحديد مستوى أوامر مسوقي الجمعية يجب أن يسجل النظام مدة كل مكالمة هاتفية وخلاصات المكالمات الناجحة (أي التي ينتج عنها طلب شراء بطاقات).
4. يجب أن يحتفظ النظام بمعلومات شاملة عن المساهمين، وتتضمن هذه المعلومات إلى جانب المعلومات العادية (العنوان، ورقم الهاتف، وغيرها)، تفاصيل تاريخية مثل تاريخ أول مساهمة وآخر مساهمة له في حملة مع عدد الحملات التي ساهم فيها. كما يحتفظ النظام بأية قيود أو تفضيلات خاصة بالمساهم (مثل الأوقات التي لا يرغب أن يتصل فيها أحد به، أو بطاقة الائتمان التي يستخدمها في العادة لشراء البطاقات).
5. يجب إعطاء أولويات لمعالجة اتصالات التسويق. يجب إعادة جدولة المكالمات التي لم يُجِب عليها أحد أو تلك التي أجابت عليها آلة الهاتف لتعاد محاولة الاتصال بها لاحقاً. ومن المهم عند تكرار محاولة الاتصال برقم اختيار أوقات بديلة لوقت الاتصال السابق.
6. يمكن أن نكرر محاولات الاتصال إلى أن يبلغ عدد المحاولات حداً معيناً، وقد تختلف قيمة هذا الحد من نمط اتصالات لنمط آخر. فقد يختلف مثلاً عدد المحاولات المسموحة لاتصال "تسويق" عادي عن عدد المحاولات المسموحة لاتصال لتذكير المساهم بدفع ثمن بطاقات.
7. لتسهيل إدخال المعطيات من قبل المسوقين يفضل تصنيف خلاصات الاتصال، أي: ناجح (تم طلب شراء بطاقات)، غير ناجح، أعد الاتصال لاحقاً، بلا إجابة، تعهد بالشراء، جهاز إجابة آلي، جهاز فاكس، رقم خاطئ، عدم التمكن من الاتصال.

- **المطلوب:** ضع توصيف صفوف المسألة مع أخذ النقاط السابقة بعين الاعتبار.

مثال – التسويق عبر الهاتف

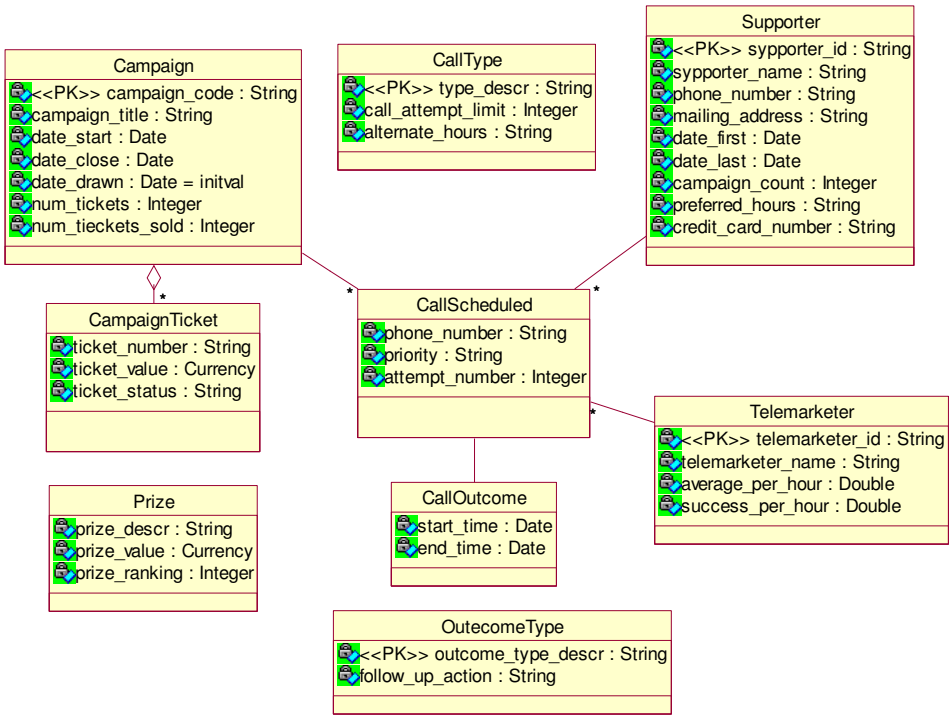
2- حل الطلب 2

- **الحل:**

يبين الشكل التالي (الشكل 14) نموذج الصفوف الناتج عن المناقشة السابقة. وقد احتفظنا بعلاقات الاقتران التي تبينت عند إنشاء نموذج صفوف العمل، ولم نضف إليها أية علاقات اقتران جديدة.

(الشكل 14):

المنارة للاستشارات
www.manaraa.com



مثال – التسويق عبر الهاتف

5 – الطلب 3

- **نص المسألة:**

عد إلى نص المسألة وإلى الطلب الثاني. يتضمن نص المسألة الملاحظة التالية: "يتضمن نظام العمل حملات إضافية خاصة لمكافأة المساهمين الذين يشترون بطاقات ولاجتذاب مساهمين جدد" ولم تنمذج هذه الملاحظة بعد.

لنفترض أن إحدى هذه الحملات الإضافية تتضمن /دفاتر بطاقات/، فإذا اشترى المساهم دفتراً كاملاً يحصل على بطاقة إضافية مجانية في الحملة الأساسية. والمطلوب:

المطلوب: حدّث نموذج الصفوف بحيث يتضمن الصف Bonus Campaign (حملة إضافية).

المنارة للاستشارات
www.manaraa.com

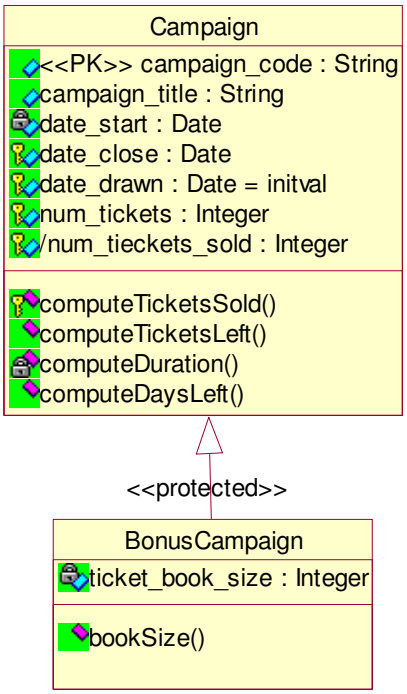
مثال – التسويق عبر الهاتف

6 – حل الطب 3

الحل:

يبين الشكل التالي (الشكل 15) نموذج الصفوف بعد إجراء التعديلات الموافقة للملاحظات المذكورة في الطب 3.

(الشكل 15):



www.manaraa.com

القسم الثاني عشر

مخططات التعاون

الكلمات المفتاحية:

نموذج النشاط، حالة النشاط، الانتقال، مخطط النشاط، نموذج التفاعل، مخطط التفاعل، نموذج التسلسل، مخطط التسلسل، نموذج التعاون، مخطط التعاون، نموذج الحالات، مخطط الحالات.

ملخص:

يُركِّز هذا الفصل على نماذج التفاعل والمخططات الخاصة بها.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- التعرف على نمذجة النشاط (الأنشطة، حالات النشاط، مخطط النشاط).
- نمذجة التفاعل.
- مخطط التسلسل.
- مخطط التعاون.
- نموذج الحالات (الحالات والانتقالات، مخطط الحالات).

المنارة للاستشارات
www.manaraa.com

نموذج النشاط

- يُعبِّر نموذج النشاط (activity model) عن تدفق الأحداث في حالة الاستخدام بصيغة بيانية، ويهدف إلى تقليص المسافة بين تمثيل سلوك النظام عالي المستوى (حالات الاستخدام)، وبين تمثيل سلوك النظام ذي المستوى المتدني (مخططات التسلسل ومخططات التعاون).
- يُبيّن مخطط النشاط خطوات إنجاز عملية ما، وكل خطوة هي حالة من فعل ما، وتدعى هذه الحالات (حالات النشاط)، بينما يدعى تدفق التحكم بين الحالات (انتقال).
- يُوضِّح هذا المخطط الخطوات التي يجب أن تُنفذ بالتسلسل وتلك التي يمكن تنفيذها متزامنة ضمن حالة الاستخدام.

يُعبِّر نموذج النشاط (activity model) عن تدفق الأحداث في حالة الاستخدام بصيغة بيانية، ويهدف إلى تقليص المسافة بين تمثيل سلوك النظام عالي المستوى (نماذج حالات الاستخدام) من جهة، وبين تمثيل سلوك النظام ذي المستوى المتدني (نماذج التفاعل، أي مخططات التسلسل ومخططات التعاون) من جهة أخرى.

يُبيّن مخطط النشاط خطوات إنجاز عملية ما، وكل خطوة هي حالة من فعل ما، ولذلك تدعى خطوات التنفيذ حالات النشاط.
يُوضّح هذا المخطط الخطوات التي يجب أن تنفذ بالتسلسل وتلك التي يمكن تنفيذها متزامنة، ويدعى تدفق التحكم من حالة نشاط إلى الحالة التالية انتقالاً (transition).
يمكن استنتاج حالات النشاط من توصيف تدفق الأحداث الرئيسي والتدفقات البديلة في وثيقة توصيف حالة الاستخدام، لكن هنالك فارق هام بين توصيف حالة الاستخدام ونموذج النشاط، إذ يكتب توصيف حالة الاستخدام من وجهة نظر فاعل خارجي، أما نموذج النشاط فيمثل منظور النظام من الداخل.

يمكن الاستفادة من نماذج النشاط في نواح أخرى من عملية التطوير بعيداً عن نمذجة حالات الاستخدام، إذ يمكن الاعتماد عليها لفهم إجرائية العمل على مستوى عال من التجريد وقبل توصيف أي من حالات الاستخدام، كما يمكن بالمقابل استخدامها على مستوى أدنى لتصميم خوارزميات تسلسلية معقدة أو لتصميم إجرائيات متنافسة في تطبيقات متعددة مسارات المعالجة.

مخطط النشاط

- يُبيّن مخطط النشاط الانتقالات بين الأنشطة، ولهذا المخطط حالة نشاط بدائية واحدة، وحالة نشاط نهائية أو أكثر إلا إذا كان يمثل حلقة مستمرة
- تُمثَّل الحالة البدائية بدائرة سوداء، بينما تُمثَّل الحالة النهائية بدائرة سوداء ضمن دائرة أكبر
- يمكن أن يتفرع الانتقال أو تدمج عدة انتقالات مما يولد مسارات حسابية بديلة، ويُمثِل المعين شرط التفريع
- يمكن للانتقالات أن تتشعب أو تنضم، مما يولد مسارات حسابية متنافسة أو متوازية، ويُمثَّل التشعب/ الانضمام بخط مستقيم

مثال على نموذج النشاط

- سنقوم في هذا المثال بعرض مخطط النشاط لحالة استخدام (Rent Video) وهي إحدى حالات الاستخدام في نظام لإدارة مخزن فيديو

المنارة للاستشارات
www.manaraa.com

- إن التوصيف السردي لهذه الحالة يُلخص في الجدول التالي:

حالة الاستخدام	Rent Video
وصف موجز	يرغب زبون باستئجار شريط أو قرص فيديو انتقاه من الأفلام المعروضة على الرف أو حجزه مسبقاً. فإذا لم يكن الزبون مقصراً أو ممن يتأخرون بالدفع يعار له الشريط فور دفع الأجرة. وإذا لم يُرجع الزبون الشريط خلال الفترة المسموحة ترسل له ملاحظة بتوجب دفع قيمة إضافية.
الفاعلون	الموظف (Employee)، جهاز المسح (Scanning Device).
الشروط السابقة	الشريط أو القرص متوفر وجاهز للإعارة. يملك الزبون بطاقة عضوية. أجهزة المسح تعمل. يعرف الموظف الموجود خلف المكتب كيف يستخدم النظام.
التدفق الرئيسي	يسأل الزبون الموظف عن توفر فيلم ما (قد يكون حجز الفيلم سابقاً) أو قد ينتقي شريطاً أو قرصاً عن الرف مباشرة. يمسح الشريط وبطاقة العضوية بحيث يستطيع الموظف أن يعرف إذا كان هناك تأخر بالدفع أو توجب دفع قيم إضافية ليسأل الزبون. إذا لم يكن الزبون مديناً يمكن أن يستأجر 8 أفلام كحد أقصى. أما إذا كان ترتيب الزبون "غير موثوق" فيطلب منه أن يدفع أجرة فترة الإعارة الأساسية عن كل شريط أو قرص. وعند تلقي المبلغ المطلوب يُحدَّث المخزون وتُسلم الأشرطة والأقراص للزبون إلى جانب إيصال الدفع. يمكن أن يدفع الزبون نقداً أو ببطاقة ائتمان أو إلكترونياً. يخزن كل سجل تأجير (تحت حساب الزبون) تاريخ الاستئجار وتاريخ توجب الإعادة إلى جانب معلومات التعريف بالموظف. ويتم إنشاء سجل مستقل لكل فيلم مستأجر. تولد حالة الاستخدام ملاحظة للزبون بتوجب دفع قيمة إضافية إذا لم يعد الفيلم خلال يومين من تاريخ توجب إعادته، وتولد ملاحظة ثانية إذا تأخر يومين آخرين (وعندئذ يصنف الزبون كزبون "مقصر").
التدفقات البديلة	لا يملك الزبون بطاقة عضوية. تُفعَّل عندئذ حالة الاستخدام "Maintain Customer" لإصدار بطاقة جديدة. يحاول الزبون استئجار عدد كبير من الأفلام. لا يمكن تأجير الزبون أي فيلم بسبب تصنيفه كزبون مقصر. تعذر مسح وسيط تخزين الفيلم أو بطاقة العضوية بسبب وجود خلل فيهما. رفض تحويل النقد الإلكتروني أو الدفع ببطاقة الائتمان.
الشروط اللاحقة	تم تسليم الأفلام وتحديث قاعدة المعطيات بما يتفق مع الوقائع الجديدة.

- يظهر في الشكل التالي مخطط النشاط لحالة الاستخدام Rent Video، ويعكس هذا المخطط وضوحاً التوصيف السردي لحالة الاستخدام (كما في الجدول السابق)، فتبدأ المعالجة بمسح وسيط التخزين وبطاقة الزبون، وهما نشاطان مستقل أحدهما عن الآخر (تظهر هذه الحقيقة على المخطط بعلامة التشعّب).

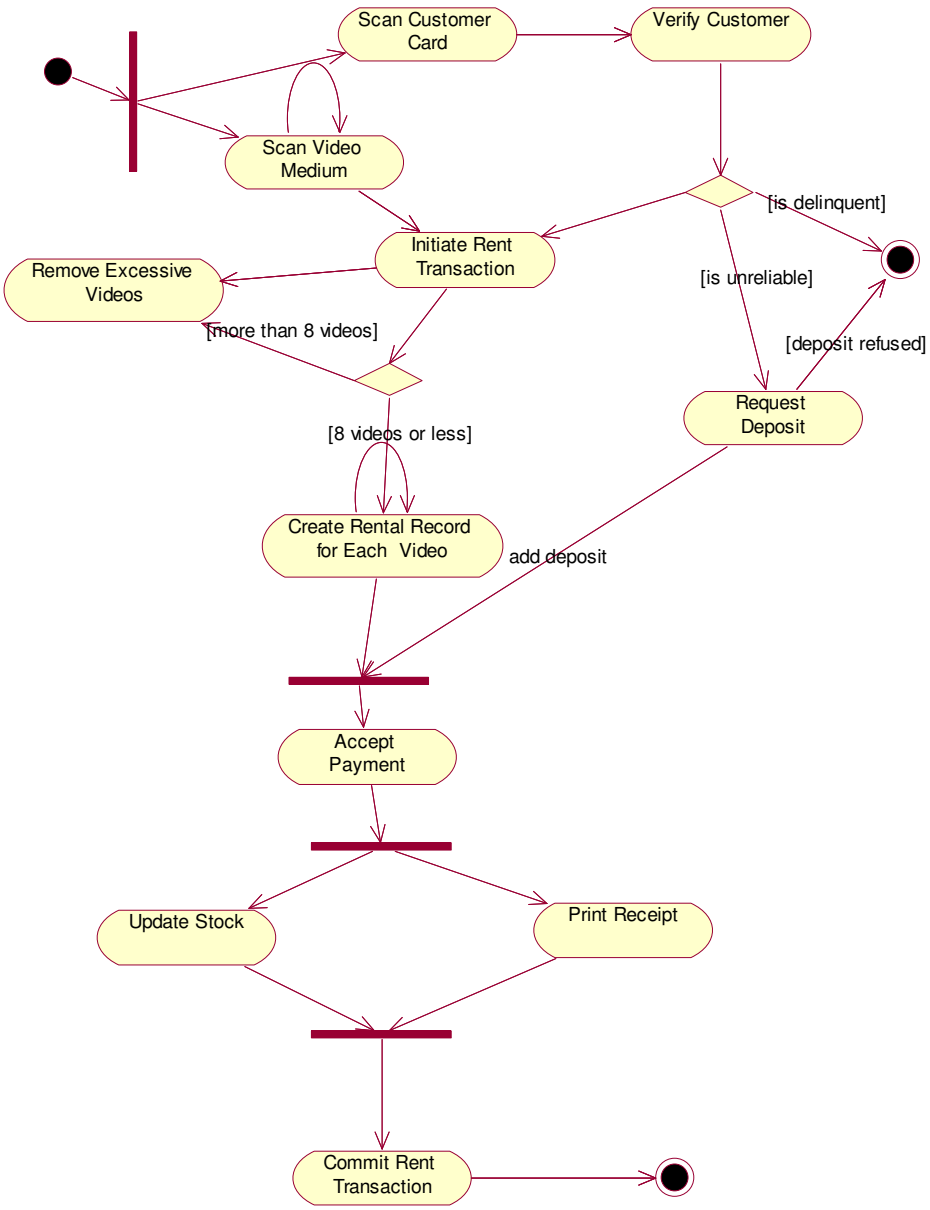
يتحقق النشاط Verify Customer من تاريخ الزبون، لتستمر المعالجة بعد اختبار شروط التفريع، فإذا كان مقصراً (delinquent)

المنارة للاستشارات
www.manaraa.com

يُنهى تنفيذ الحالة Rent Video، أما إذا لم يكن الزبون موثوقاً (unreliable) فسيطلب منه أن يدفع الأجور قبل إكمال المناقلة، وإذا كان سجله نظيفاً يُفعّل النشاط Initiate Rent Transaction.

يتحقق شرط التفريع الثاني من عدم إعارة أكثر من 8 أفلام للزبون نفسه. وبعد قبول الدفع يتشعب التنفيذ إلى مسارين متزامنين للنشاطين Update Stock وPrint Receipt، ثم ينضم هذان المساران قبل إطلاق النشاط Commit Rent Transaction حيث تنتهي المعالجة بانتهائه.

- مخطط النشاط لحالة الاستخدام "Rent Video":



www.manaraa.com

نموذج التفاعل

- تُصوِّر نمذجة التفاعل، التفاعلات اللازمة بين مجموعة أغراض لتنفيذ حالة استخدام، وتُستخدَم في مرحلة متقدمة من تحليل المتطلبات بعد أن يكون نموذج الصفوف الأساسي قد اكتمل.
- من هنا نلاحظ الفارق الأساسي بين نمذجة النشاط ونمذجة التفاعل، فكلا النموذجين يصور سلوك النظام في حالة استخدام واحدة، لكن تصور نمذجة النشاط هذا السلوك على مستوى تجريد أعلى، فهي تُبيّن تسلسل الأحداث لكن دون ربط هذه الأحداث بالأغراض، أما نمذجة التفاعل فتبين تسلسل الأحداث (الرسائل) بين أغراض متعاونة.
- يوجد نوعان من مخططات التفاعل: مخطط التسلسل ومخطط التعاون.

تُصوِّر نمذجة التفاعل التفاعلات اللازمة بين مجموعة أغراض لتنفيذ حالة استخدام، وتُستخدَم في مرحلة متقدمة من تحليل المتطلبات وبعد أن يكون نموذج الصفوف الأساسي قد أصبح معروفاً بحيث يُستخدَم هذا النموذج كمرجع للأغراض.
من هنا نلاحظ الفارق الأساسي بين نمذجة النشاط ونمذجة التفاعل، فكلا النموذجين يُصوِّر سلوك النظام في حالة استخدام واحدة، لكن تُصوِّر نمذجة النشاط هذا السلوك على مستوى تجريد أعلى فهي تبين تسلسل الأحداث لكن دون ربط هذه الأحداث بالأغراض، أما نمذجة التفاعل فتبين تسلسل الأحداث (الرسائل) بين أغراض متعاونة.
يوجد نوعان من مخططات التفاعل: مخطط التسلسل Sequence Diagram ومخطط التعاون Collaboration Diagram. لكن ثمة تكافؤ بين هذين المخططين، ويمكن في الواقع استخدام أي منهما، وتوفر معظم الأدوات المساندة في هندسة البرمجيات إمكانية تحويل أحد النموذجين آلياً إلى النموذج الآخر. أما الفارق بين النموذجين فيتلخص في أن نموذج التسلسل يُركِّز على التسلسل الزمني بينما يُركِّز نموذج التعاون على العلاقات بين الأغراض.

مخطط التسلسل

- يُرسَم مخطط التسلسل كبيان ثنائي الأبعاد، فتظهر الأغراض على بعده الأفقي بينما يظهر تسلسل الرسائل على بعده الشاقولي من الأعلى باتجاه الأسفل.
- يُدعى كل خط شاقولي مرسوم على محاذاة أحد الأغراض بخط حياة الغرض.
- في هذا المخطط يُمثِل السهم رسالة من غرض مستدع (أو مرسل) إلى عملية (أو طريقة) في الغرض المستدعى (أو المستهدَف). ويظهر لكل رسالة اسمها على الأقل، ويمكن أن تظهر وسطاء الرسالة وبعض معلومات التحكم الأخرى.
- يرسم عادة مخطط تسلسل مستقل لكل حالة من حالات الاستخدام.

يُرسَم مخطط التسلسل كبيان ثنائي الأبعاد، فتظهر الأغراض على بعده الأفقي بينما يظهر تسلسل الرسائل على بعده الشاقولي من الأعلى باتجاه الأسفل. ويُدعى كل خط شاقولي مرسوم على محاذاة أحد الأغراض بخط حياة الغرض lifeline.
في هذا المخطط يمثل السهم رسالة من غرض مستدع (أو مرسل) إلى عملية (أو طريقة) في الغرض المستدعى (أو المستهدَف). ويَظهر لكل رسالة اسمها على الأقل، ويمكن أن تظهر وسطاء الرسالة وبعض معلومات التحكم الأخرى، ويجب أن تتوافق الوسطاء الفعلية مع الوسطاء الصورية في طريقة الغرض المستهدَف.
قد يكون الوسيط الفعلي وسيط دخل (من المرسل إلى المستهدَف) أو وسيط خرج (يُعاد من المستهدَف إلى المرسِل)، ويمكن تمييز وسيط الدخل بالكلمة المفتاحية in (ونصطلح أن غياب أية كلمة أمام الوسيط يعني ضمناً أنه وسيط دخل) فيما قد نستخدم الكلمة المفتاحية Out لتمييز وسيط الخرج.

المنارة للاستشارات
www.manaraa.com

يرسم عادة مخطط تسلسل مستقل لكل حالة من حالات الاستخدام، وبما أنه يفضل أن يعبر عن كل حالة استخدام بمخطط نشاط، يمكن بناء مخطط تسلسل لكل مخطط من مخططات النشاط، فمن أساسيات النمذجة الجيدة معاينة النظام نفسه من عدة مناظير مترابطة فيما بينها.

مخطط التعاون

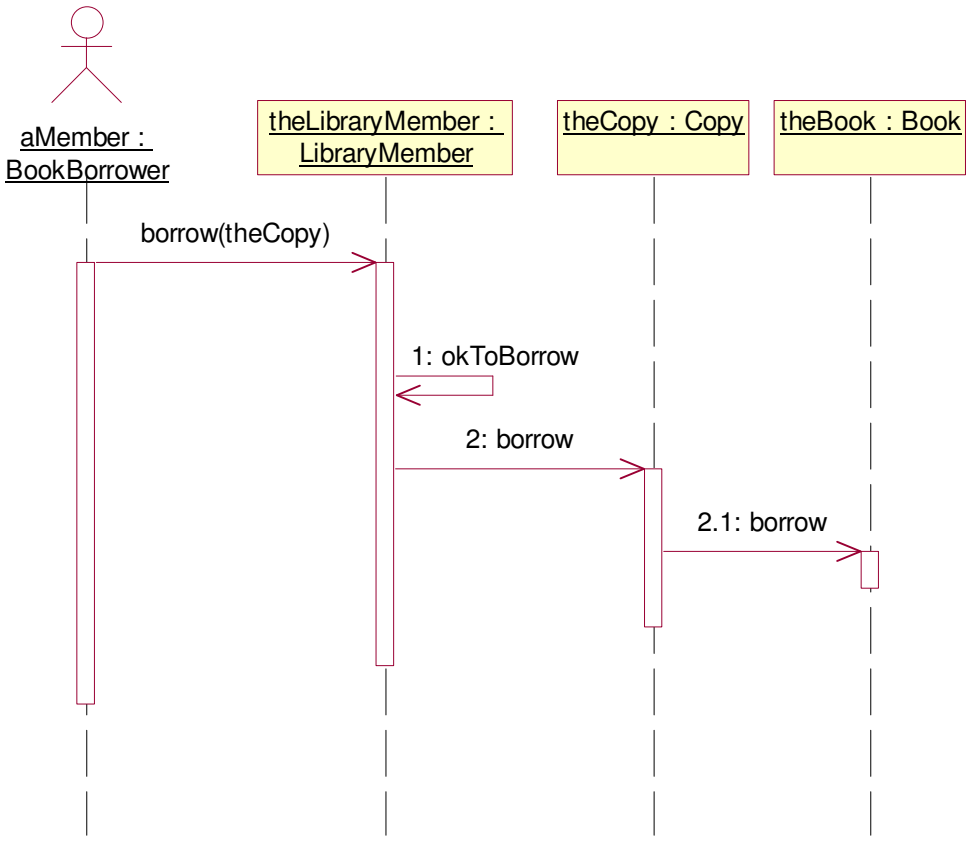
- يُعتبر مخطط التعاون أحد شكلي نموذج التفاعل في UML ويُفضل استخدامه في مرحلة التصميم.
- هناك تكافؤ بين مخطط التعاون ومخطط التسلسل من حيث أنه يمكن تحويل أحدهما إلى الآخر تلقائياً، لكن مع الانتباه إلى أن كلاً منهما يُركِّز على ملامح مختلفة لعملية تفاعل الأغراض.
- يُركِّز مخطط التسلسل على التسلسل الزمني لتبادل الرسائل بين الأغراض، لكنه يفتقر إلى الدقة في تمثيل المسارات البديلة للرسائل.
- يعرض مخطط التعاون بصيغة صريحة العلاقات السكونية بين الأغراض التي يمكن أن يجري تبادل الرسائل عبرها، فتتميز بدقة أكبر عند الحاجة لمعاينتها من الناحية الشكلية.

يُعتبَر مخطط التعاون أحد شكلي نموذج التفاعل في UML والشكل الآخر هو مخطط التسلسل (الذي عرضناه سابقاً)، ونُفضِل استخدام مخططات التسلسل في التحليل ومخططات التعاون في التصميم.
هناك تكافؤ بين مخطط التعاون ومخطط التسلسل من حيث أنه يمكن تحويل أحدهما إلى الآخر تلقائياً، مع الانتباه إلى أن كلاً منهما يُركِز على ملامح مختلفة لعملية تفاعل الأغراض (يمكن أن تُفقَد هذه الملامح أثناء التحويل).
يُركِّز مخطط التسلسل على التسلسل الزمني لتبادل الرسائل بين الأغراض، لكنه يفتقر إلى الدقة في تمثيل المسارات البديلة للرسائل، كما أنها مزعجة أيضاً عند تمثيل التعاون بين عدد كبير من الأغراض (يُحسِّن ترتيب خطوط الأغراض إمكانية قراءة النموذج).
يعرض مخطط التعاون بصيغة صريحة العلاقات السكونية بين الأغراض التي يمكن أن يجري تبادل الرسائل عبرها، فتتميز هنا بدقة أكبر عند الحاجة لمعاينة الرسائل من الناحية الشكلية.

مثال لنموذج التسلسل

- ليكن لدينا المثال التالي: حالة استخدام "عملية استعارة نسخة كتاب من مكتبة":
  - يقوم الشخص الذي يرغب بالاستعارة باحضار الكتاب المطلوب
  - يقوم النظام بالتحقق من أن هذا الشخص عضو في المكتبة، وأنه لم يقم باستعارة الحد الأقصى من الكتب
  - إذا تمت عملية التحقق بنجاح، يقوم النظام بتسجيل عملية إعارة نسخة الكتاب لهذا الشخص
- الصفوف الموجودة في المثال: الشخص المسؤول عن المكتبة (LibraryMember)، صف نسخة الكتاب (Copy)، صف الكتاب (Book)
- فاعلو النظام: الشخص الذي يقوم بعملية الاستعارة (BookBorrower)
- أغراض النظام: غرض من الصف (LibraryMember)، غرض من الصف (Copy)، غرض من الصف (Book)، نسخة من الفاعل (BookBorrower)
- مخطط التسلسل لحالة الاستخدام السابقة:

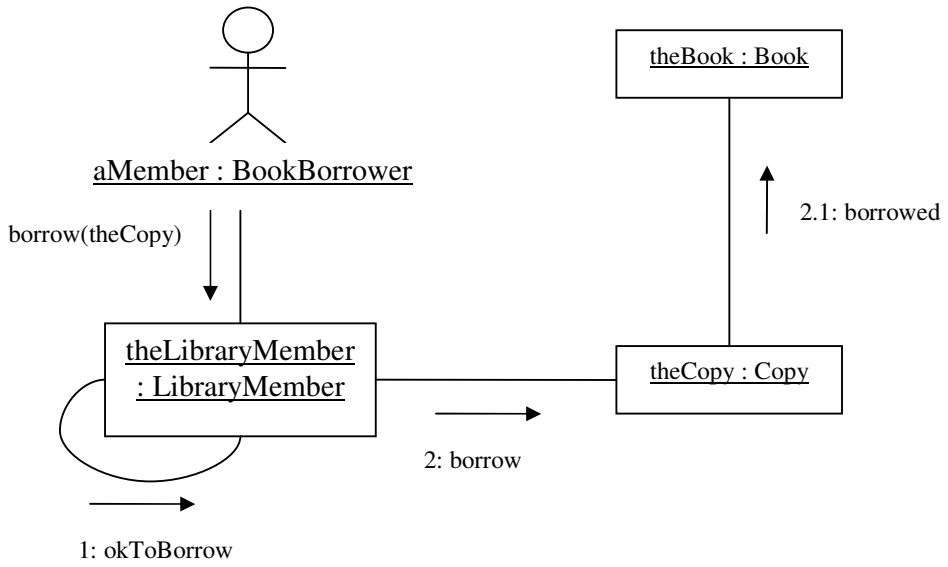
المنارة للاستشارات
www.manaraa.com



مثال لنموذج التعاون

- بالعودة إلى المثال السابق: حالة استخدام "عملية استعارة نسخة كتاب من مكتبة":
- فإن مخطط التعاون للحالة السابقة:

www.manaraa.com



نموذج مخطط الحالات

- يُعطي نموذج مخطط الحالات وصفاً تفصيلياً لصف ما
- يصور نموذج مخطط الحالات حياة الصف ويبقى الغرض هو نفسه، أي لا تتغير هويته، بل تتغير حالته فقط
- إن مخطط الحالات هو بيان يظهر الحالات (مستطيل مستدير الزوايا) والانتقالات (الأسهم) التي تسببها الأحداث
- مفاهيم الحالات والأحداث هي نفسها التي تستند إليها مخططات النشاط لكن مع فارق بسيط إذ تمثل الحالات في مخطط النشاط حالات تنفيذ الحساب وليس حالات غرض عادي

يعطي نموذج التفاعل توصيفاً تفصيلياً لحالة استخدام, بينما يعطي نموذج مخطط الحالات وصفاً تفصيلياً لصف ما, أو إذا شئنا الدقة، للتغيرات الديناميكية لحالات الصف. وتصف هذه التغيرات عادة سلوك غرض ما عبر عدة حالات استخدام.
تتعين حالة الغرض بالقيم الحالية لصفاته (الصفات الأولية والصفات التي تشير إلى صفوف أخرى)

يُصوِّر هذا النموذج، الحالات التي يمكن أن يتواجد فيها صف ما, أي يُصور تاريخ حياة الصف ويبقى الغرض هو نفسه (لا تتغير هويته)، بل تتغير حالته فقط.
إن مخطط الحالات هو بيان يُظهِر الحالات (مستطيل مستدير الزوايا) والانتقالات (الأسهم) التي تسببها الأحداث. أما مفاهيم الحالات والأحداث فهي المفاهيم نفسها التي تستند إليها مخططات النشاط لكن مع فارق بسيط إذ تمثل الحالات في مخطط النشاط حالات تنفيذ الحساب وليس حالات غرض عادي.

الحالات والانتقالات

- تُغيِّر الأغراض قيم صفاتها لكن لا تسبب كل هذه التغيرات انتقالاً من حالة لأخرى
- لنأخذ على سبيل المثال الغرض (BankAccount) الذي يمثل حساباً مصرفياً، ولنفترض أن من قواعد العمل في المصرف أن

المنارة للاستشارات
www.manaraa.com

يوقف اقتطاع الرسوم من الحساب عندما يتجاوز الرصيد (balance) القيمة 100,000$. ونقول عندئذ إن الحساب قد دخل حالة مميزة، وقبل بلوغ هذه القيمة يكون في الحالة الطبيعية

- نلاحظ في هذا المثال أن قيمة الصفة (balance) تتغير بعد كل مناقلة، لكن لا تتغير حالة الحساب إلا عندما تتجاوز قيمة هذه الصفة الحد 100,000 (زيادة أو نقصاً)
- يُبيّن هذا المثال جوهر نمذجة الحالات، إذ ننشئ عادة نماذج الحالات للصفوف التي تطرأ عليها تغيرات هامة وليس أي تغير

مخطط الحالات

- يرتبط مخطط الحالات عادة بصف، لكن يمكن عموماً ربطه بمفاهيم أخرى كحالة استخدام مثلاً.
- يحدد هذا المخطط كيف تستجيب أغراض صف معين للأحداث
- يحدد هذا المخطط (لكل حالة من حالات الغرض) الفعل الذي سيقوم به عند وقوع حدث، وقد يقوم الغرض نفسه بعدة أفعال مختلفة استجابة للحدث نفسه

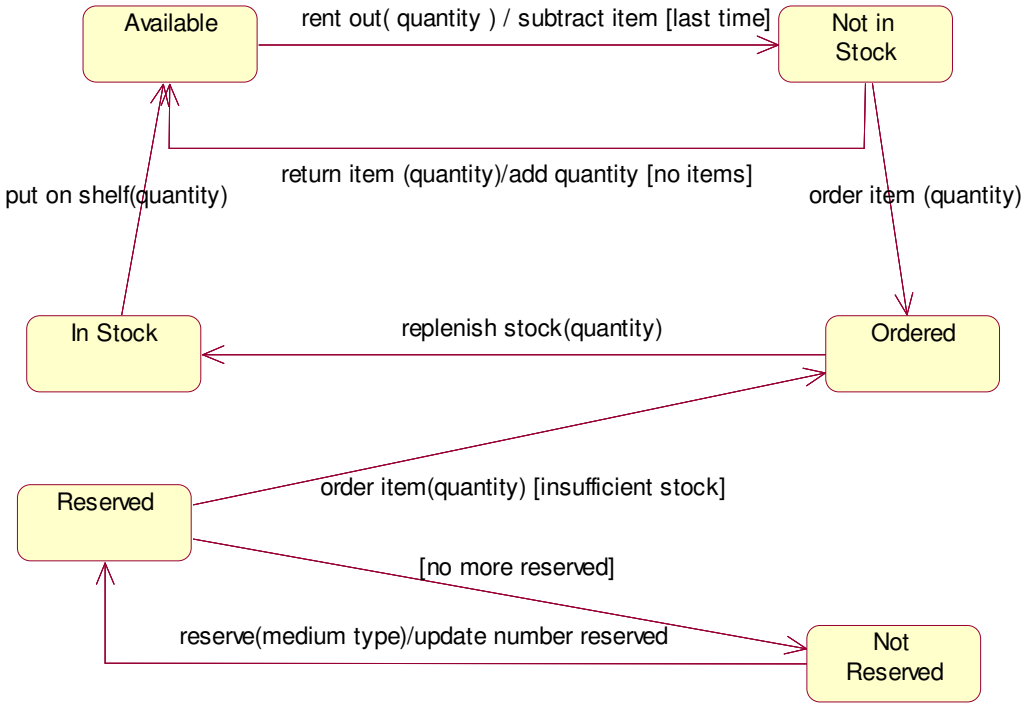
يرتبط مخطط الحالات عادة بصف، لكن يمكن عموماً ربطه بمفاهيم أخرى كحالة استخدام مثلاً، ويحدد هذا المخطط عند ربطه بصف من الصفوف كيف تستجيب أغراض الصف للأحداث، وبصيغة أكثر دقة، يحدد هذا المخطط (لكل حالة من حالات الغرض) الفعل الذي سيقوم به عند وقوع حدث، وقد يقوم الغرض نفسه بعدة أفعال مختلفة استجابة للحدث نفسه، ويتحدد الفعل المناسب عندئذ تبعاً لحالة الغرض، ويؤدي تنفيذ الفعل عادة إلى تغير حالة الغرض.

مثال لنموذج الحالات

- بالعودة إلى نظام إدارة مخزن الفيديو، وليكن لدينا الصف التالي MovieTitle، ولندرس الحالات التي يمر بها هذا الصف، ومن ثم نرسم مخطط الحالات الموافق
- يبين المخطط التالي مخطط حالات الصف MovieTitle:

المنارة للاستشارات
www.manaraa.com

Movie Title



www.manaraa.com

القسم الثالث عشر والرابع عشر

نمذجة واجهات الاستخدام

الكلمات المفتاحية:

واجهة الاستخدام البيانية (GUI)، سيطرة المستخدم، البرنامج المساق بالأحداث، مبدأ الانسجام، تشخيص واجهة الاستخدام البيانية، تخصيص واجهة الاستخدام البيانية، التسامح، التغذية الراجعة، النواحي الجمالية، قابلية الاستخدام، النوافذ الأولية، النوافذ الثانوية، مستعرض الأسطر، مستعرض الشجرة، صفحة الوب، علبة الحوار، مجلد علامات الجدولة، القائمة المنسدلة، علبة الرسالة، مكتبة MFC، الوثيقة، المعاينة، الواجهة وحيدة الوثيقة، SDI، الواجهة متعددة الوثائق، MDI، النافذة الأب، النافذة الابن، مخطط النشاط، الأنماط المحيطية، نشاط، حالة، مخطط التجوال عبر النوافذ.

ملخص:

تركز هذه الوحدة على التعرف على المفاهيم الأساسية في تصميم واجهات الاستخدام، حيث تستعرض القواعد الأساسية في عملية التصميم وكيفية الانتقال من التصميم إلى التحقيق، كما تقدم مجموعة من التوجيهات الهامة في تصميم الواجهات، وتعرفنا على أنواع الواجهات الأساسية مع أمثلة عن كل نوع، وكيفية تصميم الواجهات والتقنيات المستخدمة في عملية التصميم هذه. وتختتم هذه الوحدة بمجموعة من الأمثلة الشاملة حول تصميم الواجهات.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- تصميم الواجهة نشاط متعدد الاختصاصات
- من نموذج الواجهة الأولي إلى التحقيق
- توجيهات عامة لتصميم الواجهات
  - السيطرة للمستخدم
  - مبدأ الانسجام
  - التشخيص والتخصيص
  - التسامح
  - التغذية الراجعة
  - النواحي الجمالية وقابلية الاستخدام
- نوافذ الواجهة
  - مقدمة
  - النافذة الأولية
    - مستعرض الأسطر
    - مستعرض الشجرة
    - صفحة الوب
  - النافذة الثانوية

المنارة للاستشارات
www.manaraa.com

- علبة الحوار
- مجلد علامات الجدولة
- القائمة المنسدلة
- علبة الرسالة

- الترابط بين النوافذ
  - الوثيقة والمعاينة
  - الواجهة وحيدة الوثيقة
  - الواجهة متعددة الوثاق
- التجوال عبر النوافذ
  - تنميط مخطط النشاط بالأنماط المحيطية الملائمة لتمثيل التجوال بين النوافذ
  - مخطط التجوال عبر النوافذ
- مثال – التسويق الهاتفي
- مثال – إدارة العلاقات

المنارة للاستشارات
www.manaraa.com

تصميم الواجهة نشاط متعدد الاختصاصات

- إن تصميم واجهة الاستخدام البيانية (GUI) نشاط متعدد الاختصاصات، فهو يتطلب توفر مهارات متعددة من فريق كامل، ومن الصعب أن يمتلك شخص بمفرده كل المعارف التي يتطلبها تصميم الواجهة. يتطلب تصميم واجهة الاستخدام البيانية الجيد جمع مهارة فنان ومحلل متطلبات ومصمم ومبرمج وخبير تكنولوجي وعلماء اجتماع وسلوك. بالإضافة إلى كفاءات أخرى تعتمد على طبيعة النظام.

- تبدأ إجرائية تصميم واجهات الاستخدام البيانية لتطبيقات نظم المعلومات مع حالات الاستخدام، إذ يمتلك المحلل الذي يصف تدفق الأحداث في حالة من حالات الاستخدام تصوراً ما لواجهة الاستخدام التي تدعم التخاطب بين الإنسان والآلة، وقد يدرج المحلل في بعض الحالات وصفاً بيانياً لواجهات الاستخدام في وثيقة حالات الاستخدام.

- لا يمكن وصف التخاطب المعقد بين الإنسان والآلة بنصوص سردية فقط، وتتطلب عادة عملية جمع متطلبات الزبون ومناقشتها إعداد مشاهد أولية لواجهات الاستخدام البيانية.

- يجب أن يمتلك المصمم المعني بتوصيف التعاونات التي تحقق حالات الاستخدام صورة واضحة لشاشات واجهات الاستخدام البيانية. فإذا لم يضع المحلل صوراً كهذه من قبل يكون المصمم أول شخص يضع تصوراً لواجهات الاستخدام، وعليه أن يضعها بما ينسجم مع التقانة المعتمدة. وقد يحتاج هنا لاستشارة خبير تكنولوجي حول كيفية استثمار مزايا التقنيات الموجودة بنجاح.

- يجب إعداد نماذج أولية لواجهات الاستخدام البيانية قبل تسليم تصميم التعاون للمبرمجين كي يدققوها، وقد تتطلب هذه المهمة تدخل فنانين وأخصائيين بعلم الاجتماع والسلوك، إذ يستطيعون معاً أن يقدموا واجهة استخدام بيانية جذابة وقابلة للاستخدام.

من نموذج الواجهة الأولي إلى التحقيق

- **سيطرة المستخدم:** إن الفكرة المحورية في تصميم واجهات الاستخدام البيانية هي سيطرة المستخدم (بشرط أن يتحكم النظام، وليس المستخدم، بتكامل النظام وأمنه)

- **البرنامج المساق بالأحداث (events-driven):** نقول عن البرنامج غرضي التوجه الحديث أنه مساق بالأحداث، إذ تستجيب الأغراض لأحداث (رسائل)، وتطلق الأحداث الخارجية التي يولدها المستخدم الاتصالات الداخلية بين الأغراض

- يؤثر "مظهر" الواجهة و"جوها" على تسويق المنتج البرمجي:
  - يؤدي النموذج الأولي هنا دوراً هاماً إذ يخدم الغاية المزدوجة من تقييم جو الواجهة والتحقق من وظائفها
  - بينما يتضح المظهر الحقيقي للواجهة في مرحلة التحقيق

المنارة للاستشارات
www.manaraa.com

توجيهات عامة لتصميم الواجهات

1 – السيطرة للمستخدم

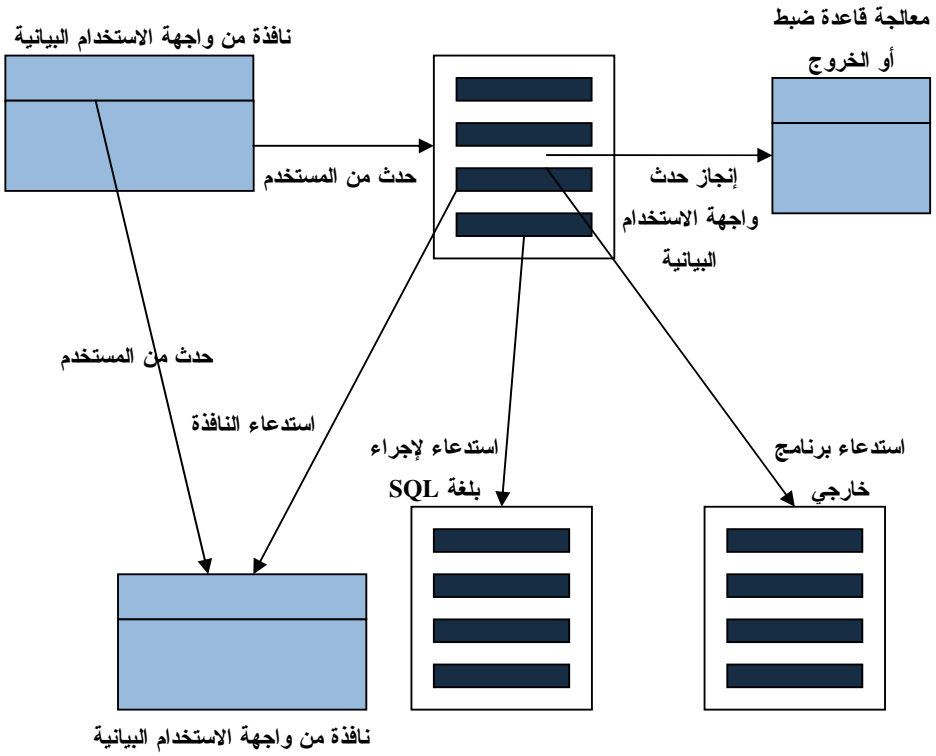
- يشكل مبدأ "سيطرة المستخدم" المبدأ الأهم في تصميم واجهات الاستخدام البيانية، ويدعوه البعض مبدأ "اللا أمومة" – إذ يجب ألا يتصرف البرنامج كما لو أنه (أم المستخدم) يقوم بالأمور عوضاً عنه. ويُفسر هذا المبدأ بأن المستخدم هو من يطلق الأفعال فإذا امتلك البرنامج التحكم يعلم المستخدم بذلك (ساعة رملية أو مؤشر انتظار أو ما شابه).

- يبين الشكل التالي تدفق التحكم النمطي عند التخاطب بين الإنسان والآلة:

  - قد يؤدي حدث يولده المستخدم (انتقاء من قائمة، أو النقر بالفأرة، أو تحريك المؤشر على الشاشة...الخ) إلى فتح نافذة أو استدعاء برنامج –عادة برنامج SQL في تطبيقات نظم المعلومات، فيمتلك البرنامج التحكم مؤقتاً

  - يمكن أن يعيد البرنامج التحكم إلى النافذة نفسها أو إلى نافذة أخرى، كما يمكن أن يستدعي مجتزأ SQL آخر أو إجراءاً خارجياً. قد يقوم البرنامج في بعض الحالات بتنفيذ بعض الأمثال بدلاً من المستخدم. وهذا ما يحدث مثلاً عندما يحتاج البرنامج للقيام بعملية حسابية تقترن بحدث صريح من المستخدم أو عندما ينقل البرنامج المؤشرة إلى حقل آخر من حقول الشاشة مع وجود إجراء معالجة يقترن بحدث الخروج من الحقل الأصلي.

(الشكل 1)



المنارة للاستشارات
www.manaraa.com

توجيهات عامة لتصميم الواجهات

2 – مبدأ الانسجام (Consistency)

- يعتبر مبدأ الانسجام ثاني أهم مبدأ لتصميم الواجهات الجيد، ويعني الانسجام التوافق مع المعايير والطرق المتبعة عادة للقيام بالأفعال

- هناك بعدان على الأقل لمبدأ الانسجام:
  1. التوافق مع المعايير التي يضعها بائع الواجهة
  2. التوافق مع الأسماء والمعايير الأخرى ذات الصلة التي تطورها المؤسسة داخلياً

- للبعدين المذكورين الأهمية نفسها ويجب ألا يتعارض البعد الثاني (الذي قد يؤثر عليه المطورون) مع البعد الأول. فإذا تم تطوير التطبيق ليعمل تحت نظام Windows يجب الالتزام "بمظهر وجو" واجهات "Windows" كما لا يجوز عندما يكون محيط التشغيل المستهدف هو نظام Macintosh استبدال قوائم خيارات النظام بقوائم من شكل مختلف.

- لا يتطلب تصميم الواجهة أن يكون المطور مبدعاً وخلاقاً إلى درجة كبيرة، بل إن ذلك قد يقلل من ثقة المستخدم نفسه ويحد من قدرته على العمل

- كما يجب أن تُعرض البرامج للمستخدمين ضمن بنية مألوفة

- يجب أيضاً عدم التقليل من أهمية التوافق مع المعايير الداخلية الخاصة بالتسميات والرماز والاختصارات، ويتضمن هذا تسمية القوائم وبنودها وترميزها وتسمية الأزرار والحقول كما يتضمن أيضاً المعايير الخاصة بوضع الأغراض على الشاشة والتوافق مع استخدام عناصر الواجهات الأخرى في التطبيقات التي طورتها الشركة سابقاً

توجيهات عامة لتصميم الواجهات

3 – التشخيص والتخصيص

- التشخيص والتخصيص توجهان مترابطان:

  - تشخيص واجهة الاستخدام البيانية: هو تخصيصها للاستخدام الشخصي، نتحدث عن التشخيص عندما يعيد المستخدم ترتيب الأعمدة وحجومها في شبكة عرض مثلاً ويحفظ هذه التعديلات كتفضيلاته الشخصية بحيث تؤخذ بالحسبان عند تشغيل البرنامج في المرة التالية.

  - أما تخصيص واجهة الاستخدام البيانية: فهو مهمة إدارية تقضي بإعداد البرمجيات لمجموعات مختلفة من المستخدمين، ونتحدث عن التخصيص عندما يستطيع البرنامج مثلاً أن يعمل بطرق مختلفة تلائم المستخدمين المبتدئين والمستخدمين المتقدمين، كأن يعرض للمبتدئين نصوص مساعدة صريحة مع رسائل تحذير إضافية عند توليد أحداث يقدّر البرنامج أنها قد تكون خطيرة.

المنارة للاستشارات
www.manaraa.com

- يصعب التمييز بين التشخيص والتخصيص في العديد من الحالات، فتغيير بنود القوائم أو إضافة قوائم جديدة هي أمثلة فيها نظر، فتصنف تشخيصاً عند إجرائها بهدف الاستخدام الشخصي وتصنف تخصيصاً عندما يقوم بها مدير النظام لمجموعة المستخدمين الواسعة.

توجيهات عامة لتصميم الواجهات

4 – التسامح

- يجب أن تتسامح الواجهة الجيدة مع الأخطاء التي قد يرتكبها المستخدمون:
  - إذ يشجع التسامح المستخدم على تحري إمكانيات الواجهة واستكشافها لأنه يعرف أنه إن أخطأ ستعيده الواجهة إلى نقطة البداية عند الضرورة
  - ويفترض هذا التسامح أن توفر الواجهة عدة مستويات من عملية التراجع (undo)

- إلا أنه يصعب تحقيق التسامح في بعض الأحيان كما هو الحال في تطبيقات قواعد المعطيات متعددة المستخدمين، حيث لا يمكن مثلاً للمستخدم الذي يسحب أموالاً من حساب مصرفي أن يتراجع عن هذه العملية، إنما يمكنه أن يصحح المشكلة فقط بإيداع المال من جديد في الحساب وبمناقلة أخرى

توجيهات عامة لتصميم الواجهات

5 – التغذية الراجعة

- يعتبر مبدأ التغذية الراجعة من المبادئ المرافقة للمبدأ الأول (السيطرة للمستخدم)، فامتلاك السيطرة يقتضي أن يعرف المستخدم ماذا يجري عندما يمتلك البرنامج التحكم المؤقت.
  - ويجب أن يضع المطور في النظام مؤشرات مرئية أو ضوئية لكل حدث يولده المستخدم

- طرق تحقيق التغذية الراجعة:
  - تكفي الساعة الرملية أو مؤشر الانتظار في معظم الحالات لإعلام المستخدم أن البرنامج يقوم بأمر ما
  - إلا أنه يفضل استخدام صيغة أوضح للتغذية الراجعة بالنسبة لأجزاء التطبيق التي قد تعاني مشكلات في الأداء (كعرض رسالة توضيحية)

- كما يجب ألا يفترض المطور أن أداء البرنامج سريع بما يكفي لإلغاء دور التغذية الراجعة

المنارة للاستشارات
www.manaraa.com

توجيهات عامة لتصميم الواجهات

6 – النواحي الجمالية وقابلية الاستخدام

- النواحي الجمالية وقابلية الاستخدام:
  - تتعلق النواحي الجمالية بالمظهر المرئي للنظام
  - بينما تتعلق قابلية الاستخدام بسهولة استخدام الواجهة وبساطتها وفعاليتها ووثوقيتها
  - ومن الواضح أن الأمرين يتعلقان برضا المستخدم. وهنا تظهر حاجة مطور الواجهة لمساعدة فنان واختصاصي بعلم الاجتماع وعلم السلوك
- هناك العديد من القواعد الذهبية التي ينصح بإتباعها لتصميم واجهة جميلة قابلة للاستخدام، ومن المواضيع التي تتناولها هذه القواعد:
  - تركيز وحركة العين البشرية
  - استخدام الألوان
  - معنى التوازن والتناظر
  - محاذاة العناصر والفواصل بينها
  - معنى التناسب في الأبعاد
  - تجميع العناصر المترابطة
- من المهم أن نتذكر في هذا السياق قاعدة أساسية وهي أن "البسيط جميل"
  - ويمكن تحقيق البساطة في التطبيقات المعقدة عبر إتباع منهج "فرق تسد" الذي يعني هنا عدم إظهار المعلومات إلا عند الحاجة إليها، وفي نوافذ مستقلة أيضاً

نوافذ الواجهة

1 – مقدمة

- ينطوي تصميم واجهة الاستخدام البيانية على وجهين رئيسين، يرتبطان بالبيئة الأساسية المعتمدة:
  - تصميم النوافذ
  - تصميم عناصر إدخال المعطيات وتنقيحها
- يتألف تطبيق Windows النمطي من:
  1. **النافذة الأولية:** وهي نافذة التطبيق الرئيسية الوحيدة
  2. **النوافذ الثانوية:** هي مجموعة من النوافذ المنبثقة التي تدعم أنشطة المستخدم في النافذة الأولية. ومن أهمها العمليات التقليدية التي تجرى على قاعدة معطيات، أو ما اصطلحنا على اختصاره بالأحرف CRUD (أي إنشاء، قراءة، تحديث، حذف)

المنارة للاستشارات
www.manaraa.com

نوافذ الواجهة

2- النافذة الأولية

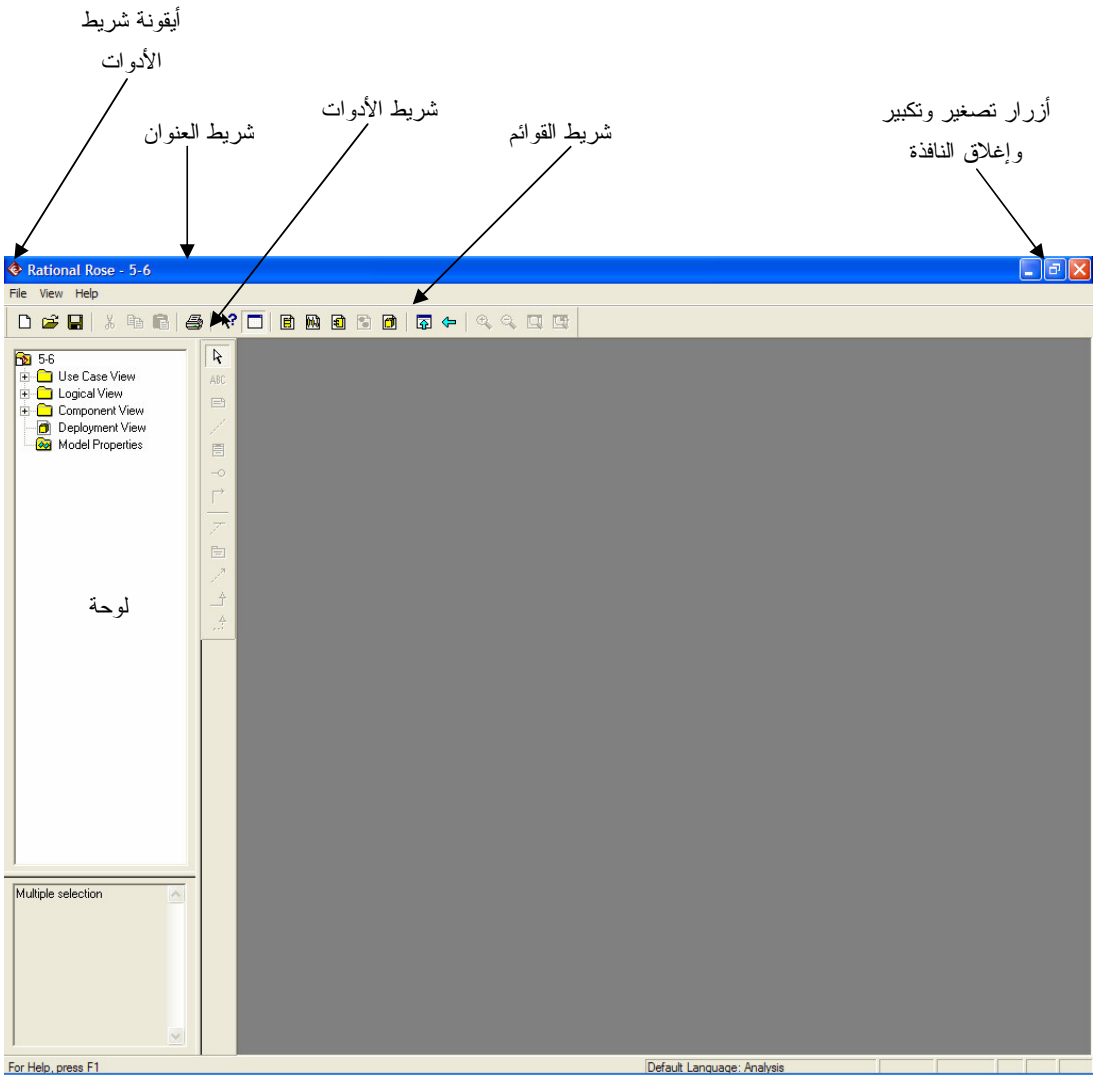
- للنافذة الأولية حدود (إطار) تضم:
  - شريط عنوان النافذة
  - شريط القوائم
  - أشرطة الأدوات
  - شريط الحالة
  - محتويات النافذة القابلة للمعاينة والتعديل
  - كما تضم حدود النافذة الأولية عند الضرورة أشرطة التدرج الأفقية والشاقولية

- يمكن ترتيب محتويات النافذة المعدة للعرض والتعديل في ألواح تسمح بمعاينة ومعالجة معلومات مختلفة لكنها مترابطة بطريقة ما. يبين الشكل التالي نافذة أولية تظهر بعد نجاح الدخول إلى تطبيق، وتعرض اللوحة اليسرى في النافذة خارطة التطبيق بأسلوب كالذي يستخدمه مستكشف Windows (يظهر في الزاوية اليمنى العليا لهذه اللوحة زر إغلاق يسمح للمستخدم بإخفائها عند الرغبة). تذكرك الملاحظات المكتوبة على الشكل بالتسميات الشائعة في نظام Windows.

- تتميز النافذة الأولية عادة بوجود شريط القوائم وشريط الأدوات الذي يحوي أزراراً لأوامر القوائم الأكثر استخداماً بما يسمح للمستخدم بتنفيذ هذه الأوامر بسرعة.

المنارة للاستشارات
www.manaraa.com

(الشكل 2):



www.manaraa.com

نوافذ الواجهة

2- النافذة الأولية

(1) – مستعرض الأسطر

- قد تستخدم النافذة الأولية في تطبيقات نظم المعلومات لعرض سجلات قاعدة معطيات، كسجلات الموظفين، بصيغة سطرية، فتدعى مثل هذه النافذة أحياناً مستعرض الأسطر (row browser)
  - يستطيع المستخدم التجول عبر السجلات صعوداً وهبوطاً بواسطة شريط التدرج الشاقولي أو مفاتيح لوحة المفاتيح (مثل الأسهم أو End, Home, Page Down, Page Up)

- في أية لحظة يكون سطر واحد فقط (سجل) فعالاً في المستعرض:
  - ويؤدي النقر المزدوج فوقه إلى عرض نافذة تنقيح تظهر فيها التفاصيل المتعلقة بذلك السجل، وتسمح هذه النافذة بتعديل محتويات السجل
- يمكن استخدام الألواح لتقسيم النافذة شاقولياً أو أفقياً أو بالصيغتين معاً

نوافذ الواجهة

2- النافذة الأولية

(2) – مستعرض الشجرة

- من الطرق الشائعة الاستخدام لعرض النافذة الأولية مستعرض الشجرة الذي يعرض السجلات المترابطة كعناوين مع إزاحات نسبية، ويحوي كل عنوان عناصر تحكم لتوسيع الشجرة وطيها
  - من الأمثلة المعروفة لمستعرض الشجرة طريقة عرض المجلدات في مستكشف Windows

- وخلافاً لمستعرض الأسطر يسمح مستعرض الشجرة بإجراء التعديلات على محتويات النافذة مباشرة دون فتح نافذة خاصة للتنقيح، وذلك باستخدام عمليات السحب والإفلات

المنارة للاستشارات
www.manaraa.com

نوافذ الواجهة

2- النافذة الأولية

(3) – صفحة الوب

- يمكن التعامل مع صفحة الوب عند استخدامها كنقطة إدخال لتطبيق وب كنوع خاص من النوافذ الأولية:
  - لكن خلافاً لتطبيقات نظم المعلومات النمطية لا يستخدم شريط القوائم أو شريط الأدوات في صفحة الوب لأداء مهام التطبيق، بل لأداء أنشطة التجوال العامة
  - بينما يتم التعامل مع أحداث المستخدم في تطبيقات الوب عادة من خلال الأزرار والارتباطات الفائقة

نوافذ الواجهة

3- النافذة الثانوية

- بغض النظر عن بعض تطبيقات نظم المعلومات البدائية، تعتبر النافذة الثانوية مكملة للنافذة الأولية المرتبطة بها، فهي تمدد وظيفتها، وعلى وجه الخصوص بالنسبة للعمليات التي تغير قاعدة المعطيات (أي عمليات الإضافة insert والحذف delete والتحديث update).

- تكون النافذة الثانوية عادة مُلزِمة (modal) بالنسبة للنافذة الأولية، أي يجب أن يجيب المستخدم على طلب أو سؤال النافذة الثانوية ويغلقها قبل أن يتخاطب مع أي نافذة أخرى في التطبيق. ويمكن من حيث المبدأ ألا تكون النوافذ الثانوية ملزمة لكن لا ينصح بذلك.

- من أمثلة النوافذ الثانوية البسيطة نافذة الدخول إلى تطبيق ما (logon window):
  - توضح نافذة الدخول الفوارق البصرية الأساسية بين النافذة الأولية والنافذة الثانوية:
    - إذ لا تحوي النافذة الثانوية أية أشرطة (شريط قوائم، شريط أدوات، أشرطة تدرج، شريط الحالة)
    - ويتم التقاط الأحداث التي يولدها المستخدم بواسطة أزرار الأوامر مثل موافق (OK) وإلغاء (Cancel) والمساعدة (Help)

- للنوافذ الثانوية صيغ وأشكال مختلفة فقد تكون:
  1. علبة حوار (dialog box)
  2. مجلد بعلامات جدولة (tab folder)
  3. قائمة منسدلة (drop-down list)
  4. علبة رسالة (message box)

المنارة للاستشارات
www.manaraa.com

نوافذ الواجهة
3- النافذة الثانوية
(1) – علبة الحوار

- تعتبر علبة الحوار غالباً مرادفة لمفهوم النافذة الثانوية، فهي تعكس معظم الخصائص المطلوبة من النافذة الثانوية:
  - حيث تدعم الحوار بين المستخدم والتطبيق، ويقتضي هذا الحوار أن يدخل المستخدم بعض المعلومات التي تهم التطبيق

نوافذ الواجهة
3- النافذة الثانوية
(2) – مجلد علامات الجدولة

- يعتبر مجلد علامات الجدولة مفيداً عندما تتجاوز كمية المعلومات الواجب عرضها في النافذة الثانوية مساحة هذه النافذة وبحيث يمكن تقسيم هذه المعلومات إلى مجموعات منطقية
  - وتظهر في أي لحظة معلومات صفحة واحدة فقط من الصفحات المجدولة

نوافذ الواجهة
3- النافذة الثانوية
(3) – القائمة المنسدلة

- قد يكون مناسباً في بعض الحالات استبدال الصفحات المجدولة بقائمة منسدلة (أو مجموعة من القوائم المنسدلة):
  - تعطي القائمة المنسدلة مجموعة من الخيارات يستطيع المستخدم أن ينتقي واحداً منها.
  - ويمكن أن تسمح القائمة المنسدلة للمستخدم بإضافة قيمة جديدة إلى مجموعة الخيارات بحيث تظهر عند فتح القائمة في المرة التالية.
- قد لا تقتصر القائمة المنسدلة على قائمة بسيطة من القيم، بل قد تكون القيم مرتبة في متصفح بصيغة شجرة.

نوافذ الواجهة
3- النافذة الثانوية
(4) – علبة الرسالة

- **علبة الرسالة:** هي نافذة ثانوية تعرض رسالة للمستخدم قد تكون تحذيراً أو توضيحاً أو شرطاً استثنائياً أو غيره.
  - تعطي أزرار الأوامر في علبة الرسالة للمستخدم خياراً أو أكثر للإجابة.

المنارة للاستشارات
www.manaraa.com

الترابط بين النوافذ

- يظهر التطبيق بالنسبة للمستخدم كمجموعة من النوافذ المتعاونة:
  - وتقع على عاتق مصمم واجهة الاستخدام البيانية مهمة تنظيم الارتباطات بين النوافذ في بنية منسجمة وسهلة الفهم
  - إذ يجب ألا يشعر المستخدم أبداً أنه تائه بين مجموعة نوافذ مفتوحة أمامه

- يجب أن يكون الارتباط بين النافذة الأولية والنافذة الثانوية الأعلى المفتوحة حالياً عبارة عن مسار وليس هرمياً، ويمكن تحقيق هذا الأمر بجعل النافذة الثانوية ملزمة بالنسبة للنافذة الأسبق

- يجب أن يسهل تصميم الواجهة البيانية حركة المستخدم وتجواله بين النوافذ، لكن يبقى لبنية شريط القوائم الدور الأهم في توضيح إمكانيات التطبيق للمستخدم. إذ تتضح الارتباطات بين النوافذ، وإن بصيغة غير مباشرة، عبر الأوامر التي تقدم للمستخدم وكيفية تنظيمها في شريط القوائم

الترابط بين النوافذ

1 - الوثيقة والمعاينة

- يعتمد تصميم واجهات الاستخدام البيانية في بيئة Microsoft Windows على مكتبة الصفوف المستخدمة لتحقيق أغراض النوافذ وعناصر التحكم المستخدمة فيها، أي على واجهة برمجة التطبيقات API. وتدعى هذه المكتبة بمكتبة MFC (Microsoft Foundation Classes).

- تتطلب البرمجة في بيئة Windows:
  - إنشاء أغراض أمثال من صفوف المكتبة MFC
  - إنشاء صفوف جديدة خاصة بالتطبيق ترث بعض الوظائف العامة من صفوف المكتبة MFC
  - قبول بنية محددة للترابط والتفاعل بين النوافذ، وتعرف هذه البنية باسم منهج الوثيقة/المعاينة

- الوثيقة وفقاً لمفهوم المكتبة MFC هي مجموعة من معطيات التطبيق التي يستطيع المستخدم أن يتفاعل معها:
  - وتحوي الوثيقة أي نمط من المعطيات وليس نصوصاً فقط
  - كما يُشتق غرض الوثيقة في المكتبة MFC من الصف CDocument.

- يمكن إعادة عرض جزء فقط من المعطيات المخزنة في الغرض CDocument على الشاشة، ويدعى هذا الجزء المعاينة (View) وهو مشتق من الصف CView:
  - ويمكن أن نجد عدة معاينات للوثيقة نفسها
  - أما من الناحية التقنية فيكون الغرض CView مستقلاً عن النافذة (الإطار) التي يُعرض ضمنها

المنارة للاستشارات
www.manaraa.com

الترابط بين النوافذ

2 – الواجهة وحيدة الوثيقة

- **الواجهة وحيدة الوثيقة:** يمكن أن تتألف واجهة الاستخدام البيانية في بعض التطبيقات البسيطة من نافذة أولية وحيدة تُفتح من خلالها وثيقة واحدة فقط في لحظة محددة.
  - تدعم المكتبة MFC هذا النوع من الواجهات تحت الاسم SDI (Single Document Interface)

الترابط بين النوافذ

3 – الواجهة متعددة الوثائق

- تحتاج التطبيقات الأكثر تعقيداً إلى فتح العديد من الوثائق في الوقت نفسه. وقد تكون هذه الوثائق من النمط نفسه لكنها غالباً من أنماط مختلفة
  - تدعم المكتبة MFC هذه التطبيقات بالاسم MDI (Multiple Document Interface)

- التطبيق ذو الواجهة متعددة الوثائق:
  - يستخدم التطبيق ذو الواجهة متعددة الوثائق نافذة أولية واحدة فقط، وتدعى النافذة الأب
  - ويسمح هذا التطبيق بفتح عدة وثائق ضمن إطار هذه النافذة، ويشار إلى كل منها باسم النافذة الابن. وتسلك كل من هذه النوافذ كما لو أنها نافذة أولية يمكنها الظهور ضمن النافذة الأب فقط (وليس على سطح المكتب)

- تتجلى حقيقة استخدام الواجهة MDI لنافذة أولية واحدة فقط من خلال شريط القوائم الوحيد الذي تتشارك على استخدامه كل النوافذ الأبناء، التي تتشارك أيضاً في استخدام شريط الأدوات وشريط الحالة، مع أنه يمكن تعديل الأوامر التي يقدمها شريط القوائم وشريط الأدوات لتعكس الوظائف المناسبة للنافذة الابن النشطة حالياً

التجوال عبر النوافذ

- **لا يعطي الوصف البياني لنوافذ واجهات الاستخدام معلومات عن كيفية التجوال الفعلي بين النوافذ**
  - يجب علينا أن نصمم نظام التجوال بين النوافذ عبر مخطط يبين نوافذ التطبيق وأغراض التحكم التي تسمح للمستخدم بالانتقال من نافذة إلى أخرى

- **لا تتضمن لغة UML تقنية بيانية لنمذجة التجوال بين النوافذ:**
  - تصميم نموذج خاص لهذه الغاية
  - الإفادة من الأنماط المحيطية في UML وتخصيص أحد مخططاتها بحيث يسمح بتمثيل آلية التجوال بين النوافذ

المنارة للاستشارات
www.manaraa.com

التجوال عبر النوافذ

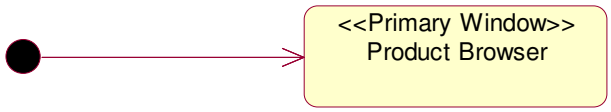
1 – تنميط مخطط النشاط بالأنماط المحيطية الملائمة لتمثيل التجوال بين النوافذ

- يشكل مخطط النشاط في UML أداة جيدة لتمثيل التجوال عبر النوافذ:
  - يظهر مخطط النشاط الانتقالات عادة بين الأنشطة المختلفة
  - كما يمكن استخدامه لتمثيل حالات الأغراض لكونه شكل من أشكال مخططات الحالات
  - يمكننا استخدام هذه الثنوية لمحاكة الثنوية التي تتطلبها أغراض واجهات الاستخدام البيانية. إذ تشبه نافذة واجهة الاستخدام حالة تنتظر وقوع أحداث (أنشطة)

- يمكن تنميط الحالات والأنشطة في مخطط النشاط:
  - إذ تستطيع الأنماط المحيطية تمييز أنواع مختلفة من النوافذ ومن الأغراض الأخرى التي تدوم بين الأحداث –أي تلك التي تمتد فترة وجودها على فترة زمنية طويلة نسبياً
    - يبين الشكل التالي (الشكل 4) حالة تم تمييزها كنافذة أولية، وتمثل هذه الحالة مستعرضاً للمنتجات (شبكة) يُعرض في نافذة التطبيق الأولية، وهي أيضاً الحالة البدائية في النموذج
  - يمكن لأنماط الأنشطة المحيطية كذلك أن تميز أنواعاً مختلفة من عناصر تحكم الواجهات التي يمكن استخدامها لإطلاق أحداث تؤثر على الحالات. وخلافاً للحالات فإن فترة استمرار النشاط قصيرة جداً – بل يمكن القول أنها لحظية نسبياً.

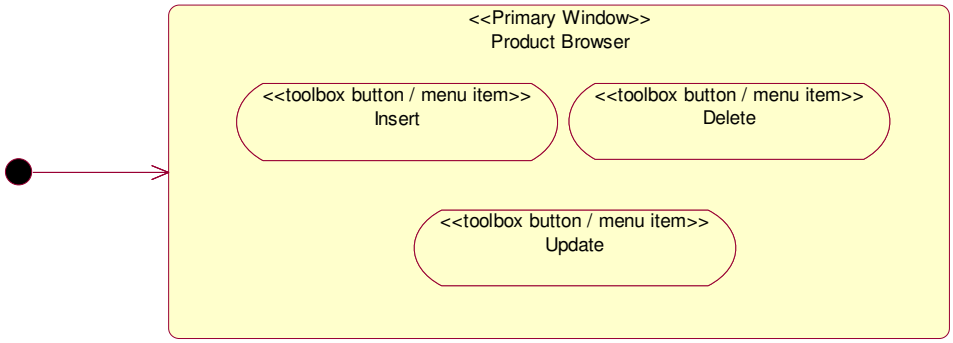
- يمكن رسم الأنشطة (الأشكال البيضوية) ضمن الحالة (المستطيل مستدير الزوايا) التي تؤثر عليها،
  - ويبين الشكل التالي (الشكل 5) ثلاثة أنشطة ضمن الحالة Product Browser، وهي نافذة يمكن استخدامها لبدء حدث إضافة منتج جديد أو حذف منتج موجود أو تعديله. ويمكن إطلاق أي من الأحداث الثلاثة بواسطة زر أمر في علبة أدوات أو بنود قائمة أوامر. كما يدعم النقر المزدوج على سطر المنتج في المستعرض حدث تحديث معلومات المنتج.

(الشكل 4):



المنارة للاستشارات
www.manaraa.com

(الشكل 5):



التجوال عبر النوافذ

1 – تنميط مخطط النشاط بالأنماط المحيطية الملائمة لتمثيل التجوال بين النوافذ (تتمة)

- تعتمد قائمة الأنماط المحيطية الكاملة للحالات والأنشطة اللازمة لتصميم نظام التجوال بين النوافذ على بنية الواجهة البيانية المختارة للتطبيق.
- نستعرض فيما يلي قائمة جزئية من الأنماط المحيطية التي يمكن استخدامها مع واجهات نظام Microsoft Windows:
  - الحالات (النوافذ)
  - نافذة أولية
    - لوحة في نافذة أولية
    - مستعرض أسطر
    - مستعرض شجرة
    - صفحة وب
  - نافذة ثانوية
    - علبة حوار
    - علبة رسالة
    - مجلد علامات جدولة
  - معطيات النافذة
    - علبة نص (text box)
    - علبة انتقاء (combo box)
    - علبة إزادة/إنقاص (spin box)
    - عمود
    - سطر

المنارة للاستشارات
www.manaraa.com

- مجموعة حقول
- الأنشطة (عناصر تحكم النافذة)
  - بند قائمة منسدلة
  - بند قائمة منبثقة
  - زر في شريط أدوات
  - زر أمر
  - نقر مزدوج
  - قائمة انتقاء
  - مفتاح على لوحة المفاتيح
  - مفتاح وظيفي على لوحة المفاتيح
  - مفتاح مسرع على لوحة المفاتيح
  - زر تدرج
  - زر إغلاق النافذة

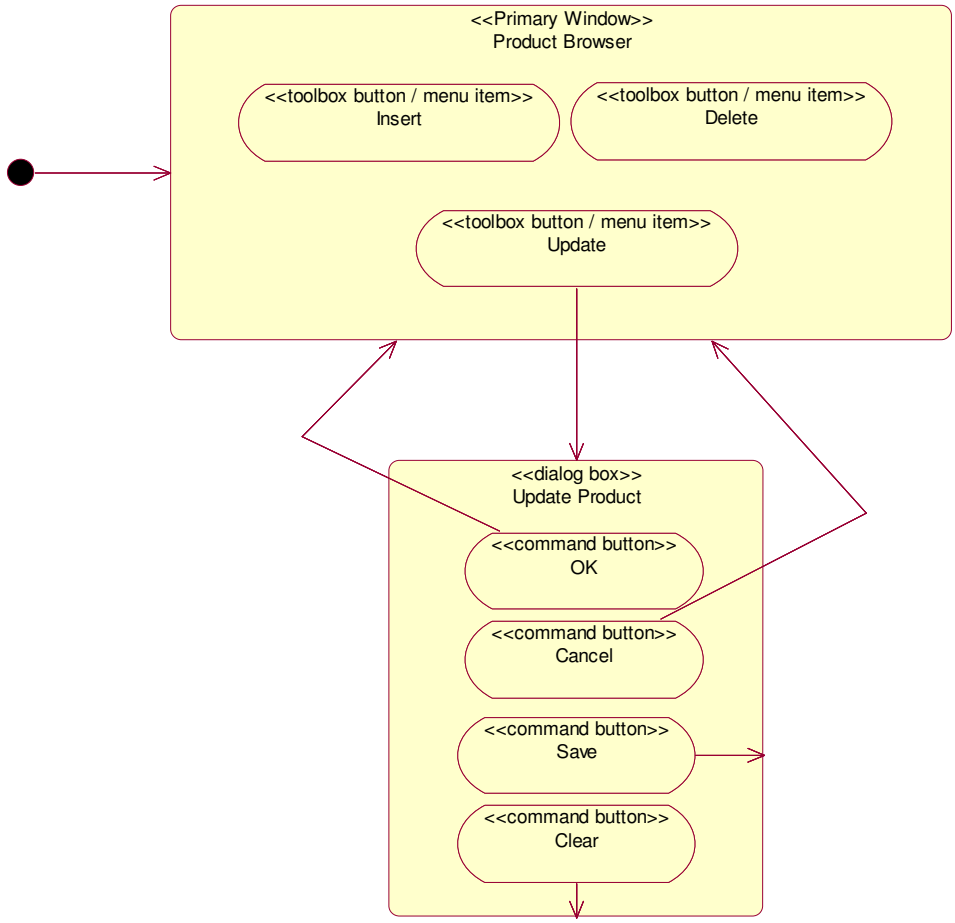
التجوال عبر النوافذ

2 – مخطط التجوال عبر النوافذ

- **مخطط التجوال عبر النوافذ:** بعد تحديد الأنماط المحيطية للحالات والأنشطة يمكننا استخدام خطوط الانتقال للوصل بينها، فنحصل بذلك على مخطط التجوال عبر النوافذ الذي تسبب فيه الأنشطة انتقالات بين الحالات.

- **مثال:** يعرض الشكل التالي الحالات الناتجة عن تفعيل الأنشطة الموجودة في النافذة Product Browser. كما جرى أيضاً توسيع الحالة التي ينقل إليها الحدث Update، وتحوي النافذة Update Product («dialog box») أربعة أنشطة («command buttons»)، ويؤدي الضغط على الزر OK أو على الزر Cancel للانتقال من جديد إلى النافذة Product Browser. ولا يؤدي الضغط على الزر Save أو على الزر Clear إلى تغيير النافذة النشطة (إذا كنت تريد تصوير تغير الحالة ضمن النافذة النشطة عليك أن توسع النموذج بأنماط محيطية إضافية).
(الشكل 6):

المنارة للاستشارات
www.manaraa.com



مثال – التسويق الهاتفي

- **نص المسألة:**

تسعى إحدى الجمعيات الخيرية إلى زيادة رصيدها من خلال بيع بطاقات اليانصيب في حملات خيرية. وتحتفظ الجمعية بقائمة بأسماء من شاركوا سابقاً، بحيث تنتقى مجموعة جزئية من هذه الأسماء عند بدء حملة جديدة لبيعهم البطاقات عبر الهاتف أو الاتصال بهم مباشرة عبر البريد.

تتبع الجمعية أساليب خلاقة لكسب مساهمين جدد، بما في ذلك منح نقاط إضافية للمساهمين الذين يشترون مجموعة بطاقات. ولا تنتقي الجمعية الزبائن المحتملين عشوائياً باستخدام أدلة الهاتف والوسائل المشابهة.

ولدعم عملها قررت الجمعية التعاقد على تطوير تطبيق تسويق هاتفي جديد بحيث يدعم النظام الجديد حوالي خمسين مسوقاً يعملون معاً في الوقت نفسه، كما يجب أن يجدول النظام المكالمات الهاتفية تبعاً لأفضليات محددة مسبقاً ولشروط أخرى محددة.

يجب أن يستطيع النظام طلب الاتصالات المجدولة، وعند فشل الاتصال عليه أن يعيد جدولته بحيث يحاول إجراءه من جديد لاحقاً.

كما يجب أن يسمح النظام بتسجيل نتائج المكالمات بما فيها طلبات شراء البطاقات وأية تغيرات أخرى تخص المساهمين.

المنارة للاستشارات
www.manaraa.com

- **المطلوب:** صمم مخطط التجوال بين النوافذ آخذاً بعين الاعتبار التفاصيل الإضافية الواردة أدناه:
  - تظهر نافذة واجهة التطبيق الرئيسية كنافذة فارغة تحوي فقط عنوان التطبيق وزرين، يسمح أحدهما Next Call للمسوق بإجراء الاتصال التالي فيما يسمح الآخر Quit بإنهاء التطبيق. يؤدي الضغط على الزر Next Call إلى عرض رسالة ترحيب ثم تملأ النافذة بمعلومات عن الاتصال الذي يجري حالياً.
  - يمكن للمسوّق، خلال حديثه مع المساهم أن يتخاطب مع النظام من خلال أزرار أوامر مختلفة، وهناك عدة أزرار في شريط الأدوات تسمح بتسجيل خلاصة المكالمة (مثل طلب شراء بطاقات، إعادة جدولة المكالمة، فشل المكالمة). هناك أيضاً مجموعة أزرار تسمح بمعاينة معلومات تفصيلية إضافية عن الحملة أو عن المساهم.

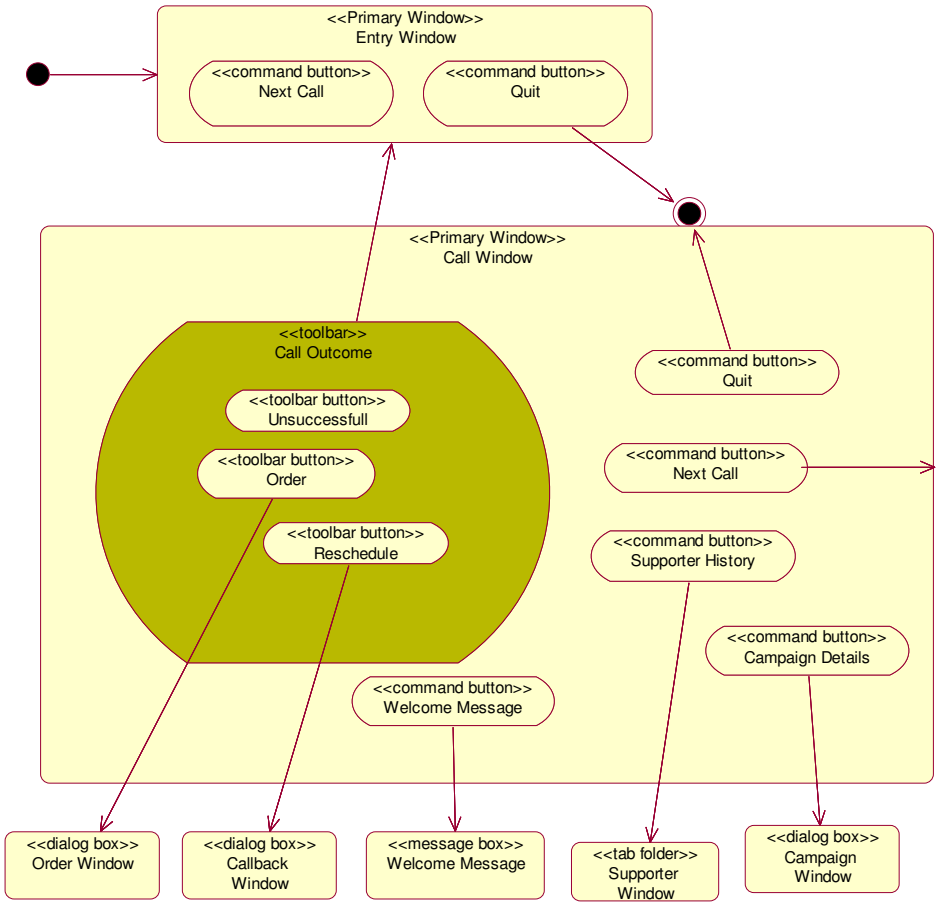
مثال – التسويق الهاتفي

الحل

- **الحل:** يبين الشكل التالي (الشكل 7) مخطط التجوال عبر النوافذ لهذا المثال، ومع أن النافذتين Call Window, Entry Window هما في الواقع نافذة واحدة «primary window» لكننا نمثلهما صراحة كحالتين للنافذة. تكون الحالة Entry Window نشطة بعد بدء تشغيل التطبيق وعندما تكون خلاصة المكالمة هي الفشل (Unsuccessful). ومن المحتمل أن تعيد أحداث النوافذ الثانوية التحكم إلى النافذة Entry Window لكن النموذج لا يعكس هذه الحقيقة، ويؤدي النقر على الزر Quit في كل من الحالتين إلى إنهاء التطبيق.

(الشكل 7):

المنارة للاستشارات
www.manaraa.com



مثال – إدارة العلاقات

- **نص المسألة:**

تعتمد إحدى شركات أبحاث التسويق على قاعدة من الزبائن هم في الواقع مؤسسات تشتري منها تقارير تحوي خلاصة تحليل الأسواق. ويشتري بعض كبار الزبائن برمجيات تخصصية من الشركة، فتزودهم الشركة بمعلومات خام غير مصاغة بحيث يتمكن أولئك الزبائن من استخدامها لتوليد تقارير خاصة بهم.

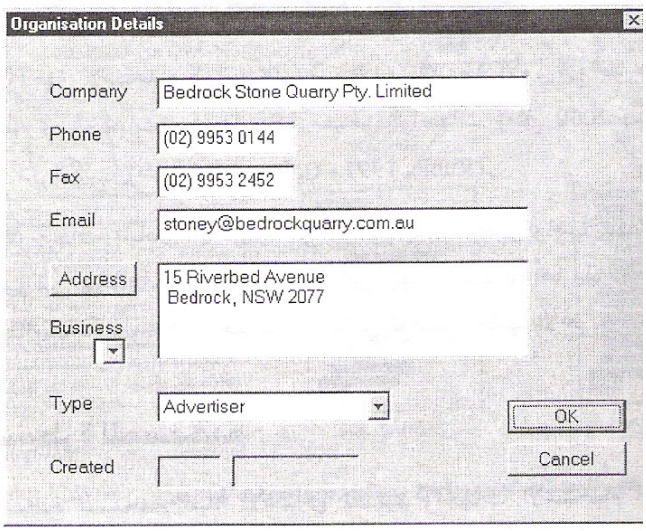
تسعى هذه الشركة دوماً لكسب زبائن جدد حتى لو لم يهتموا إلا بنمط واحد من التقارير. وحيث أن أولئك الزبائن هم زبائن محتملون ولم يصبحوا بعد زبائن فعليين تفضل الشركة أو تسميهم "اتصالات" (Contacts).

تسعى الشركة إلى تطوير نظام إدارة علاقات جديد يمكن لكل موظفي الشركة استخدامه لكن بمستويات مختلفة من حقوق الوصول. يجب أن يسمح النظام بجدولة وإعادة جدولة الأنشطة بحيث يتمكن الموظفون من التعاون فيما بينهم لكسب زبائن جدد وتعزيز العلاقات القائمة حالياً.

المنارة للاستشارات
www.manaraa.com

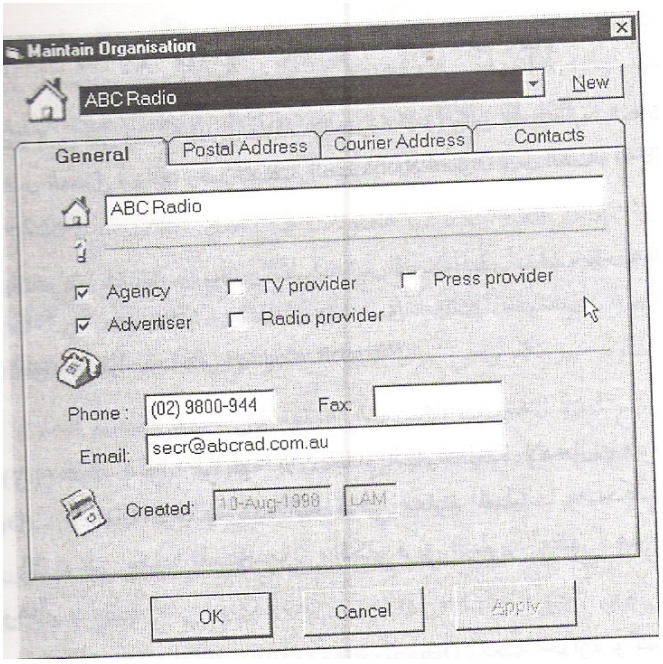
- **المطلوب:**
  - يبين الشكل التالي (الشكل 8) نموذجاً أولياً لواجهة الاستخدام البيانية المعدة لإدخال معلومات المؤسسة (Organization). إن الغاية الرئيسية لهذه النافذة هي معاينة المعطيات والتحكم بالأغراض في النافذة. ستغير الآراء التي يبديها المستخدمون وأفكار فريق التطوير "مظهر" هذه النافذة، وربما تغير أيضاً "جوها".
  - يبين الشكل التالي (الشكل 9) التحقيق الممكن لنافذة Organization كعلبة حوار. وكما تلاحظ فقد اختار المبرمج استخدام الصفحات المجدولة وعدداً من التعديلات الأخرى للانسجام مع مبادئ تصميم واجهات الاستخدام البيانية في Microsoft Windows.
  - لا تسمح النافذة الأولية للتطبيق (الشكل 10) بإجراء بعض العمليات على الأحداث. فإدخال حدث جديد مثلاً أو تعديل حدث موجود يتم عبر نافذة ثانوية –علبة حوار.
  - يجب أن تظهر عند النقر المزدوج فوق حدث ما في النافذة الأولية علبة حوار تعرض كل التفاصيل المتعلقة بذات الحدث. كما تعرض أيضاً معلومات عن المهمة التي تضم الحدث بالإضافة إلى معلومات عن الشركة التي يرتبط بها الحدث.
  - تتضمن تفاصيل الحدث التي يمكن عرضها وربما تعديلها نوع الحدث (الحقل action)، ووصفه (الحقل notes)، وتاريخ ووقت إنشاء الحدث المستخدم الذي أنشأه، ووقت جدولة الحدث ووقت استحقاقه ووقت إنجازه.
  - **المطلوب في هذا المثال تصميم علبة الحوار التي تسمح بالتعامل مع الحدث بما يتناسب مع المتطلبات المذكورة أعلاه.**

(الشكل 8):

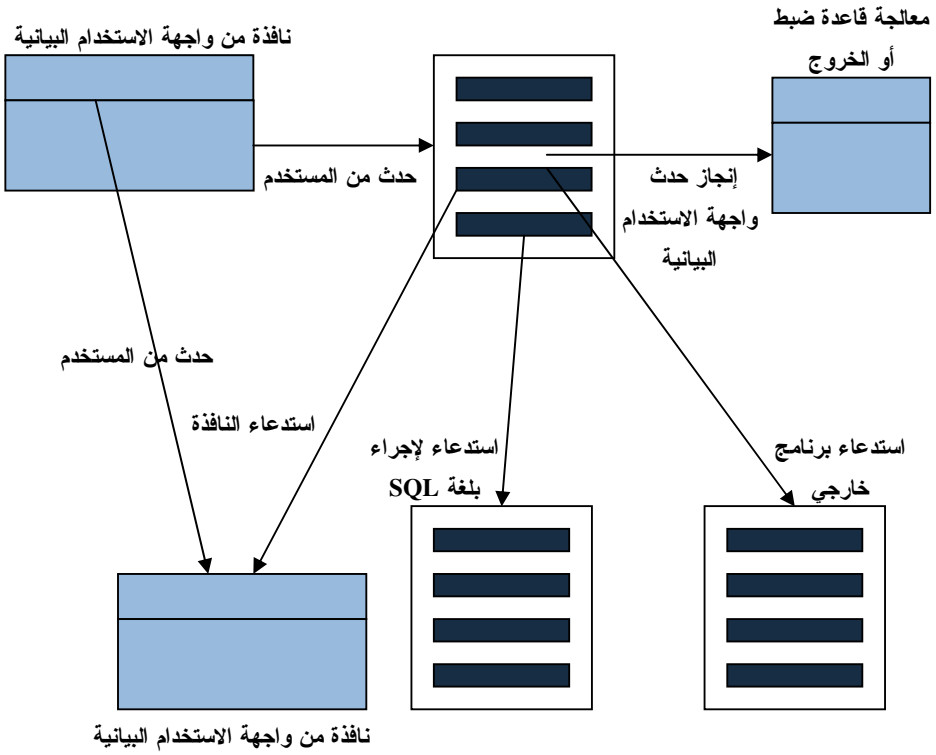


(الشكل 9):

المنارة للاستشارات
www.manaraa.com



(الشكل 10):



www.manaraa.com

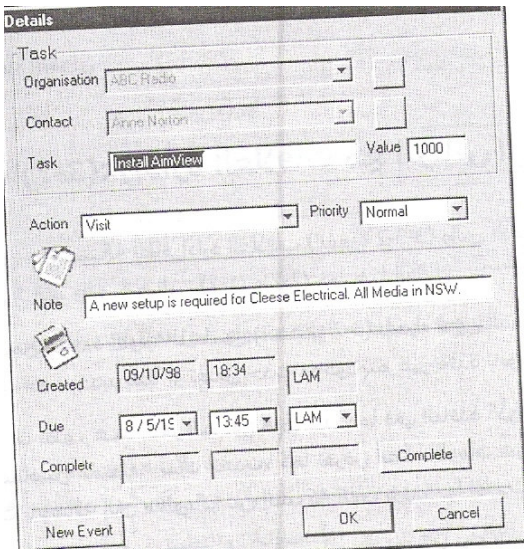
مثال – إدارة العلاقات

الحل

- **الحل:**

يبين الشكل التالي (الشكل 11) الحل المقترح لهذا المثال. لاحظ أن الحقلين Contact, Organization غير قابلين للتنقيح لأنه لا يمكن تعديل الحدث. كذلك الأمر أيضاً بالنسبة لقيم الحقول الخاصة بتاريخ ووقت إنشاء الحدث (Created).

(الشكل 11):

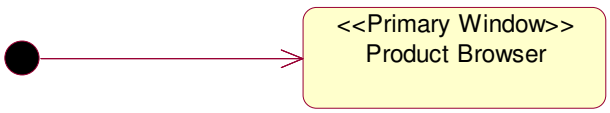


مثال – إدارة العلاقات

طلبات إضافية

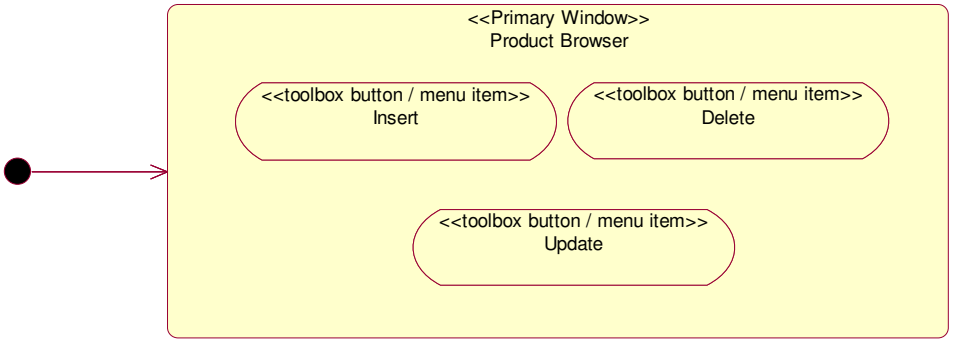
- **طلبات إضافية:**
  - خذ النافذة الأولية في (الشكل 12) وعلبة الحوار في (الشكل 13) وأنشئ مخطط التجوال لنوافذ هذين الشكلين. يجب أن يميز المخطط ويمثل أحداث المستخدم الرئيسية على النافذة الأولية وعلى علبة الحوار. كما يجب أن يأخذ بالحسبان أيضاً استخدام التقويم.
  - يجب أن يتضمن مخطط التجوال ثلاث نوافذ: النافذة الأولية Contact Management («primary windows») وعلبة الحوار Task/Event Details («dialog box») وعلبة انتقاء التقويم Calendar («combo box»).

(الشكل 12):



www.manaraa.com

(الشكل 13):



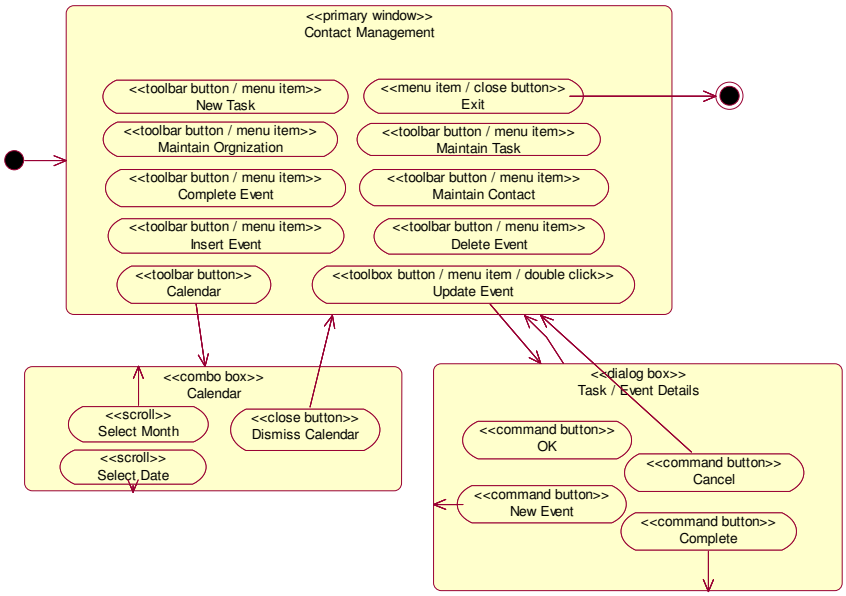
مثال – إدارة العلاقات

حل الطلبات الإضافية

- **حل الطلبات الإضافية:**
  - يبين الشكل التالي (الشكل 14) مخطط التجوال عبر النوافذ المقترح كحل لهذا المثال، ويؤدي إطلاق الحدث («toolbar button») Calendar إلى عرض عنصر التقويم Calendar («combo box»)، ويؤدي النقر على زر إغلاق النافذة إلى إخفائها، ولا تختفي علبة الانتقاء عند انتقاء الشهر («scroll») أو التاريخ («select»).
  - يتم الوصول إلى علبة الحوار Task/Event Details («dialog box») عبر الحدث Update Event («toolbar button/menu item/double click»)، ويؤدي الضغط على زر الأمر OK أو على زر Cancel إلى إحصاء علبة الحوار وعودة التحكم إلى النافذة الأولية، ويؤدي تنشيط زر الأمر Complete إلى ملء حقول النافذة مع بقاء التحكم في علبة الحوار إذ قد يرغب المستخدم بإنشاء حدث جديد، ويؤدي النقر على الزر New Event إلى مسح محتويات الحقول مما يسمح للمستخدم بإدخال تفاصيل حدث جديد.

(الشكل 14):

المنارة للاستشارات
www.manaraa.com



www.manaraa.com

القسم الخامس عشر

الاختبارات

الكلمات المفتاحية:

الاختبارات، إدارة التغييرات، المراجعة العامة، التدقيق، الاختبار إزاء التوصيف (اختبار الصندوق الأسود)، الاختبار إزاء الرماز (اختبار الصندوق الأبيض)، اختبار قيود النظام، اختبار واجهة الاستخدام، اختبار قاعدة المعطيات، اختبار الترخيص، اختبار تجاوز الفشل، اختبار التشكيل، اختبار التثبيت، اختبار الإجهاد.

ملخص:

يُركِّز هذا الفصل على المفاهيم والطرق الأساسية للاختبارات.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- التعرف على اختبارات النظام
- المراجعة العامة
- التدقيق
- الاختبار إزاء التوصيف
- الاختبار إزاء الرماز
- اختبار قيود النظام
- إدارة التغييرات

وثائق الاختبارات وإدارة التغييرات

المنارة للاستشارات
www.manaraa.com

مقدمة

- لا يقصد بالاختبارات مجرد البحث عن أخطاء البرامج، بل يجب اختبار مخرجات كل مرحلة من مراحل دورة التطوير.
- لا يقتصر تطبيق إدارة التغيرات على التحسينات التي يطلب الزبائن إدخالها أو على النواقص التي تُكتشَف أثناء الاختبارات، بل تُعتبر حجر أساسي لإدارة المشروع ككل.
- يجب توثيق طلبات التعديل كما يجب تتبع آثار كل تغيير على مخرجات التطوير وإعادة اختبارها بعد إنجاز التعديلات.
- الهدف الأساسي لقابلية التتبع هو تمكين المطور من توليد توثيق كامل للنظام صحيح ومنسجم عبر مختلف النماذج والوثائق الخاصة بالنظام.

ليست الاختبارات وإدارة التغيرات مراحل مستقلة في دورة حياة المنتج البرمجي، بل هي أنشطة تمتد لتغطي كامل دورة الحياة هذه. ولا يقصد بالاختبارات مجرد البحث عن أخطاء البرامج، بل يجب اختبار مخرجات كل مرحلة من مراحل دورة التطوير. بالمثل لا يقتصر تطبيق إدارة التغيرات على التحسينات التي يطلب الزبائن إدخالها أو على النواقص التي تكتشف أثناء الاختبارات. بل تعتبر حجر أساسي لإدارة المشروع ككل.

يجب توثيق طلبات التعديل كما يجب تتبع آثار كل تغيير على مخرجات التطوير وإعادة اختبارها بعد إنجاز التعديلات.
تعتبر قابلية التتبع الأساس الذي تستند إليه الاختبارات وإدارة التغيرات. فهي تصور وترتبط وتتعقب كل حقائق التطوير الهامة، بما فيها المتطلبات، والهدف الأساسي منها هو تمكين المطور من توليد توثيق كامل للنظام صحيح ومنسجم عبر مختلف النماذج والوثائق، من المتطلبات إلى التوثيق التقني وتوثيق الاستخدام.

اختبار خدمات النظام

- نوعان رئيسان من الاختبارات المنهجية:
- اختبارات غير معتمدة على التنفيذ (مراجعات صورية): المراجعة العامة، التدقيق
- اختبارات معتمدة على التنفيذ: الاختبار إزاء التوصيف، الاختبار إزاء الرماز

يميز Schach (1996) بين اختبار خدمات النظام غير الصوري وبين الاختبار المنهجي. حيث يُجري أي مطور اختباراً غير صوري عند نمذجة أو تحقيق خدمة من خدمات النظام. لكن يبقى هذا الاختبار غير كامل بطبيعته، فالشخص الذي يُطوِّر الخدمة هو آخر من يمكن أن يكتشف أخطاءها.
للاختبار غير الصوري معنى عام ويجب إتمامه باختبار منهجي، وهناك نوعان رئيسان من الاختبارات المنهجية:
– اختبارات غير معتمدة على التنفيذ (مراجعات صورية): المراجعة العامة، التدقيق
– اختبارات معتمدة على التنفيذ: الاختبار إزاء التوصيف، الاختبار إزاء الرماز

المراجعة العامة

- المراجعة العامة هي شكل من أشكال مراجعة الأفكار الصورية والتي يمكن إجراؤها في أي مرحلة من مراحل التطوير

المنارة للاستشارات
www.manaraa.com

- يُسلَّم المشاركون في اجتماع المراجعة كل المواد التي تجب مراجعتها قبل موعد الاجتماع بأيام قليلة، وذلك ليقوموا بدراستها وتزويد المنسق بملاحظاتهم عليها قبل الاجتماع
- هدف اجتماع المراجعة هو تحديد المشكلة دون الاهتمام بطريقة حلها
- تُضفي عملية المراجعة دقة ومهارة على إجرائية التطوير، كما تساهم في تحسين الإنتاجية والالتزام بمواعيد المشروع، وينتج عنها معلومات هامة جداً يمكن الاستفادة منها في تحسين جودة البرمجيات

المراجعة العامة هي شكل من أشكال مراجعة الأفكار الصورية والتي يمكن إجراؤها في أي مرحلة من مراحل التطوير، وهي عبارة عن لقاء ودي بين المطورين له غايات واضحة ومخطط له بعناية مع جدول أعمال ومدة محددة. وتُجرِي معظم فرق تطوير نظم المعلومات مراجعات من هذا النوع بمعدل مرة كل أسبوع.

يُسلَّم المشاركون في اجتماع المراجعة كل المواد التي تجب مراجعتها (النماذج، الوثائق، رماز البرنامج، ...) قبل موعد الاجتماع بأيام قليلة. حيث يَجمع هذه المواد ويُوزعها على المشاركين منسق المراجعة، ليقوم المشاركون بدراستها وتزويد المنسق بملاحظاتهم قبل الاجتماع أيضاً. يكون الاجتماع قصيراً نسبياً (من ساعتين إلى ثلاث ساعات على الأكثر)، فيقدم المنسق خلال الاجتماع الملاحظات التي تلقّاها لتناقش بنداً بنداً.

إن الهدف من الاجتماع هو تحديد المشكلة وليس معرفة المطور المسؤول عنها أو محاولة إيجاد الحل المناسب لها.

هناك الكثير من الأدلة التي تثبت جدوى المراجعة العامة، فهي تضفي دقة ومهارة على إجرائية التطوير، وتساهم في تحسين الإنتاجية والالتزام بمواعيد المشروع، كما ينتج عنها معلومات هامة جداً يمكن الاستفادة منها في تحسين جودة البرمجيات.

التدقيق

- التدقيق هو اجتماع ودي يَجري بإشراف مباشر من إدارة المشروع، ويهدف إلى تحديد العيوب والنواقص، ثم جدولة من يجب أن يقدم الحلول ومتى
- تَجري اجتماعات التدقيق بتواتر أقل من اجتماعات المراجعة العامة، وقد تهتم بدراسة مواضيع محددة وحرجة، وهي أكثر صورية ودقة
- قد يُعقَد اجتماع تمهيدي قبل جلسة التدقيق يُقدِم فيه مطور المنتج الموضوع العام، وتُسلَّم المواد التي يجب تدقيقها إلى المشاركين خلال الاجتماع التمهيدي أو قبله

التدقيق هو اجتماع ودي أيضاً، كالمراجعة العامة، لكنه يجري بإشراف مباشر من إدارة المشروع. وهو يهدف أيضاً إلى تحديد العيوب والنواقص، والتحقق من أنها نواقص بالفعل وتسجيلها، ثم جدولة من يجب أن يقدم الحلول ومتى.

وخلافاً لاجتماعات المراجعة العامة تجري اجتماعات التدقيق بتواتر أقل وقد تهتم بدراسة مواضيع محددة وحرجة، وهي أكثر صورية ودقة.

يُنظَم اجتماع التدقيق في عدد من الأطوار، تبدأ بطور التخطيط الذي يحدد الأعضاء المشاركين والناحية المستهدفة للتدقيق.

قد يُعقَد اجتماع تمهيدي قبل جلسة التدقيق يُقدِم فيه مطور المنتج الموضوع العام، وتُسلَّم المواد التي يجب تدقيقها إلى المشاركين خلال الاجتماع التمهيدي أو قبله.

يُعقَد الاجتماع التمهيدي عادةً قبل اجتماع التدقيق بأسبوع لإعطاء فريق التدقيق الوقت الكافي لدراسة المواد والتحضير للاجتماع. يجري

المنارة للاستشارات
www.manaraa.com

خلال الاجتماع تحديد العيوب وتسجيلها وترقيمها، ليُحضِّر المُنسق بعد الاجتماع مباشرةً سجل العيوب الذي يُفضَّل أن يُسجَّل في أداة لإدارة التغيرات مقترنة بالمشروع.

يُطلَب من المطور عادة أن يتلافى العيوب بسرعة وأن يسجل حلولها في أداة إدارة التغيرات، ليتحقق المنسق من أن المطور قد أصلح مواطن الخلل وليقرر إذا كانت إعادة التدقيق ضرورية أم لا. وعند إقناعه بالحل يُرسِل المنسق (بالاتفاق مع مدير المشروع) مجتزأ التطوير إلى مجموعة ضمان جودة البرمجيات في المؤسسة (إذا كانت هذه المجموعة موجودة)، وذلك للتأكد من جودة المجتزأ.

الاختبار إزاء التوصيف (اختبار الصندوق الأسود)

- إن الاختبار إزاء التوصيف هو شكل من الاختبارات المعتمدة على التنفيذ، فهو يُطبَّق على المنتج البرمجي التنفيذي وليس على الوثائق أو النماذج.
- إن المبدأ الأساسي لهذا الاختبار هو أن يتعامل المطور مع مجتزأ الاختبار كصندوق أسود يأخذ دخلاً ما ويولد خرجاً ما، دون محاولة فهم الخوارزميات الحسابية أو منطق البرنامج.
- يَكشُف هذا النوع من الاختبارات عيوباً يصعب عادة كشفها بوسائل أخرى، وهو يكتشف على وجه الخصوص الوظائف المفقودة.

إن الاختبار إزاء التوصيف هو شكل من الاختبارات المعتمدة على التنفيذ، فهو يُطبَّق على المنتج البرمجي التنفيذي وليس على الوثائق أو النماذج، ويُعرَف هذا الاختبار بعدة أسماء مختلفة كاختبار الصندوق الأسود، أو الاختبار الوظيفي أو الاختبار الموجه بالدخل/الخرج... الخ.
إن المبدأ الأساسي لهذا الاختبار هو أن يتعامل المطور مع مجتزأ الاختبار كصندوق أسود يأخذ دخلاً ما ويولد خرجاً ما، دون محاولة فهم الخوارزميات الحسابية أو منطق البرنامج.

يتطلب هذا الاختبار أن تُشتَق متطلبات الاختبار من متطلبات حالات الاستخدام ليجري بعدئذ تحديدها وتوثيقها في خطة اختبار منفصلة وفي وثائق حالات اختبار. تعطي هذه الوثائق سيناريو للاختبار، ويمكن تسجيل السيناريوهات في أداة تسجيل لاستخدامها لاحقاً.
يَكشِف هذا النوع من الاختبارات عيوباً يصعب عادةً كشفها بوسائل أخرى، وهو يكشف على وجه الخصوص الوظائف المفقودة (أي شيء ما تم توثيقه كمتطلب حالة استخدام لكنه لم يبرمج أبداً).

الاختبار إزاء الرماز (اختبار الصندوق الأبيض)

- اختبار الصندوق الأبيض هو طريقة تصميم لحالات اختبار تعتمد بنية التحكم المُستخدمة في تصميم الإجرائيات ضمن المجتزأ البرمجي الذي يجري اختباره
- يضمن أن جميع المسارات المستقلة ضمن المجتزأ البرمجي قد جُرّبت مرة واحدة على الأقل
- يضمن أن جميع القرارات المنطقية قد جُرّبت من الوجهتين (الشرط محقق، الشرط غير محقق)
- يضمن تنفيذ جميع الحلقات على حدودها وضمن مجالات حدودها العملية
- يضمن تجربة بنى المعطيات الداخلية لضمان صلاحيتها

إن الاختبار إزاء الرماز أو ما يُعرف باختبار الصندوق الأبيض هو طريقة تصميم لحالات اختبار تعتمد بنية التحكم المستخدمة في تصميم الإجرائيات ضمن المجتزأ البرمجي الذي يجري اختباره

المنارة للاستشارات
www.manaraa.com

باستخدام طرائق اختبار الصندوق الأبيض، يمكن أن يستنتج مهندس البرمجيات حالات اختبار تحقق ما يلي:

- تضمن أن جميع المسارات المستقلة ضمن المجتزأ البرمجي قد جُرّبت مرة واحدة على الأقل
- تُجرّب جميع القرارات المنطقية من الوجهتين (الشرط محقق، الشرط غير محقق)
- تُنفّذ جميع الحلقات على حدودها وضمن مجالات حدودها العملية
- تُجرّب بنى المعطيات الداخلية لضمان صلاحيتها

اختبار قيود النظام

- يُعتبر اختبار قيود النظام من الاختبارات المعتمدة على التنفيذ، وهو يهدف إلى التأكد من تحقيق قيود النظام كما وردت في المتطلبات وفي وثائق الاختبار
- يشمل اختبار قيود النظام اختبارات مثل: اختبار واجهة الاستخدام، اختبار قاعدة المعطيات، اختبار الترخيص، اختبار الأوامر، اختبار الإجهاد، اختبار تجاوز الفشل، اختبار التشكيل، اختبار التثبيت

يُعتبر اختبار قيود النظام من الاختبارات المعتمدة على التنفيذ، وهو يهدف إلى التأكد من تحقيق قيود النظام كما وردت في المتطلبات وفي وثائق الاختبار. ويشمل اختبار قيود النظام اختبارات مثل:

- اختبار واجهة الاستخدام
- اختبار قاعدة المعطيات
- اختبار الترخيص
- اختبار الأوامر
- اختبار الإجهاد
- اختبار تجاوز الفشل
- اختبار التشكيل
- اختبار التثبيت

اختبار واجهة النظام

- تتداخل عملية اختبار واجهة الاستخدام البيانية مع مجمل مراحل إجرائية تطوير البرمجية. فهي تبدأ في مرحلة تحليل المتطلبات من خلال تضمين وثائق حالات الاستخدام رسوماً أولية للنوافذ والنمذجة الأولية للواجهات.
- يقوم فريق الاختبار بإجراء اختبارات منهجية على واجهات الاستخدام وذلك بعد تحقيق النظام، كما يختبر الزبائن الواجهات قبل التسليم النهائي للبرمجيات.

تتداخل عملية اختبار واجهة الاستخدام البيانية مع مجمل مراحل إجرائية تطوير البرمجية. فهي تبدأ في مرحلة تحليل المتطلبات وتتجلى من خلال أنشطة عديدة كتضمين وثائق حالات الاستخدام رسوماً أولية للنوافذ والنمذجة الأولية للواجهات. وتُركِّز هذه الاختبارات المُبكرة على تلبية المتطلبات الوظيفية وقابلية الاستخدام.

بعد أن يتم تحقيق النظام تظهر الحاجة لإجراء اختبارات منهجية على واجهات الاستخدام، يبدأ بها المطورون أنفسهم ثم يجري فريق الاختبار

المنارة للاستشارات
www.manaraa.com

اختبارات خاصة به. وأخيراً وقبل تسليم البرمجيات يختبر الزبائن الواجهات.

وفي ما يلي قائمة ببعض الأسئلة التي تظهر في وثيقة اختبار الواجهات بعد التحقيق:

- هل يتناسب اسم النافذة مع وظيفتها؟
- هل النافذة إلزامية أم لا؟ وهل يجب أن تكون كذلك؟
- هل تم التمييز بين الحقول الإجبارية والحقول الاختيارية بصرياً؟
- هل يمكن تغيير حجم النافذة، تحريكها، إغلاقها، واستعادتها؟ وهل يجب توفر ذلك؟
- هل تفتقد النافذة لحقل ما؟
- هل يتحقق برنامج الزبون من صلاحية القيم التي يدخلها المستخدم؟
- هل تملأ القوائم المنسدلة بقيم صحيحة من قاعدة المعطيات؟
- هل رسائل الأخطاء واضحة ومقروءة ويستطيع المستخدم الاستجابة لها بسهولة؟

اختبار قاعدة المعطيات

- يتداخل اختبار قاعدة المعطيات مع العديد من أنواع الاختبارات الأخرى، إذ تعتمد معظم اختبارات الصندوق الأسود على مدخلات قاعدة المعطيات ومخرجاتها.
- يشتمل اختبار قاعدة المعطيات بعد التحقيق على كم كبير من اختبارات الصندوق الأبيض (أي الاختبار إزاء الرماز)، وأهم جزء في اختبار قاعدة المعطيات هو اختبار المناقلات.

يتداخل اختبار قاعدة المعطيات، كاختبار واجهات الاستخدام البيانية، مع العديد من أنواع الاختبارات الأخرى. إذ تعتمد معظم اختبارات الصندوق الأسود على مدخلات قاعدة المعطيات ومخرجاتها. لكن يبقى هناك ضرورة لإجراء اختبارات منهجية على قاعدة المعطيات.
يشتمل اختبار قاعدة المعطيات بعد التحقيق على كم كبير من اختبارات الصندوق الأبيض (أي الاختبار إزاء الرماز)، وأهم جزء في اختبار قاعدة المعطيات هو اختبار المناقلات، ويمكن إجراء اختبارات مستقلة لبعض النواحي الأخرى في قاعدة المعطيات كالأداء والتنافس.
وفي ما يلي قائمة ببعض الأسئلة التي تظهر في وثيقة اختبار قاعدة المعطيات:

- هل تنفذ المناقلة مع المدخلات الصحيحة كما هو متوقع؟
- هل يعيد النظام إلى واجهة الاستخدام معلومات صحيحة؟ هل تبقى محتويات قاعدة المعطيات صحيحة بعد المناقلة؟
- إذا أوقفت المناقلة قبل انتهائها. هل يعيد النظام معلومات صحيحة إلى واجهة الاستخدام؟ هل تبقى محتويات قاعدة المعطيات صحيحة؟
- شغِّل المناقلة نفسها على التزامن في عدة إجراءات واجعل إحدى المناقلات تضع قفلاً على مورد معطيات تحتاج إليه المناقلات الأخرى. هل يقدم النظام شرحاً مفهوماً لما يجري للمستخدمين؟ هل تبقى محتويات قاعدة المعطيات صحيحة بعد إنهاء المناقلات؟

اختبار الترخيص

- يهتم اختبار الترخيص بحماية أغراض الزبون (واجهة الاستخدام) وأغراض المُخدِّم (قاعدة المعطيات) من الاستخدام غير المُرخَّص به. كما يجب التأكد أن آليات الأمن في الزبون وفي المُخدِّم، ستؤمن حماية النظام من عمليات الدخول غير المرخص بها.
- تصنف سماحيات المخدم ضمن فئتين:
- سماحيات الوصول إلى أغراض المُخدِّم (الجداول، المناظر، الأعمدة، الإجراءات المخزنة،...)

المنارة للاستشارات
www.manaraa.com

- سماحيات تنفيذ عبارات SQL (الانتقاء، التحديث، الإضافة، الحذف،...)

يمكن النظر إلى اختبار الترخيص كامتداد طبيعي للنوعين الأوليين من اختبارات قيود النظام. حيث يجب حماية أغراض الزبون (واجهة الاستخدام) وأغراض المُخدِّم (قاعدة المعطيات) من الاستخدام غير المرخّص به. كما يجب التأكد أن آليات الأمن في الزبون وفي المُخدِّم ستحمي النظام من عمليات الدخول غير المُرخَّص بها.
من الواضح أن قاعدة المعطيات هي التي ستعاني من نتائج اختراق أمن النظام، ومع ذلك تبدأ الحماية من برنامج الزبون. ويفضل أن تكون واجهة استخدام البرنامج قادرة على إعادة تشكيل ذاتها ديناميكياً لتتناسب مع مستوى الصلاحيات الممنوح للمستخدم الحالي، وإذا لم يكن للمستخدم حقوق الوصول المناسبة يفضل حجب إمكانية الوصول إلى بعض قوائم الأوامر أو الأزرار أو حتى نوافذ بأكملها.

تصنف سماحيات المُخدِّم ضمن فئتين:
– سماحيات الوصول إلى أغراض المُخدِّم (الجداول، المناظر، الأعمدة، الإجراءات المخزنة،...)
– سماحيات تنفيذ عبارات SQL (الانتقاء، التحديث، الإضافة، الحذف،...)

اختبار القيود الأخرى

– اختبار الإجهاد: تصمم اختبارات الإجهاد لدراسة تصرف النظام عند فرض طلبات غير اعتيادية (مثل انخفاض الموارد، أو التزاحم غير العادي على الموارد). ويُقرن اختبار الإجهاد غالباً باختبار الأوامر وقد يتطلب أدوات برمجية وعتادية خاصة.
– اختبار تجاوز الفشل: يهتم بدراسة استجابة النظام لأشكال مختلفة من المشكلات العتادية أو الشبكية أو البرمجية. ويرتبط هذا النوع من الاختبارات بإجراءات الاستعادة التي تدعمها نظم إدارة قواعد المعطيات.
– اختبار التشكيل: يتحقق اختبار التشكيل من كيفية عمل النظام تحت تشكيلات برمجية وعتادية متنوعة. ففي معظم بيئات البرمجة يتوقع أن يعمل النظام بنجاح على مختلف محطات عمل الزبائن التي تتصل بقاعدة المعطيات عبر بروتوكولات شبكية مختلفة، لكن قد تتواجد على بعض محطات عمل الزبائن برمجيات مثبتة تتعارض مع الإعدادات المتوقعة.

– اختبار التثبيت: يشكل اختبار التثبيت امتداداً لاختبار التشكيل يهدف إلى التحقق من صحة عمل النظام على كل المنصات التي ثُبِّت عليها. وهذا يعني إعادة تشغيل اختبارات خدمات النظام.

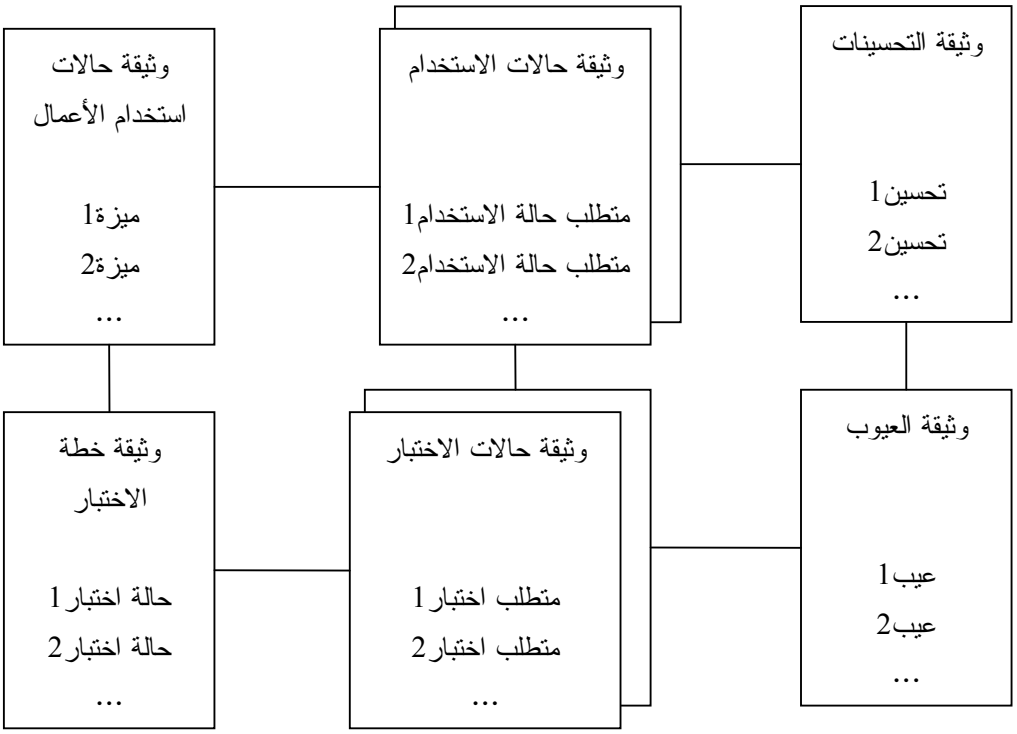
إدارة التغييرات

تكشف الاختبارات عيوب النظام التي يجب بالتالي تحديد مصادرها وتلافيها، ولبلوغ هذه الغاية يجب صياغة هذه العيوب كطلبات تغيير توزع على المطورين لتحقيقها. وقد تكون بعض طلبات التغيير عبارة عن تحسينات إضافية وليست عيوباً. وسواء كانت التغييرات المطلوبة عيوباً أم تحسينات فهي تؤثر على النظام وربما يجب ترتيب أولوياتها وتعقبها إلى مصادرها الأساسية في وثائق حالات الاستخدام ووثائق الاختبارات وغيرها.
تعتبر إدارة التغيرات مهمة كبيرة في أي مشروع برمجي يشارك في تطويره عدة مطورين، لذلك ومن أجل إدارة التغيرات بنجاح لا بد من توفر أداة لإدارة طلبات التغيير (جزء من أداة مساندة في هندسة البرمجيات).

المنارة للاستشارات
www.manaraa.com

وثائق الاختبارات وإدارة التغييرات

- تُشكِّل وثائق الاختبارات وإدارة التغيرات جزء من وثائق النظام الأخرى.
- تُرقّم متطلبات الاختبار وتُنظّم في بينة هرمية بطريقة مماثلة لطريقة تنظيم متطلبات حالات الاستخدام، ويظهر تقابل مباشر بين العديد من متطلبات الاختبار ومتطلبات حالات الاستخدام.



تحتوي وثيقة حالات الاختبار على مقاطع متطلبات لاختبار واجهة الاستخدام البيانية واختبار قاعدة المعطيات إلى جانب متطلبات اختبار المكونات العامة القابلة لإعادة الاستخدام.

تُشكِّل وثائق الاختبارات وإدارة التغيرات جزء من وثائق النظام الأخرى.

ويمكن كتابة خطة الاختبارات الأولية بالاعتماد على مزايا النظام المحددة في التوصيف.

ويستخدم بعدئذ نموذج حالات العمل لكتابة وثائق حالات الاختبار وتحديد متطلباته. وتُوثّق العيوب التي تُكتشف في مرحلة الاختبار في وثيقة العيوب، وتُسجَل متطلبات حالات الاستخدام غير المحققة في وثيقة التحسينات.

تُرقّم متطلبات الاختبار وتُنظَّم في بينة هرمية بطريقة مماثلة لطريقة تنظيم متطلبات حالات الاستخدام، ويظهر تقابل مباشر بين العديد من متطلبات الاختبار ومتطلبات حالات الاستخدام.

تحدد مقاطع أخرى في وثيقة حالات الاختبار متطلبات اختبار واجهة الاستخدام البيانية واختبار قاعدة المعطيات إلى جانب متطلبات اختبار المكونات العامة القابلة لإعادة الاستخدام.

المنارة للاستشارات
www.manaraa.com

القسم السادس عشر

الجودة

ملخص:

سنعرض في هذه الجلسة لمحة عامة عن مفهوم جودة البرمجيات، والذي يشتمل على عدة أفكار تتعلق بإدارة الجودة والأدوات المستخدمة لذلك، وكيفية إجراء المراجعات على النظام أثناء عملية مراقبة الجودة بالإضافة إلى آليات القياس واستراتيجيات الاختبار المتبعة وكيفية توثيق التغيرات.

أهداف تعليمية:

سيتعرف الطالب في هذا الفصل على المفاهيم التالية:

- تعريف الجودة
- جودة التصميم وجودة التطابق
- عملية إدارة جودة البرمجيات
- عملية التحكم بكيفية إدارة الجودة
- تكلفة الجودة
- الفرق بين تكلفة الجودة وتكلفة إصلاح العيوب!
- طرائق حساب تكلفة الجودة:
  - تكاليف منع حدوث الأخطاء
  - تكاليف التقييم والاختبار
  - تكاليف التعافي من المشاكل.
- آليات ضمان الجودة
  - المراجعات والاختبارات
  - توثيق الاختبارات.
- قياس الجودة
- عوامل الجودة البرمجية:
  - عوامل قابلة للقياس المباشر
  - وعوامل غير قابلة للقياس المباشر.
- عوامل McCall في تقييم جودة البرمجيات.

المنارة للاستشارات
www.manaraa.com

مقدمة

- سنتحدث في هذه الجلسة عن مفهوم ضمان جودة البرمجيات والمعايير المرتبطة بهذا الموضوع
- سنقوم بإلقاء الضوء على عملية إدارة الجودة والأدوات المستخدمة لتحقيق تلك العملية
- سنتحدث عن عملية مراقبة الجودة، وكيفية إجراء المراجعات والاختبارات
- سنركز من خلال الشرائح التي سندرسها أهمية الاختبارات في تقييم جودة البرمجيات بالإضافة إلى آليات القياس واستراتيجيات الاختبار المتبعة وكيفية توثيق التغيرات
- سنستعرض بعض الإجرائيات المستخدمة لتقييم جودة البرمجيات ولقياسها ولمقارنة التطبيقات بعضها مع بعض.

تمهيد

- يهدف مفهوم ضمان جودة البرمجيات بشكل رئيسي إلى تأمين برمجيات عالية الجودة، وذلك من خلال تطبيق قواعد ومعايير مراجعة واختبار على كافة مراحل تطوير المنتج البرمجي، أي منذ مرحلة جمع المتطلبات وحتى الانتهاء من مرحلة توليد الرماز، وذلك لكي نؤكد على استمرارية عملية ضمان الجودة أثناء كافة مراحل إجرائية التطوير
- يمكننا أن نلاحظ مجموعة من النقاط الرئيسية التي تحدد مراحل عملية ضبط الجودة، وهي:
  - عملية إدارة الجودة
  - الأدوات المستخدمة والطرائق المتبعة لضبط الجودة
  - استراتيجيات الاختبار والمراجعة
  - طرائق توثيق التغيرات
  - طرائق قياس التغيرات ومقارنتها مع المنتجات الأخرى.
- سنقوم من خلال الشرائح التالية بدراسة النقاط التي قمنا بالإشارة إليها بأسلوب مفصل.

المنارة للاستشارات
www.manaraa.com

تعريف الجودة

- تعريف:
  - تشير عادةً جودة شيء محدد إلى خصائصه القابلة للقياس، من طول أو عرض أو وزن أو لون أو مرونة وغيرها، إلا أن المشكلة التي تواجه تعريف جودة البرمجيات، هي أن البرمجيات عبارة عن كيانات غير ماديّة لا يمكن قياسها بالواحدات المعروفة، مما أدى بالضرورة إلى إيجاد واحدات قياس خاصة بالبرمجيات، كتعقيد التطبيق أو حجم الوظائف أو عدد أسطر الرماز ...وغيرها

- تخضع البرمجيات لمعايير جودة مغايرة نوعاً ما لتلك المعايير التي تخضع لها المنتجات المصنّعة، فإذا كانت الجودة بشكل عام تصنّف ضمن نوعين أساسيين هما جودة التصميم وجودة التطابق، أو ما يُعرف باسم درجة التوافق، أي جودة الخواص التي وضعها المصممون ودرجة اتّباع مواصفات التصميم أثناء مرحلة التصنيع، فإن التعبير عن جودة تصميم البرمجيات يمكن أن يتحدد من خلال جودة تحديد وتوصيف المتطلبات وتصميم النظام من جهة، ودرجة توافق التنجيز مع المعايير المحددة له أثناء مرحلة التصميم من جهة أخرى

إدارة الجودة والتحكم بها

- يمكننا أن نعرّف عملية إدارة جودة البرمجيات بأنها مجموع عمليات الاختبار والمراجعة والتنقيح التي يخضع لها المنتج خلال مراحل دورة حياته، وذلك للتحقق من أن النواتج توافق المتطلبات المحددة أثناء تحليل النظام. ويتضمن ذلك عمليات التغذية الراجعة التي تربط ما بين مراحل التطور تلك، إذ غالباً ما تؤدي عمليات ضمان الجودة التي يتم تطبيقها أثناء عملية التطوير إلى إعادة تصنيع بعض المجتزآت، وهنا يمكننا أن نلاحظ الترابط ما بين نتائج التغذية الراجعة وبين عمليات إدارة جودة المنتج

- أما بالنسبة لعملية التحكم بكيفية إدارة الجودة، فيمكن لها أن تكون عملية مؤتمتة جزئياً أو كلياً، وذلك من خلال أدوات خاصة أو من خلال خطة مراقبة يدوية معيّنة، ولا توجد أي مشكلة في عملية التحكم تلك بكافة أشكالها، إلا أنه من الضروري أن تتم الإشارة إلى أهمية توافق تلك العملية مع مبادئ قابلة للقياس يتم تطبيقها على كافة مراحل عملية التطوير وذلك لاكتشاف المشاكل والأخطاء وإصلاحها

- تتضمن عملية إدارة الجودة، إمكانية إصدار تقارير اختبار ملائمة وتزويدها للعاملين على تطوير المنتج البرمجي، وذلك بهدف تأمين الصورة الصحيحة عن المنتج وعن جودته، ما يسمح للقائمين على إدارة عملية التطوير بتقييم المنتج أو تأمين الموارد المناسبة لتطويره وتحسينه.

المنارة للاستشارات
www.manaraa.com

تكلفة الجودة

تعريف

- تتباين تكلفة إيجاد الأخطاء وإصلاحها من مرحلة إلى أخرى أثناء تطوير أي منتج برمجي، إذ تزداد وبشكل ملحوظ كلما تأخرت عملية اكتشاف الأخطاء، فلا تعتبر مسألة اكتشاف عيب ما في مرحلة تحليل المنتج البرمجي وإصلاح ذلك العيب، بالعملية المكلفة نسبياً عند مقارنتها بتكلفة ذلك الإجراء بعد اكتشاف الخطأ من قبل الزبون أثناء استخدامه للنظام

- بتطبيقنا لمبدأ "الوقاية خير من العلاج" على كافة مراحل تطوير المنتج البرمجي نكون قد قمنا بتطبيق سياسة ضمان الجودة، إلا أن لعملية الوقاية تلك تكلفة بحد ذاتها، أو بعبارة أخرى، تخضع عمليات ضمان جودة البرمجيات لتكلفة مضافة إلى تكلفة المنتج بحد ذاته، ولكن بالنظر إلى ما يمكن أن توفره عملية ضمان الجودة من مصاريف وتكاليف أخرى تُضاف أثناء اكتشاف العيوب، يمكننا أن نؤكد أن لعملية إدارة الجودة وزن هام أثناء مراحل تطوير أي منتج برمجي

تكلفة الجودة

الفرق بين تكلفة الجودة وتكلفة إصلاح العيوب

- قبل أن نبدأ بالحديث عن كيفية احتساب أو تقدير تكاليف عملية ضمان الجودة، لابد أولاً أن نوضح أنَّه من الممكن أن تنجم عن عملية إصلاح العيوب تكاليف تتجاوز تكاليف عملية ضمان الجودة بعشرات المرات، خاصة إذا ما تم اكتشاف العيوب أثناء مرحلة الاستثمار الفعلي للتطبيق

- مثال:

لنفترض أن مؤسسة ما قد أنفقت ما يعادل 500 ساعة عمل إضافية في اختبار وتنقيح رماز يبلغ 100000 سطر، وكانت النتيجة أنه قد تم تفادي 200 خطأ محتملاً، وعلى افتراض أن الأجرة الوسطية لساعة عمل المبرمج تساوي 30 $، سيترتب بالتالي تكلفة إضافية على عاتق المؤسسة تعادل 30 × 500= 15 000 دولار لتجنب 200 خطأ محتملاً، أي ما يعادل 75$ لكل خطأ

وعلى افتراض أنه لم تتم إجراء عمليات الاختبار تلك وأن تكلفة التعافي من كل خطأ يظهر على أرض الواقع بعد استثمار النظام تساوي وسطياً 1000$ (وذلك بعد تضمين تكاليف تعطُّل الزبون وتكاليف الإصلاح وإعادة التركيب)، وعلى افتراض أنه لم يظهر سوى عيب وحيد عن كل ألف سطر رماز غير مختبر وفق الإجرائية السابقة، يكون بالتالي عدد العيوب الناتجة يساوي 100 عيب وتكلفة إصلاحها تساوي 100000$ أي ما يعادل حوالي سبعة أضعاف التكلفة السابقة

على الرغم من أن المثال السابق مبني على قيم افتراضية، إلا أن تطبيقه العملي وارد، إذ تثبت ذلك العديد من الدراسات التي تقوم بها الشركات والمؤسسات البرمجية.

طرائق حساب تكلفة الجودة

- يتم عادةً حساب تكلفة عملية ضمان الجودة والتحكم بها من خلال تقديرات مادية تقريبية تعطي دلالة وسطية عن التكاليف المرتبطة بتلك العملية بحيث يمكن توزيع تلك التكاليف ضمن ثلاثة أنماط رئيسية، وهي: تكاليف منع حدوث الأخطاء وتكاليف التقييم

المنارة للاستشارات
www.manaraa.com

والاختبار وتكاليف التعافي من المشاكل بعد الانتهاء من تصميم المنتج وإعداده. سنقوم فيما يلي بشرح كل نمط من أنماط التكـــاليف هذه:

- تكاليف منع حدوث الأخطاء:

تتضمن تكاليف منع حدوث الأخطاء كل ما يتعلق بالإنفاقات المخصصة لإدارة عملية الجودة، سواء كانت على أدوات الاختبار أو مخططات التقييم أو وثائق المراجعات التقنية أو طرائق التدريب والاختبار التي يتم التخطيط لها

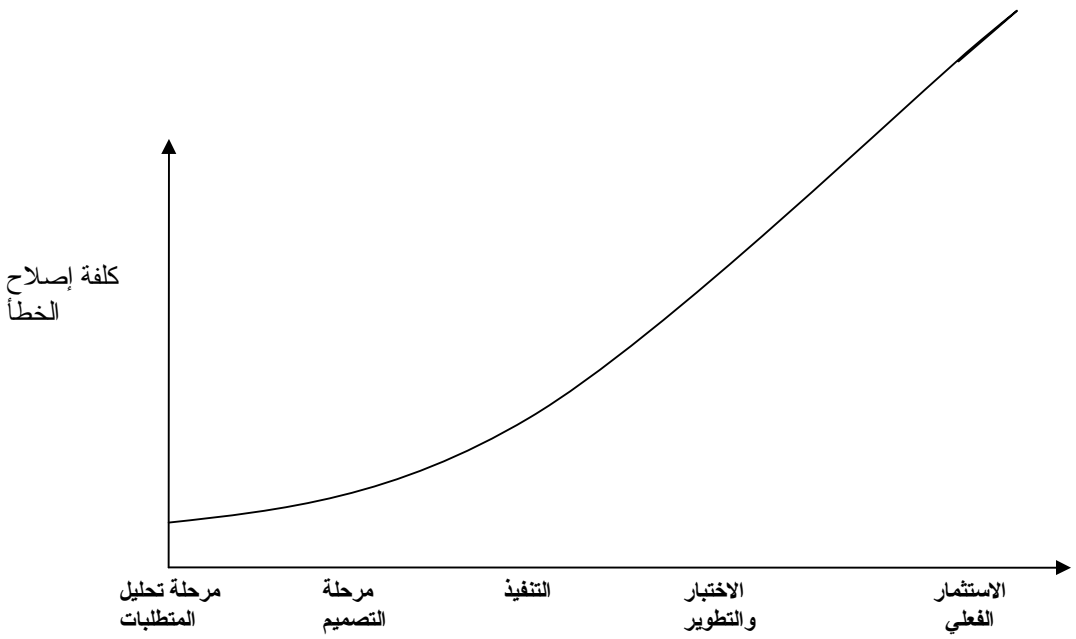
- تكاليف التقييم والاختبار:

تتضمن تكاليف التقييم والاختبار كافة الإنفاقات التي تتم على الإجراءات التي تضمن توصيف حالة المنتج الحالية، أي تقييم وضع مختلف إجرائيات المنتج ووظائفه في الوقت الحالي. ويتضمن ذلك تكلفة إجرائيات التفتيش والاختبار والمعاينة والصيانة

- تكاليف التعافي من المشاكل:

تتضمن تكاليف التعافي من المشاكل، كافة الإنفاقات التي تتم من أجل التعافي من عطل محدد ظهر على المنتج بعد الانتهاء من تطويره، ويمكننا هنا أن نلاحظ نوعين مختلفين من الأعطال، أعطال تظهر على المنتج قبل تسليمه للزبون، وأعطال تظهر بعد تسليم المنتج. إذ تختلف تكاليف إصلاح الأعطال ما بين هذين النوعين ومن المنطقي أن نتوقع تكاليف أكبر بكثير للنوع الثاني، أي عند اكتشاف المشاكل بعد التسليم

تتضمن تكاليف التعافي من الأعطال قبل مرحلة التسليم، تكاليف الإصلاح وتحليل الأخطاء وإعادة التشغيل من جديد، أما تكاليف التعافي من الأعطال بعد مرحلة الشحن للزبون فتتضمن تكاليف معالجة الشكوى واسترجاع المنتج وإصلاحه وإعادة شحنه، بالإضافة إلى تكاليف الدعم التقني والكفالة التي ترافق عملية الاستخدام ومدتها.



المنارة للاستشارات
www.manaraa.com

آليات ضمان الجودة
والاختبارات المراجعات

- تلعب عمليات المراجعة التي تُطبق على المنتج البرمجي خلال مراحل تطويره دور أداة التنقية أو التنقيح التي تساهم في اكتشاف الأخطاء والشوائب وإصلاحها، سواء كانت أخطاء تحليلية أو تصميمية أو أخطاء في كتابة الرماز أو في الخوارزميات المستخدمة

- يعتبر الخطأ جزءاً من الطبيعة البشرية، ومهمة تقصي الأخطاء وإصلاحها ليست بالمهمة المستحيلة، إلا أن المشكلة الحقيقية تكمن في صعوبة اكتشاف الشخص لأخطائه بشكل مباشر، في حين يمكن للآخرين أن يلحظوها بعناء أقل في كثير من الأحيان، أو بطريقة أخرى، يصعب على مَن ارتكب خطأً برمجياً ما أن يلحظ خطأه، في حين يمكن لمراقب خارجي أن يكتشف أخطاءً لم يلحظها المبرمج، بمجرد مراقبة أسلوب التطوير من منظور آخر أو من خلال رأي مخالف، ونركز هنا على مفهوم الرأي المخالف لما له من أهمية كبيرة في تطوير النظم البرمجية، فكثيراً ما تكشف الآراء المتعارضة أخطاءً لم يُلتفت إليها مسبقاً

- إن عملية المراجعة هي عبارة عن صيغة لاستثمار قدرات الفريق أو مجموعة الأشخاص المرتبطين بعملية التطوير من أجل اكتشاف الأخطاء المحتملة في المنتج البرمجي، أو للبحث عن التحسينات التي يمكن أن يتم تطبيقها، أو لحصر الأجزاء التي لا تتطلب المزيد من التطوير، أي التي تؤدي مهماتها بأسلوب مناسب

- يمكن أن تتضمن كوادر عمليات المراجعة والاختبار أشخاصاً بعيدين عن دورة التطوير المرتبطة بالمنتج الذي يتم اختباره، أو بأسلوب آخر، أشخاصاً عاديين تُلقى على عاتقهم مهمات استخدام النظام –أو مجتزآت محددة منه– بغرض البحث عن الأخطاء أو العيوب المحتملة فيه

آليات ضمان الجودة
توثيق الاختبارات

- تعد مهمة تدوين وتسجيل الاختبارات من الخطوات التي لا ينبغي إهمالها أثناء سير عمليات ضمان الجودة، إذ من المهم أن نحتفظ بملخص عن كافة الإجراءات التي كانت تعاني من مشاكل أو أخطاء وذلك على شكل تقارير خاصة تُضاف إلى سجلّ المشروع للسماح بملاحقة وتتبع تلك الأخطاء

- ينبغي على تقرير التوثيق أن يحتوي على مجموعة النقاط التالية:
  - معلومات عن الشخص الذي يقوم بإجراء عملية الاختبار وتاريخ تلك العملية
  - معلومات حول موضوع الاختبار الذي تتم مراجعته
  - قائمة تفقد تحتوي على التوابع والمهمات الرئيسية للمجتزأ الذي يتم اختباره، بالإضافة إلى قائمة بكافة مكوناته، وذلك للمساعدة على تدوين الملاحظات حول كل مشكلة محتملة في أي جزء
  - النتائج والتعليقات والقرارات التي ينبغي اتخاذها

يمكن كذلك أن يحتوي هذا التقرير على ملحقات أو مرفقات من أنماط مختلفة لتساعد على التعرف على الأخطاء التي تم اكتشافها،

المنارة للاستشارات
www.manaraa.com

إذ يمكن أن توكل مهمات تصحيح الأخطاء المكتشفة إلى أشخاص آخرين من أعضاء فريق التطوير.

آليات ضمان الجودة

نصائح وإرشادات

ينبغي الانتباه إلى مجموعة من النقاط الهامة التي ترتبط بعملية الاختبار، نذكر منها:

- وضع جدول أعمال زمني يتقيد به كافة أعضاء فريق الاختبار، وذلك لضبط الحالات التي يتم فيها التركيز على بعض الأخطاء وإعطائها من الزمن ما يزيد عن اللزوم
- عرض المشاكل كافة دون اقتراح حلول لها، إذ تُلقى على عاتق فريق الاختبار مهمة البحث عن الأخطاء وليس إيجاد الحلول لها، بينما يمكن أن تُسند مهمات التصحيح إلى أفراد آخرين وذلك لاستثمار الوقت
- اختبار التطبيق من دون التطرق إلى المُطوِّر، إذ ينبغي أن تُثار الأخطاء بلطف وليونة كما ينبغي إيجاد طرائق مناسبة لإدارة هذا النوع من الجلسات لتحويلها إلى جلسات نقد بنّاء وليس محاكمة بسبب الأخطاء
- توثيق الاختبارات وكتابتها ضمن تقارير للعودة إليها من أجل عمليات التصحيح
- الحد من المناقشات والمجادلات التي يمكن أن تنشأ نتيجةً للاختلاف في بعض الآراء حول موضوع محدد، إذ من المهم في مثل هذه الحالات أن يتم تأجيل هذا النوع من المناقشات ليتم التحدث فيها في وقت آخر بعد الاجتماع
- مراجعة المراجعات

تقييم الجودة أم قياس الجودة

قبل أن نتحدث عن الطرائق المستخدمة في تقييم جودة المنتجات البرمجية، لابد أن نشير أولاً إلى سبب استخدام مصطلح تقييم الجودة بدلاً من المصطلح قياس الجودة البرمجية.

فكما أشرنا سابقاً، يمكننا تعريف القياس بأنه إسناد قيم وأعداد ثابتة إلى واصفات لكيانات مختلفة، وذلك وفقاً لقواعد وأسس ناظمة محددة، إلا أنه لا يمكننا بسهولة أن نستنتج واحدة قياس مناسبة لجودة البرمجيات في الوقت الراهن، وبالتالي نفضل استخدام مصطلح تقييم الجودة بدلاً من قياس الجودة، على الرغم من وجود معايير متعددة لمقارنة جودة المنتجات البرمجية في الوظائف التي تؤديها، كما سنرى في الشرائح التالية

عوامل الجودة البرمجية

- مازال الجدل حول الموضوع الذي يعتبر أنَّ البرمجيات غير قابلة للقياس، وأننا لن نستطيع إيجاد مقاييس مناسبة لها، مستمراً، إلا أن الكثيرين يرفضون هذا المنطق، فهناك العديد من المعايير التي تساهم في تقييم جودة المنتجات البرمجية وتزودنا بأساليب لإدراك الوضع الحالي لتلك المنتجات قبل فوات الأوان

- يمكن تصنيف العوامل التي تؤثر في جودة البرمجيات، ضمن مجموعتين أساسيتين، هما:
  - عوامل قابلة للقياس المباشر، كعدد الأعطال مثلاً
  - وعوامل غير قابلة للقياس المباشر، كقابلية الاستخدام، أو قابلية الصيانة وإصلاح الأخطاء

المنارة للاستشارات
www.manaraa.com

- اقترح McCall مجموعة من العوامل التي تؤثر في تقييم جودة البرمجيات، ويمكن تصنيفها ضمن ثلاثة مجموعات رئيسية، هي:
  - عوامل ترتبط بخصائص المنتج التشغيلية
  - عوامل ترتبط بقابلية المنتج للتطوير
  - وعوامل ترتبط بمدى قدرة المنتج على التكيف مع البيئة المحيطة به

- سنقوم فيما يلي بشرح عوامل McCall في تقييم جودة البرمجيات، بالتفصيل:
  - العوامل المرتبطة بخصائص المنتج التشغيلية:
    - الفعالية: وتعني مقدار حاجة التطبيق من موارد حاسوبية ورماز من أجل القيام بمهماته التي صمم من أجلها
    - السلامة: وهي مدى التحكم بولوج المستخدمين غير المخولين إلى النظام أو إلى المعطيات
    - قابلية الاستخدام: أي الجهد اللازم للتحكم بالتطبيق وتعلم استخدامه
    - الاعتمادية: أي مدى الثقة بالنظام، أو بطريقة أخرى، مدى إنجاز التطبيق للمهمات المرجوة منه بالدقة المطلوبة
    - الصِّحة: أي مدى تحقيق النظام للمتطلبات الوظيفية التي حُدّدت مسبقاً أثناء تحليل النظام، أو بطريقة أخرى، مدى موافقة النظام لأهداف الزبون
  - العوامل المرتبطة بقابلية المنتج للتطوير:
    - المرونة: أي الجهد اللازم لتعديل أو تطوير نظام قيد الاستثمار
    - قابلية الصيانة: أي الجهد اللازم لتحديد وإصلاح الأخطاء في النظام
    - الاختبارية: أي الجهد اللازم لاختبار النظام وإعداده للعمل
  - العوامل المرتبطة بتكيف المنتج مع البيئة المحيطة به:
    - قابلية النقل (المحمولية): أي إمكانية نقل النظام من بيئة إلى أخرى سواء كانت برمجية أو عتادية
    - إعادة الاستخدامية: أي مدى إمكانية إعادة استخدام البرنامج أو بعض مجتزآته ضمن تطبيقات أخرى
    - الترابطية مع النظم: أي مدى إمكانية ربط النظام مع نظم أخرى

المنارة للاستشارات
www.manaraa.com

القسم السابع عشر

دراسة واقعية

الكلمات المفتاحية:

الفاعلون، حالات الاستخدام، المتطلبات الوظيفية، مخطط النشاط، الكيانات، الصفوف، مخطط التسلسل، مخطط الحالات.

ملخص:

يُركِّز هذا الفصل على دراسة وتحليل حالة واقعية.

أهداف تعليمية:

يهدف هذا الفصل إلى:

دراسة وتحليل حالة واقعية (التسوق الالكتروني).

المنارة للاستشارات
www.manaraa.com

تدريب موجّه في نمذجة التحليل

- سنعرض في هذه الجلسة تدريباً على النمذجة المرئية في لغة UML، ونهدف بذلك إلى توضيح مخططات UML المختلفة وكيفية انسجام هذه المخططات فيما بينها، إذ يعرض كل مخطط في UML النظام من منظور معين، ولفهم النظام بكليته يجب أن نطور عدة مخططات، من زوايا نظر مختلفة، وأن نكامل بين هذه المخططات.

نص التدريب: التسوق الالكتروني

- التسوق الالكتروني (معالجة طلب الزبون):

يعرض أحد مصنعي الحواسيب إمكانية الشراء مباشرة عبر الإنترنت، حيث يمكن أن ينتقي الزبون حاسوباً عن صفحة الويب العائدة للمصنع، وتصنف الحواسيب في فئات: مخدمات، حواسيب مكتبية، حواسيب محمولة، ويمكن أن يختار الزبون أحد التشكيلات القياسية المعروضة أو أن يبني تشكيلاً حسب رغبته، حيث تظهر المكونات القابلة للانتقاء (كالذاكرة مثلاً) بصيغة قوائم خيارات، ويستطيع النظام حساب سعر أي تشكيلة جديدة.

ولإرسال طلب الشراء يجب أن يدخل الزبون المعلومات اللازمة للشحن وللدفع، حيث يقبل الدفع ببطاقات الائتمان وبالشيكات. بعد إدخال الطلب يرسل النظام رسالة إلى زبون بواسطة البريد الإلكتروني ليؤكد فيها تفاصيل الطلب، ويمكن للزبون طيلة فترة انتظاره استلام الحاسوب أن يتحرى عن حالة الطلب آنياً وفي أي وقت.

يجري التطبيق الأساسي مجموعة خطوات ليتحقق من ملاءة الزبون وطريقة الدفع، ولطلب قطع الحاسوب المُشكَّل من المخزن، ولطباعة الفاتورة، وليطلب من المخزن شحن الحاسوب إلى الزبون.

التحليل: الخطوة 1

- عد إلى نص المثال المذكور سابقاً وحاول تحديد الفاعلين في تطبيق التسوق الآني بعد الآخذ بعين الاعتبار التوسيع التالي للمتطلبات:
  - يستخدم الزبون صفحة الويب الخاصة بالمصنع لمعاينة التشكيلات القياسية للمخدمات والحواسيب المكتبية والحواسيب المحمولة، حيث يكون السعر ظاهراً أيضاً
  - يختار الزبون معاينة معلومات تفصيلية عن التشكيل، ربما بهدف شرائه كما هو أو بهدف بناء تشكيل يلائمه أكثر. ويمكن حساب سعر أي تشكيل عند طلب الزبون
  - قد يختار الزبون أن يطلب شراء الحاسوب آنياً، أو أن يطلب أن يتصل به مندوب المبيعات ليشرح له بعض التفاصيل أو يفاوضه على السعر قبل اعتماد الطلب
  - لاعتماد الطلب يجب أن يملأ الزبون استمارة تتضمن عنوان الشحن وعنوان الفاتورة، مع تفاصيل عن طريقة الدفع (بطاقة ائتمان أو شيك)
  - بعد إدخال طلب الزبون إلى النظام يرسل مندوب المبيعات طلباً بصيغة إلكترونية إلى المخزن يتضمن تفاصيل التشكيل المطلوب.
  - يرسل للزبون عبر البريد الإلكتروني تفاصيل المناقلة بما فيها رقم الطلب ورقم حساب الزبون بحيث يستطيع في أي وقت أن يتحرى عن حالة طلب
  - يحصل المخزن على الفاتورة من مندوب المبيعات ويشحن الحاسوب إلى الزبون
- وفي ما يلي مخطط فاعلي النظام:

المنارة للاستشارات
www.manaraa.com

التحليل: الخطوة 2

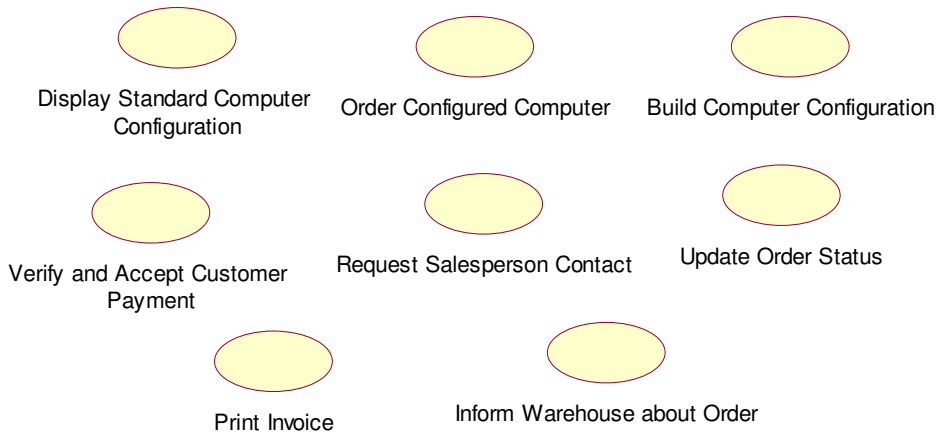
- عُد إلى الخطوة 1 وحدد حالات الاستخدام في تطبيق التسوق الآني
- يمكننا إنجاز هذه الخطوة ببناء جدول يربط المتطلبات الوظيفية بالفاعلين وبحالات الاستخدام، ونشير هنا إلى أن بعض الوظائف تقع خارج نطاق التطبيق وبالتالي يجب عدم تحويلها إلى حالات استخدام
- يربط الجدول التالي المتطلبات الوظيفية التي عددناها في الخطوة 1 بالفاعلين وبحالات الاستخدام، وتقع مهام المخزن المتعلقة بتشكيل الحاسوب وشحنه إلى الزبون خارج نطاق الوظائف

الرقم	المتطلب	الفاعل	حالة الاستخدام
1	يستخدم الزبون صفحة الويب الخاصة بالمصنع لمعاينة التشكيلات القياسية للمخدمات والحواسيب المكتبية والحواسيب المحمولة حيث يكون السعر ظاهراً أيضاً.	Customer	Display Standard Computer Configuration
2	يختار الزبون معاينة معلومات تفصيلية عن التشكيل، ربما بهدف شرائه كما هو أو بهدف بناء تشكيل يلائمه أكثر، ويمكن حساب سعر أي تشكيل عند طلب الزبون.	Customer	Build Computer Configuration
3	قد يختار الزبون أن يطلب شراء الحاسوب أنياً، أو أن يطلب أن يتصل به مندوب المبيعات ليشرح له بعض التفاصيل أو يفاوضه على السعر قبل اعتماد الطلب.	Customer, Salesperson	Order Configured Computer, Request Salesperson Contact
4	لاعتماد الطلب يجب أن يملأ الزبون استمارة تتضمن عنوان الشحن وعنوان الفاتورة، مع تفاصيل عن طريقة الدفع (بطاقة ائتمان أو شيك).	Customer	Order Configured Computer, Verify and Accept Customer Payment
5	بعد إدخال طلب الزبون إلى النظام يرسل مندوب المبيعات طلباً بصيغة الكترونية إلى	Salesperson, Warehouse	Inform Warehouse About Order

المنارة للاستشارات
www.manaraa.com

		المخزن يتضمن تفاصيل التشكيل المطلوب.	
Order Configured Computer, Update Order Status	Salesperson, Customer	يُرسَل للزبون عبر البريد الالكتروني تفاصيل المناقلة بما فيها رقم الطلب ورقم حساب الزبون بحيث يستطيع في أي وقت أن يتحرى عن حالة الطلب.	6
Print Invoice	Salesperson, Warehouse	يحصل المخزن على الفاتورة من مندوب المبيعات ويشحن الحاسب إلى الزبون.	7

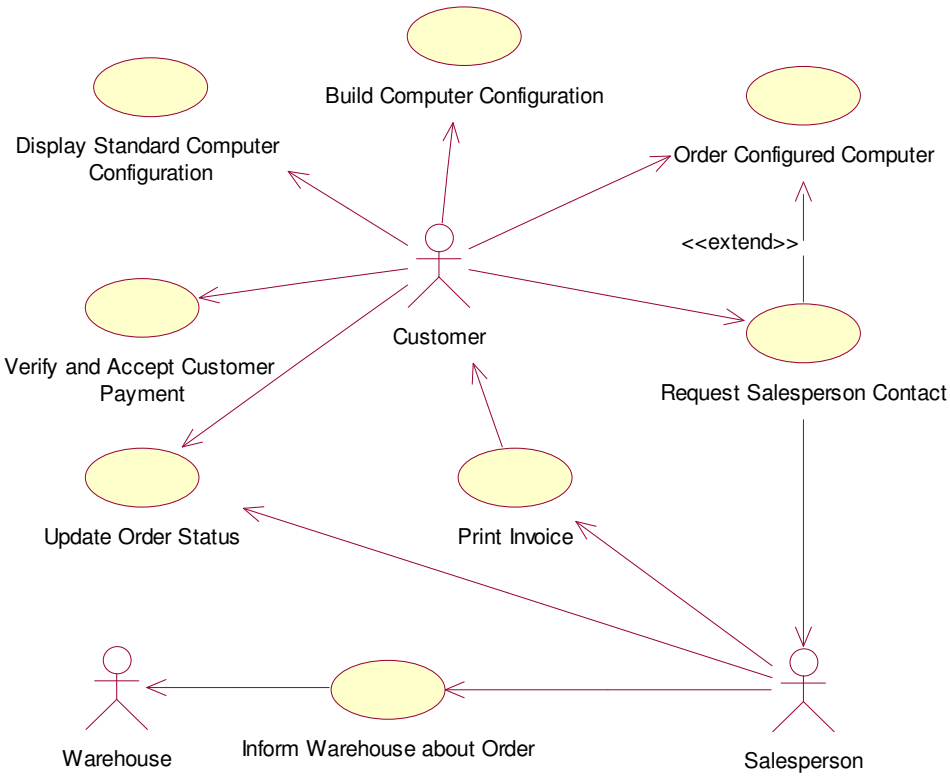
- ويبين الشكل التالي التدوين البياني لحالات الاستخدام:



التحليل: الخطوة 3

- ارسم مخطط حالات الاستخدام بعد دراسة الحالات التي سبق ذكرها
- يمكن إنجاز هذه الخطوة بالاعتماد مباشرة على معلومات تضمنتها الخطوات السابقة مع الأخذ بعين الاعتبار لأمر إضافي وحيد وهو العلاقات القائمة بين حالات الاستخدام (كما مرّ معنا)
- ويبين الشكل التالي مخطط حالات الاستخدام:

المنارة للاستشارات
www.manaraa.com



التحليل: الخطوة 4

- عد إلى المراحل السابقة لكتابة وثيقة توصيف لكل حالة من حالة الاستخدام، يمكنك الاعتماد على معلوماتك العامة في استنتاج تفاصيل غير مذكورة ضمن المتطلبات
- سنعرض فيما يلي توصيف مختصر لحالة الاستخدام "Order Configured Computer ":

حالة الاستخدام	Order Configured Computer
وصف موجز	تسمح هذه الحالة للزبون (Customer) بإدخال طلب الشراء. ويتضمن ذلك تزويد النظام بعنوان الشحن والفاتورة بالإضافة إلى تفاصيل طريقة الدفع.
الفاعلون	Customer
الشروط السابقة	يوجه الزبون (Customer) أحد برامج تصفح الإنترنت إلى صفحة الويب الخاصة بالمصنع. تعرض الصفحة معلومات تفصيلية عن مكونات الحاسوب إلى جانب سعره.
التدفق الرئيسي	تبدأ حالة الاستخدام هذه عندما يقرر الزبون أن يطلب شراء الحاسوب بانتقاء الوظيفة Continue (أو وظيفة باسم مشابه) عند ظهور تفاصيل الطلب على الشاشة. يطلب النظام من الزبون أن يدخل معلومات تفصيلية تتضمن: اسم مندوب المبيعات (إذا كان معروفاً)، تفاصيل الشحن (اسم الزبون

المنارة للاستشارات
www.manaraa.com

	وعنوانه). تفاصيل الفاتورة (إذا كانت مختلفة عن تفاصيل الشحن). طريقة الدفع (شيك أو بطاقة ائتمان) وأية تعليقات أخرى. يختار الزبون الوظيفة Purchase (أو وظيفة باسم مشابه) لإرساء الطلب إلى المصنّع. يسند النظام لطلب الشراء رقماً وحيداً مميزاً ورقم حاسب الزبون ويخزن معلومات الطلب في قاعدة المعطيات. يرسل النظام رقم الطلب ورقم الزبون إلى الزبون بواسطة البريد الإلكتروني، كتأكيد على قبول طلب الشراء.
التدفقات البديلة	ينشط الزبون وظيفة الشراء Purchase قبل إدخال كل المعلومات الضرورية. فيعرض النظام عندئذ رسالة خطأ ويطلب من الزبون إتمام المعلومات الناقصة. ينتقي الزبون الوظيفة Reset (أو وظيفة باسم مشابه) للعودة إلى طلب شراء فارغ، فيسمح النظام للزبون بإدخال المعلومات من جديد.
الشروط اللاحقة	إذا اكتملت حالة الاستخدام بنجاح يسجل طلب الشراء في قاعدة معطيات النظام، وإلا فتبقى حالة النظام كما هي دون تغيير.

المنارة للاستشارات
www.manaraa.com

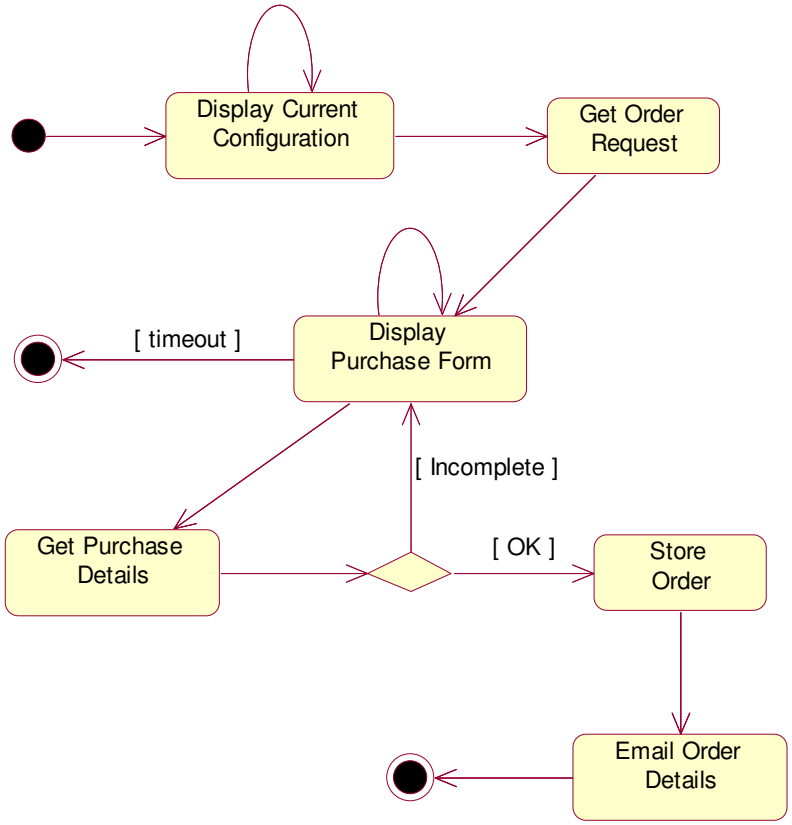
التحليل: الخطوة 5

- أوجد أنشطة حالات الاستخدام السابقة
- سنعرض فيما يلي أنشطة حالة الاستخدام "Order Configured Computer"، حيث يبين الجدول التالي الأنشطة ضمن تدفق الأحداث الرئيسي وفي التدفقات البديلة:

الرقم	عبارة حالة الاستخدام	حالة النشاط
1	تبدأ حالة الاستخدام هذه عندما يقرر الزبون أن يطلب شراء الحاسب بانتقاء الوظيفة continue (أو وظيفة باسم مشابه) عند ظهور تفاصيل الطلب على الشاشة.	Display Current Configuration; Get Order Request
2	يطلب النظام من الزبون أن يدخل معلومات تفصيلية تتضمن: اسم مندوب المبيعات (إذا كان معروفاً)، تفاصيل الشحن (اسم الزبون وعنوانه)، تفاصيل الفاتورة (إذا كانت مختلفة عن تفاصيل الشحن)، طريقة الدفع (شيك أو بطاقة ائتمان)، وأية تعليقات أخرى.	Display Purchase Form
3	يختار الزبون الوظيفة purchase (أو وظيفة باسم مشابه) لإرساء الطلب إلى المصنع.	Get Purchase Details
4	يسند النظام لطلب الشراء رقماً وحيداً مميزاً ورقم حساب الزبون ويخزن معلومات الطلب في قاعدة المعطيات.	Store Order
5	يرسل النظام رقم الطلب ورقم الزبون إلى الزبون بواسطة البريد الإلكتروني، كتأكيد على قبول طلب الشراء.	Email Order Details
6	ينشط الزبون وظيفة الشراء purchase قبل إدخال كل المعلومات الضرورية. فيعرض النظام عندئذ رسالة خطأ ويطلب من الزبون إتمام المعلومات الناقصة.	Get Purchase Details; Display Purchase Form
7	ينتقي الزبون الوظيفة reset (أو وظيفة باسم مشابه) للعودة إلى طلب شراء فارغ، فيسمح النظام للزبون بإدخال المعلومات من جديد.	Display Purchase Form

- وفي ما يلي مخطط النشاط لحالة الاستخدام "Order Configured Computer":

المنارة للاستشارات
www.manaraa.com



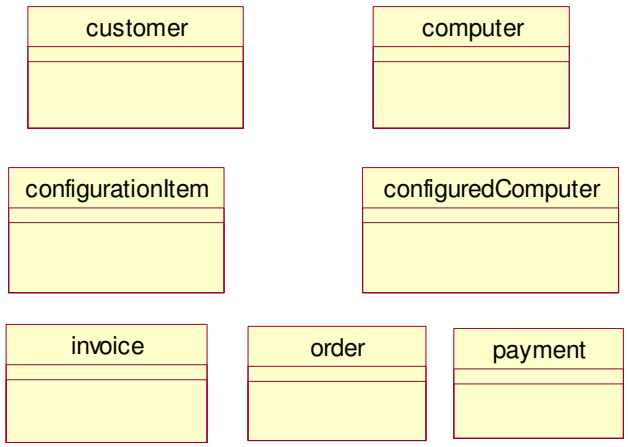
التحليل: الخطوة 6

- عد إلى المتطلبات المعرفة في النص، وأوجد صفوف الكيانات لتطبيق التسوق الآني.
- يمكننا بإتباع المنهج نفسه الذي اتبعناه في تحديد الفاعلين وحالات الاستخدام، أن ننشئ جدولاً يساعد في تحديد الصفوف من تحليل المتطلبات الوظيفية. يسند الجدول المتطلبات الوظيفية المذكورة في الخطوة 1 إلى صفوف الكيانات.
- إسناد المتطلبات إلى صفوف الكيانات:

المنارة للاستشارات
www.manaraa.com

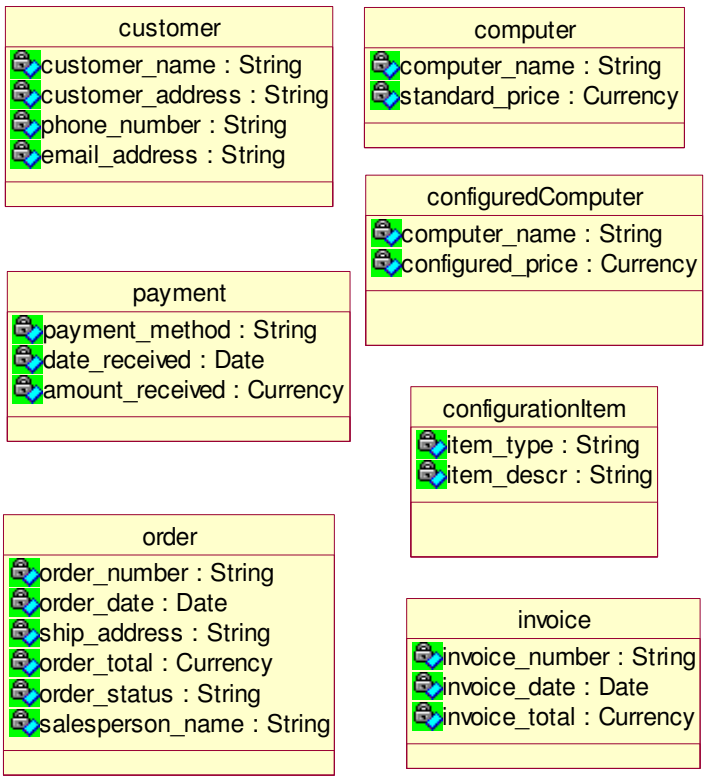
الرقم	المتطلب	صف الكيان
1	يستخدم الزبون صفحة الويب الخاصة بالمصنع لمعاينة التشكيلات القياسية للمخدمات والحواسيب المكتبية والحواسيب المحمولة حيث يكون السعر ظاهراً أيضاً.	Customer, Computer (Standard Configuration product)
2	يختار الزبون معاينة معلومات تفصيلية عن التشكيل، ربما بهدف شرائه كما هو أو بهدف بناء تشكيل يلائمه أكثر، ويمكن حساب سعر أي تشكيل عند طلب الزبون.	Customer, Configured Computer (Configured Product), ConfigurationItem
3	قد يختار الزبون أن يطلب شراء الحاسوب أنياً، أو أن يطلب أن يتصل به مندوب المبيعات ليشرح له بعض التفاصيل أو يفاوضه على السعر قبل اعتماد الطلب.	Customer, Configured Computer Order, Salesperson
4	لاعتماد الطلب يجب أن يملأ الزبون استمارة تتضمن عنوان الشحن وعنوان الفاتورة، مع تفاصيل عن طريقة الدفع (بطاقة ائتمان أو شيك).	Customer, Order, Shipment, Invoice, Payment
5	بعد إدخال طلب الزبون إلى النظام يرسل مندوب المبيعات طلباً بصيغة الكترونية إلى المخزن يتضمن تفاصيل التشكيل المطلوب.	Customer, Order, Salesperson, (Configured Computer), ConfigurationItem
6	يُرسّل للزبون عبر البريد الالكتروني تفاصيل المناقلة بما فيها رقم الطلب ورقم حساب الزبون بحيث يستطيع في أي وقت أن يتحرى عن حالة الطلب.	Order, Customer, OrderStatus
7	يحصل المخزن على الفاتورة من مندوب المبيعات ويشحن الحاسب إلى الزبون.	Invoice, Shipment

- وفي ما يلي الصفوف الأساسية للنظام:



- ومن خلال المتطلبات قم بتحديد الواصفات الأساسية للصفوف السابقة.

المنارة للاستشارات
www.manaraa.com

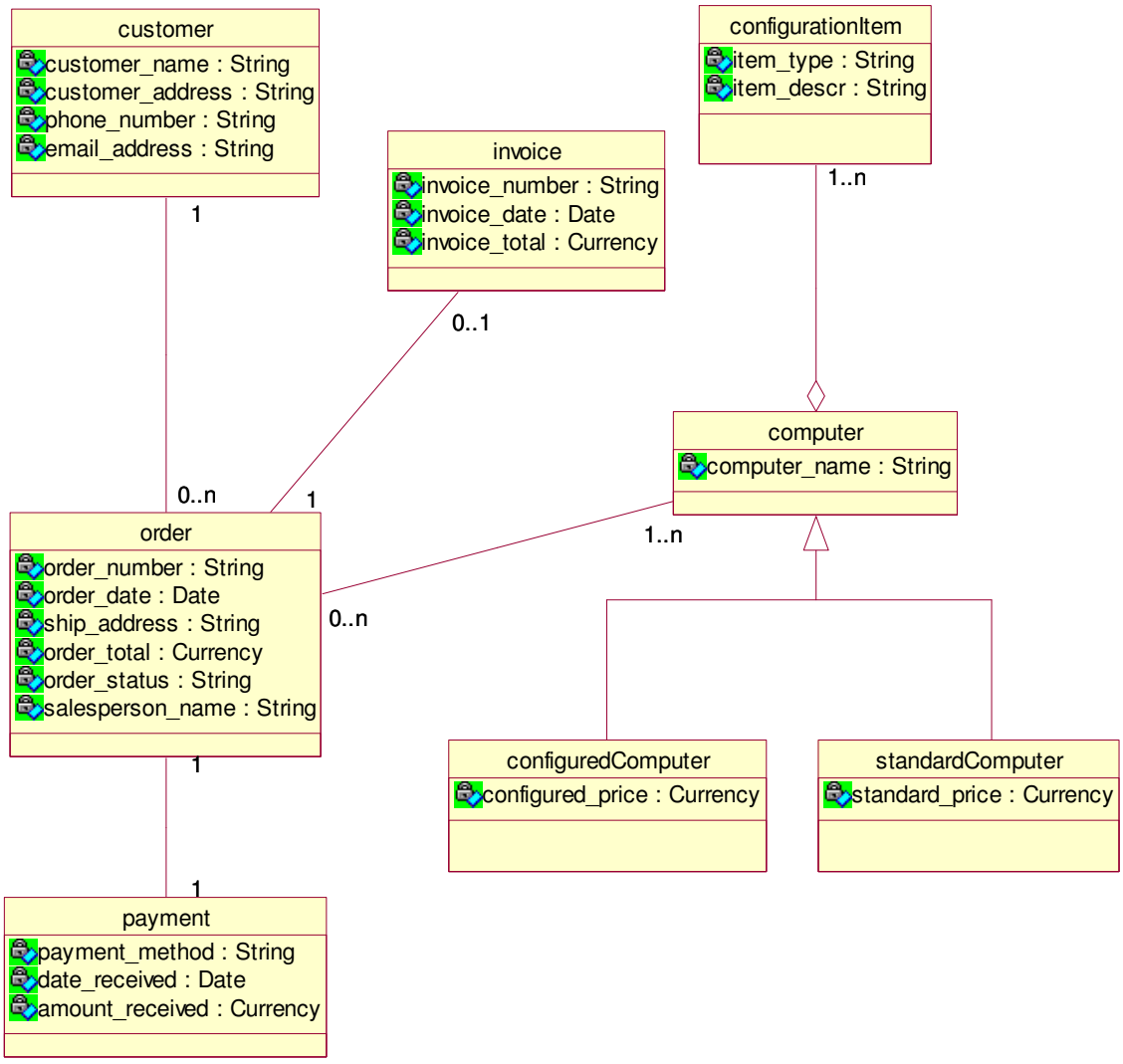


التحليل: الخطوة 7

- بالأخذ بعين الاعتبار الصفوف السابقة، حاول أن تضع مسارات التواصل التي تفرضها حالات الاستخدام بين تلك الصفوف، ثم أضف إلى نموذج الصفوف علاقات الاقتران والتجميع والتعميم المناسبة.
- يبين الشكل التالي علاقات الاقتران الأكثر وضوحاً بين صفوف النموذج، وقد اعتمدنا في تحديد تعداد علاقة الاقتران على بعض الفرضيات، فالطلب Order مثلاً يرد من زبون Customer واحد، بينما يمكن أن يضع الزبون نفسه عدة طلبات، ولا يقبل الطلب Order إلى بعد تحديد طريقة الدفع Payment (لذلك توجد بينهما علاقة اقتران واحد لواحد)، وليس من الضروري أن يقترن بكل طلب فاتورة Invoice، لكن تقترن كل فاتورة دوماً بطلب واحد، كما يمكن أن يتضمن الطلب الواحد أكثر من Configured Computer، ويمكن أن يطلب حاسوب معين Configured Computer عدة مرات وقد لا يطلب إطلاقاً.

  كما نلاحظ علاقتي تجميع، فلكل غرض Computer غرض ConfigurationItem أو أكثر. وبالمثل يتألف كل ConfiguredComputer من ConfigurationItem أو أكثر.
- يبين المخطط التالي، مخطط الصفوف الكامل (حيث يحوي على جميع علاقات الاقتران والتجميع والتعميم، مع تعديل لبعض الواصفات وفق ما تقضيه هرمية التعميم):

المنارة للاستشارات
www.manaraa.com



التحليل: الخطوة 8

- عد إلى مخطط النشاط لحالة الاستخدام "Order Configured Computer" (الذي تم رسمه سابقاً) وارسم مخطط التسلسل للخطوة الأولى منه، أي Display Current Configuration. ومن ثم قم برسم مخطط التسلسل الموافق للمخطط ككل (ولتبسيط المخطط لا تظهر تبادل الرسائل بين الغرضين Computer و ConfigurationItem، كما لا تقم بإظهار أغراض الصفوف الفرعية).
- يبين الشكل التالي مخطط التسلسل للنشاط Display Current Configuration. يقرر الفاعل الخارجي (Customer) أن يعرض تشكيلاً لحاسوب فتُرسل الرسالة Open New إلى الغرض ConfWin الذي ينتمي للصف ConfigurationWindow، فينتج عن ذلك إنشاء الغرض الجديد aConfWin.

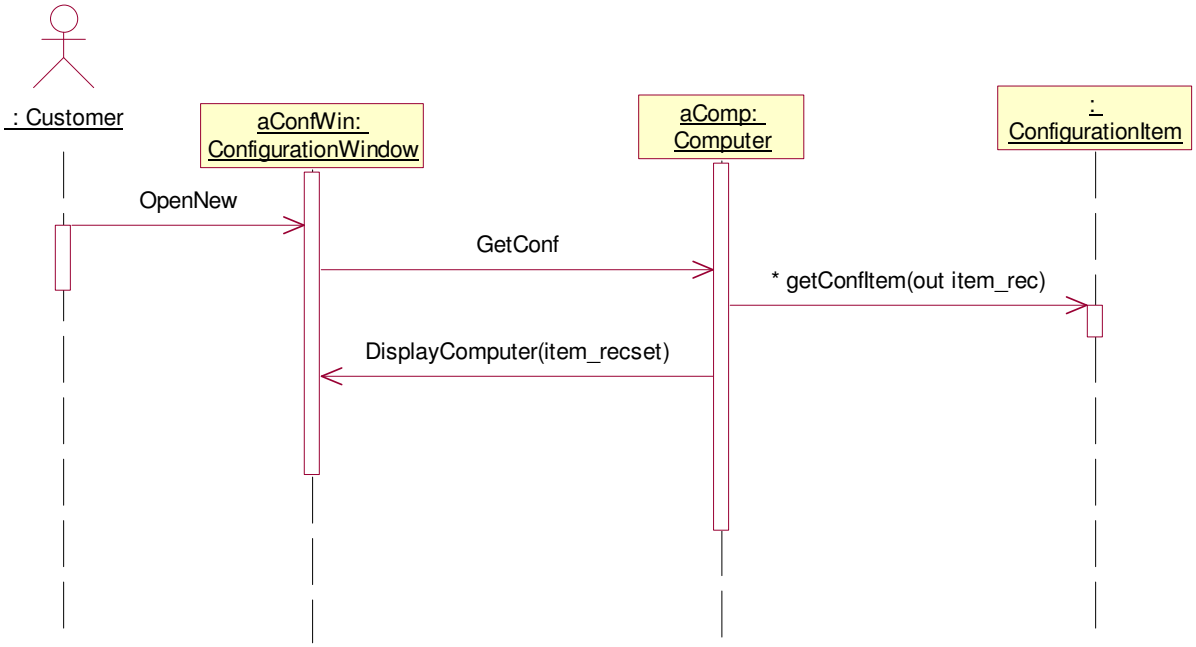
على الغرض aConfWin أن "يعرض نفسه" مع معطيات التشكيل الخاصة به، ولذلك يرسل رسالة إلى الغرض aComp: Computer، وaComp هو غرض من الصف Standard Computer أو من الصف ConfiguredComputer، والصف Computer هو صف مجرد.

يستخدم الغرض aComp وسيط الخرج item- rec ليركب ذاته من أغراض ConfigurationItem، ويرسل عندئذ بنود التشكيل

المنارة للاستشارات
www.manaraa.com

ككتلة واحدة إلى الغرض aConfWin ضمن الوسيط I-recset للرسالة display Computer. بعدئذ يصبح بإمكان الغرض aConfWin أن يعرض نفسه.

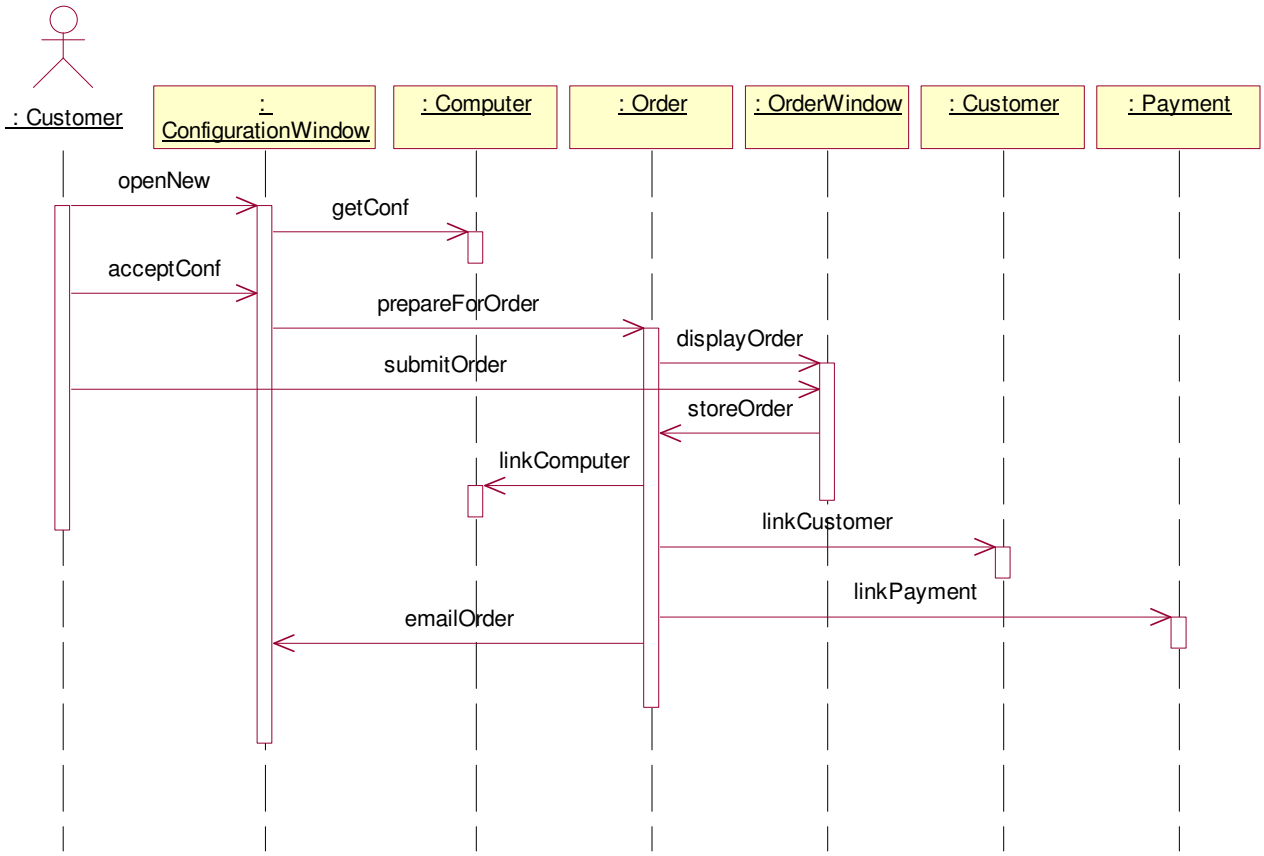
- يبين الشكل التالي مخطط تسلسل النشاط Display Current Configuration:



- يبين المخطط التالي مخطط التسلسل الموافق لمخطط النشاط "Order Configured Computer":

www.manaraa.com

www.manaraa.com



التحليل: الخطوة 9

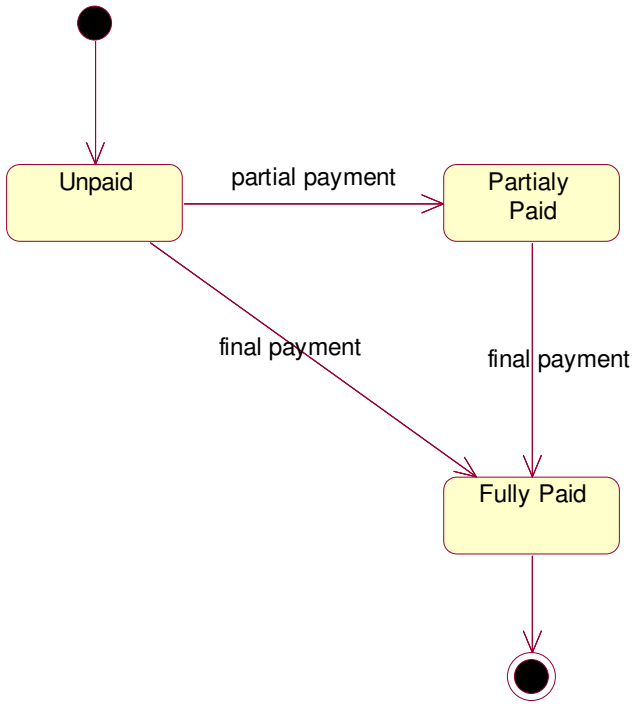
- لنأخذ الصف Invoice. نحن نعلم من نموذج حالات الاستخدام أن الزبون يحدد طريقة دفع ثمن الحاسوب (بشيك أو بطاقة الائتمان) عند ملئه استمارة الشراء وإرسالها إلى البائع.

سيؤدي هذا الأمر إلى توليد طلب شراء وبالتالي إلى تحضير فاتورة. لكن لا يوضح مخطط حالة الاستخدام متى يتم الدفع بالنسبة للفاتورة، ويمكن أن نفترض مثلاً أن الدفع يتم قبل أو بعد إرسال الفاتورة وأن التقسيط مثلاً ممكن.

ونعلم من نموذج الصفوف أن مندوب المبيعات هو الذي يحضر فاتورة طلب الشراء، لكن قد ترسل هذه الفاتورة إلى المخزن الذي يرسلها إلى الزبون مع الحاسوب. ولذلك من الهام أن يعرف النظام حالة الفاتورة.

- ارسم، بناء على ما تقدم، مخطط الحالات الذي يصور حالات الفاتورة الممكنة تبعاً لطرق الدفع.
- يبين الشكل التالي مخطط حالات الصف Invoice والأحداث المؤثرة عليها:

www.manaraa.com



التحليل: الخطوة 10

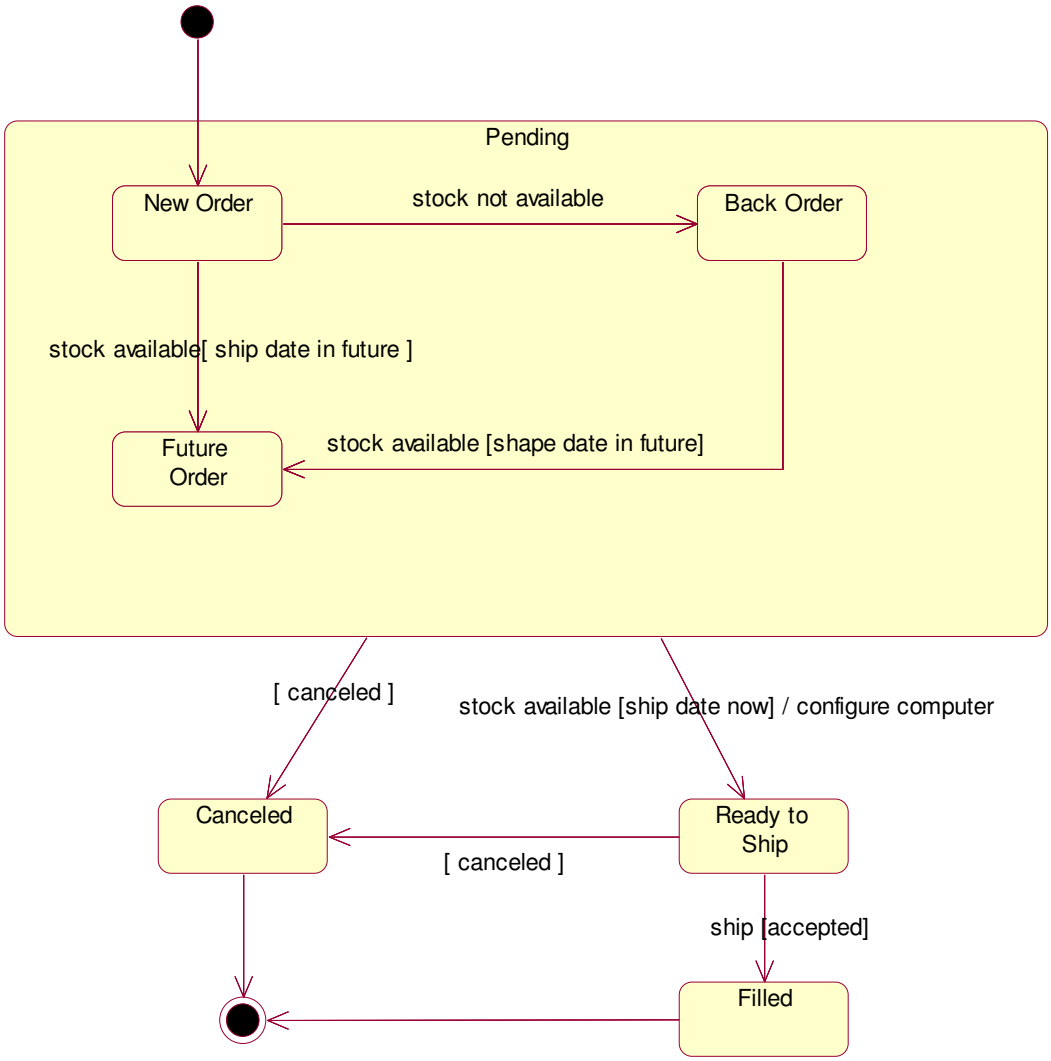
- بالعودة إلى الخطوات السابقة من المثال حاول أن تعدد الحالات التي يمكن أن يتواجد فيها الغرض Order بدءاً من ملئه بالمعلومات وإرساله إلى النظام.

تذكر أن الحاسوب قد يكون موجوداً في المخزن، وقد يحتاج للتجميع لتلبية متطلبات الزبون، كما يمكن للزبون أن يحدد التاريخ الذي يرغب فيه باستلام الحاسوب مع أنه قد يكون موجوداً في المخزن من قبل.

يمكن للزبون أن يلغي طلب الشراء في أي وقت قبل شحنه. ارسم مخطط الحالات للصف Order.

- يبين الشكل التالي مخطط حالات الصف Order والأحداث المؤثرة عليه:

المنارة للاستشارات
www.manaraa.com



www.manaraa.com

القسم الثامن عشر

دراسة واقعية

الكلمات المفتاحية:

الفاعلون، حالات الاستخدام، المتطلبات الوظيفية، مخطط النشاط، الصفوف، مخطط التسلسل، مخطط الحالات، مخطط التعاون.

ملخص:

يُركِّز هذا الفصل على دراسة وتحليل حالات واقعية.

أهداف تعليمية:

يهدف هذا الفصل إلى:

- دراسة وتحليل حالة واقعية (تسجيل الطلاب في مواد دراسية CS4 administration)
- دراسة حالة واقعية (لعبة Tic-Tac-Toe)
- دراسة حالة واقعية (مسألة الفلاسفة الطاعمين Dining Philosophers)

المنارة للاستشارات
www.manaraa.com

تدريب موجّه في نمذجة التحليل

- سنعرض في هذه الجلسة عدة تمارين على النمذجة المرئية في لغة UML، ونهدف بذلك إلى توضيح مخططات UML المختلفة وكيفية انسجام هذه المخططات فيما بينها، إذ يعرض كل مخطط في UML النظام من منظور معين، ولفهم النظام بكليته يجب أن نطور عدة مخططات، من زوايا نظر مختلفة، وأن نكامل بين هذه المخططات
- التمرين الأول: إدارة تسجيل طلاب CS4 administration
- التمرين الثاني: لعبة Tic-Tac-Toe
- التمرين الثالث: مسألة الفلاسفة Dining philosophers

نص التمرين1: CS4 administration

- إدارة تسجيل الطلاب في مواد السنة الرابعة (CS4 administration):

تجتمع اللجنة المسؤولة عن وضع المناهج في نهاية كل سنة دراسية في قسم علوم الكمبيوتر، لتحديد المواد التي يمكن على طالب CS4 أن يأخذها في السنة الرابعة (ملاحظة: المقصود بـــCS4 هو أي طالب يأخذ مادة واحدة على الأقل من مواد السنة الرابعة في قسم علوم الكمبيوتر، بغض النظر إذا كان طالباً في هذا الاختصاص أم لا).

كما يقوم رئيس القسم في نهاية كل عام بإعطاء مهمات إلى الكادر التدريسي في القسم من أجل تدريس المواد المقررة، حيث يمكن أن يسند للمحاضر مهمة تدريس مادة أو أكثر.

يقوم كل محاضر بتحديد المنهاج الذي سيتم إعطاؤه ضمن المادة التي سيقوم بتدريسها ويقوم بتسليم هذا المنهاج إلى مكتب التدريس، ليقوم أحد أعضاء مكتب التدريس بتحويل المنهاج إلى صيغة ورقية (كتاب المادة)، كما يقوم مُنسِّق السنة الرابعة بتوليد صيغة الكترونية من المنهاج.

يقوم مُنسِّق السنة الثالثة بإعطاء قائمة بأسماء الطلاب الذين انتقلوا من السنة الثالثة إلى السنة الرابعة، إلى كل من مُنسِّق السنة الرابعة وإلى مكتب التدريس.

يقوم مُنسِّق السنة الرابعة بإعطاء مكتب التدريس قائمة بالطلاب الذين سجلوا على مواد من السنة الرابعة، على الرغم من أنهم ليسوا من السنة الثالثة.

يقوم الطالب بملء الأوراق اللازمة لعملية التسجيل ويعطيها إلى مكتب التدريس، الذي يقوم بدوره بالتأكد من وجود هذا الطالب ضمن قوائم CS4، وأنه يسجل على عدد مقبول من المواد.

كل عضو من الهيئة التدريسية مسؤول عن تقديم النصيحة لمجموعة من طلاب CS4.

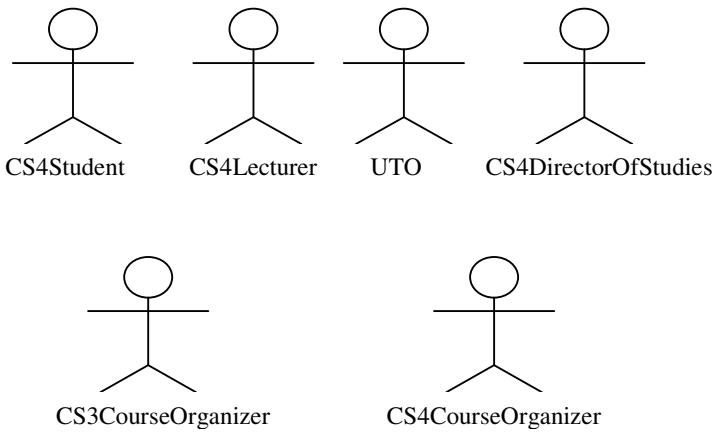
التحليل: الخطوة 1

- عد إلى نص التمرين المذكور سابقاً وحاول تحديد فاعلي النظام.
- نلاحظ من النص ممثلي النظام:
  - طالب السنة الرابعة "CS4Student".
  - المحاضر "CS4Lecturer".
  - مُنسِّق السنة الثالثة "CS3CourseOrganizer".
  - مُنسِّق السنة الرابعة "CS4CourseOrganizer".

المنارة للاستشارات
www.manaraa.com

- مسؤول الهيئة التدريسية "UTO".
- عضو الهيئة التدريسية المسؤول عن تقديم النصح "CS4DirectorOfStudies".

- وفي ما يلي مخطط فاعلي النظام:



التحليل: الخطوة 2

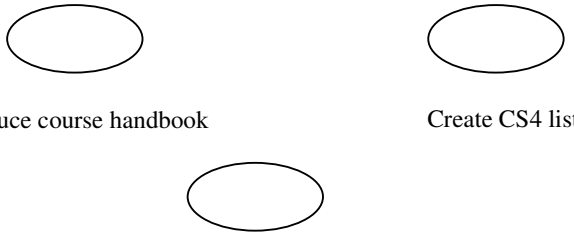
- عُد إلى نص التمرين وحدد حالات الاستخدام للنظام.
- يربط الجدول التالي المتطلبات الوظيفية التي ذكرت في نص التمرين بالفاعلين وبحالات الاستخدام.

الرقم	المتطلب	الفاعل	حالة الاستخدام
1	يقوم كل محاضر بتحديد المنهاج الذي سيتم إعطاؤه ضمن المادة التي سيقوم بتدريسها ويقوم بتسليم هذا المنهاج إلى مكتب التدريس، ليقوم أحد أعضاء مكتب التدريس بتحويل المنهاج إلى صيغة ورقية (كتاب المادة)، كما يقوم مُنسِّق السنة الرابعة بتوليد صيغة الكترونية من المنهاج.	CS4Lecturer, CS4CourseOrganizer, UTO	Produce course handbook
2	يقوم مُنسِّق السنة الثالثة بإعطاء قائمة بأسماء الطلاب الذين انتقلوا من السنة الثالثة إلى السنة الرابعة، إلى كل من مُنسِّق السنة الرابعة وإلى مكتب التدريس. يقوم مُنسِّق السنة الرابعة بإعطاء مكتب التدريس قائمة بالطلاب الذين سجلوا على مواد من السنة الرابعة، على	CS3CourseOrganizer, CS4CourseOrganizer, UTO	Create CS4 list

المنارة للاستشارات
www.manaraa.com

		الرغم من أنهم ليسوا من السنة الثالثة.	
Register for modules	CS4Student, CS4DirectorOfStudies UTO	يقوم الطالب بملء الأوراق اللازمة لعملية التسجيل ويعطيها إلى مكتب التدريس، الذي يقوم بدوره بالتأكد من وجود هذا الطالب ضمن قوائم CS4 ، وأنه يسجل على عدد مقبول من المواد. كل عضو من الهيئة التدريسية مسؤول عن تقديم النصيحة لمجموعة من طلاب CS4.	3

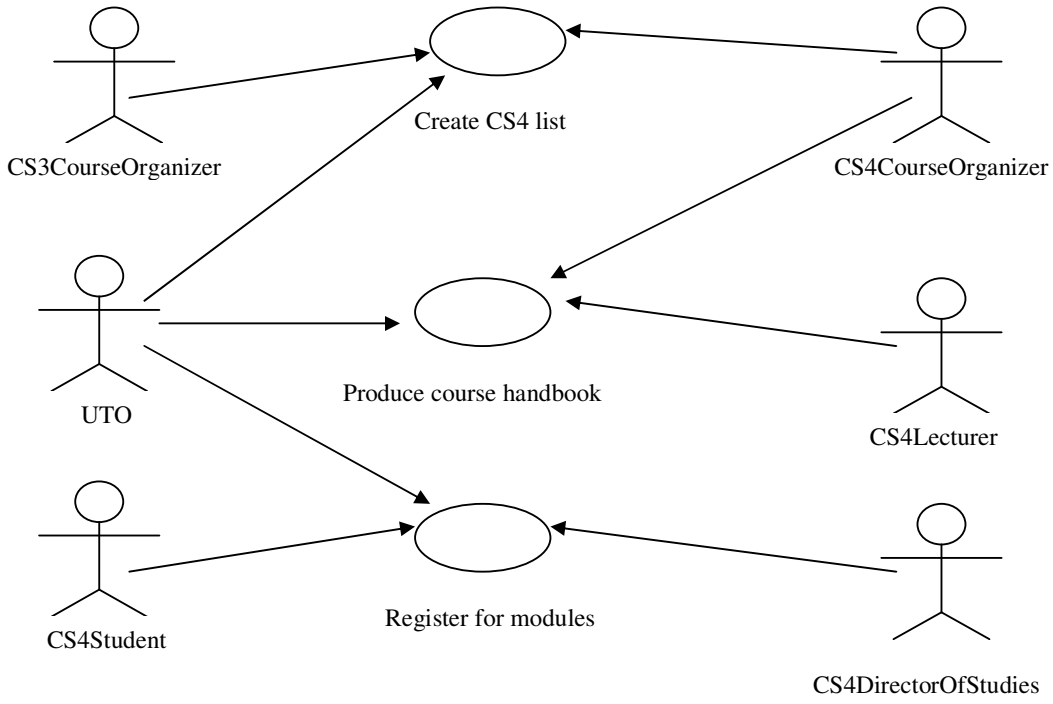
- ويبين الشكل التالي التدوين البياني لحالات الاستخدام:



التحليل: الخطوة 3

- ارسم مخطط حالات الاستخدام بعد دراسة الحالات التي سبق ذكرها.
- يمكن إنجاز هذه الخطوة بالاعتماد مباشرة على معلومات تضمنتها الخطوات السابقة مع الأخذ بعين الاعتبار لأمر إضافي وحيد وهو العلاقات القائمة بين حالات الاستخدام (كما مرّ معنا).
- ويبين الشكل التالي مخطط حالات الاستخدام:

المنارة للاستشارات
www.manaraa.com



التحليل: الخطوة 4

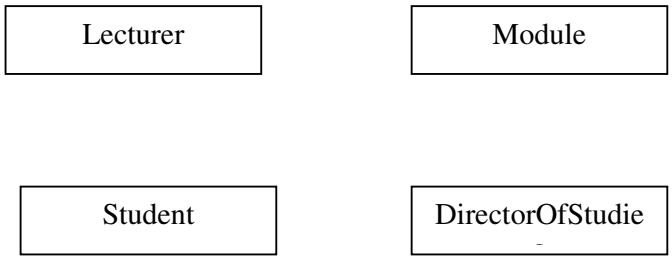
- عد إلى المتطلبات المعرفة في النص، وأوجد الصفوف الخاصة بالنظام.
- يمكننا بإتباع المنهج نفسه الذي اتبعناه في تحديد الفاعلين وحالات الاستخدام، أن ننشئ جدولاً يساعد في تحديد الصفوف من تحليل المتطلبات الوظيفية.
- إسناد المتطلبات إلى صفوف الكيانات:

الرقم	المتطلب	صف الكيان
1	تجتمع اللجنة المسؤولة عن وضع المناهج في نهاية كل سنة دراسية في قسم علوم الكمبيوتر لتحديد المواد التي يمكن على طالب CS4 أن يأخذها في السنة الرابعة.	Module, Student
2	كما يقوم رئيس القسم في نهاية كل عام بإعطاء مهمات إلى الكادر التدريسي في القسم من أجل تدريس المواد المقررة، حيث يمكن أن يسند للمحاضر مهمة تدريس مادة أو أكثر.	Lecturer, Module
3	يقوم مُنسق السنة الثالثة بإعطاء قائمة بأسماء الطلاب الذين انتقلوا من السنة الثالثة إلى السنة الرابعة، إلى كل من مُنسق السنة الرابعة وإلى مكتب التدريس.	Student
4	يقوم كل محاضر بتحديد المنهاج الذي سيتم إعطاؤه ضمن المادة التي سيقوم بتدريسها ويقوم بتسليم هذا المنهاج إلى مكتب التدريس،	Lecturer, Module

المنارة للاستشارات
www.manaraa.com

	ليقوم أحد أعضاء مكتب التدريس بتحويل المنهاج إلى صيغة ورقية (كتاب المادة)، كما يقوم مُنسق السنة الرابعة بتوليد صيغة الكترونية من المنهاج.	
5	كل عضو من الهيئة التدريسية مسؤول عن تقديم النصيحة لمجموعة من طلاب CS4.	Student, DirectorOfStudies

- وفي ما يلي الصفوف الأساسية للنظام:

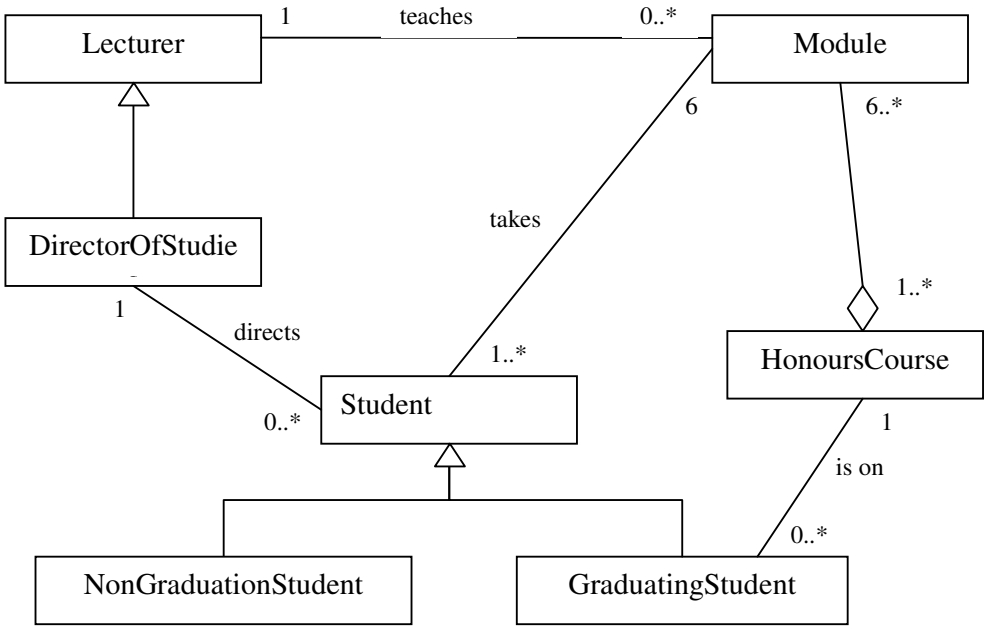


- ومن خلال المتطلبات قم بتحديد الواصفات الأساسية للصفوف السابقة.

التحليل: الخطوة 5

- بالأخذ بعين الاعتبار الصفوف السابقة، حاول أن تضع مسارات التواصل التي تفرضها حالات الاستخدام بين تلك الصفوف، ثم أضف إلى نموذج الصفوف علاقات الاقتران والتجميع والتعميم المناسبة.
- يبين المخطط التالي، مخطط الصفوف الكامل:

www.manaraa.com



نص التمرين2: Tic-Tac-Toe game

- لعبة Tic-Tac-Toe:

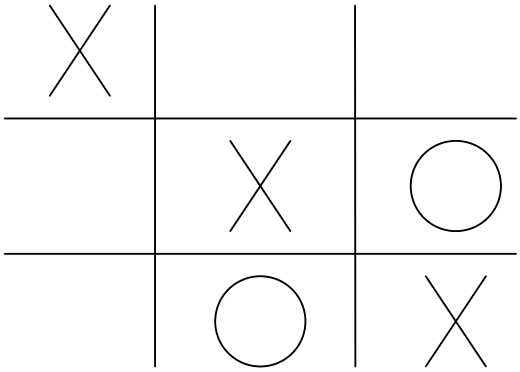
تُلعَب هذه اللعبة من قبل لاعبين على لوح مؤلف من 3×3 مربع.

يختار اللاعب الذي يبدأ باللعب أحد المربعات ليضع فيها إشارة "X"، بينما يختار اللاعب الثاني مربع فارغ ليضع إشارة "O".

يستمر اللعب بالتناوب بين اللاعبين حتى يفوز أحد اللاعبين أو تمتلئ جميع المربعات دون فوز أي منهما.

يفوز اللاعب الذي يحقق صف (أفقي، عامودي، أو قطري) من المربعات التي تحوي رمز اللعب الموافق له ("X" أو "O").

- مثال على لعبة Tic-Tac-Toe:



المنارة للاستشارات
www.manaraa.com

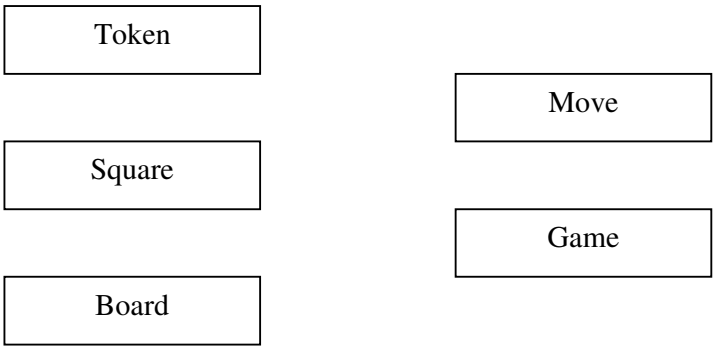
التحليل: الخطوة 1

عد إلى نص تعريف اللعبة وحاول استنتاج الصفوف الخاصة بها.

- الصفوف الخاصة بلعبة Tic-Tac-Toe:

الرقم	الوصف	الصف
1	يحتوي على المعلومات الخاصة باللاعب. يحدد اللاعبين ضمن النظام.	Player
2	يحدد علامة ضمن لوحة اللعب. يحدد موقع العلامة على الرقعة.	Token
3	يحدد التغيير والحركة ضمن دور اللاعب.	Move
4	التأكد من قواعد اللعب. تحديد الفائز. التأكد من صحة الحركات.	Game
5	يمثل رقعة اللعب كاملة.	Board
6	يمثل مربع ضمن رقعة اللعب.	Square

وفي ما يلي الصفوف الأساسية للنظام:

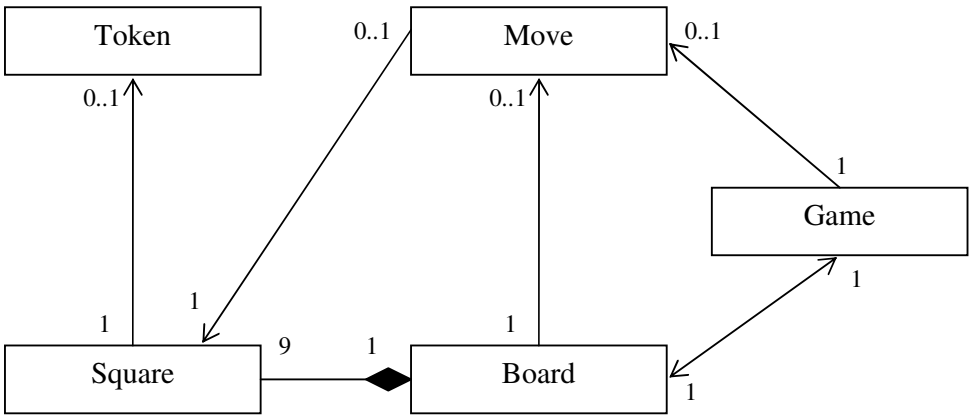


- ومن خلال المتطلبات قم بتحديد الواصفات الأساسية للصفوف السابقة.

التحليل: الخطوة 2

- بالأخذ بعين الاعتبار الصفوف السابقة، حاول أن ترسم مخطط الصفوف الموافق للعبة.
- يبين المخطط التالي، مخطط الصفوف الكامل:

المنارة للاستشارات
www.manaraa.com

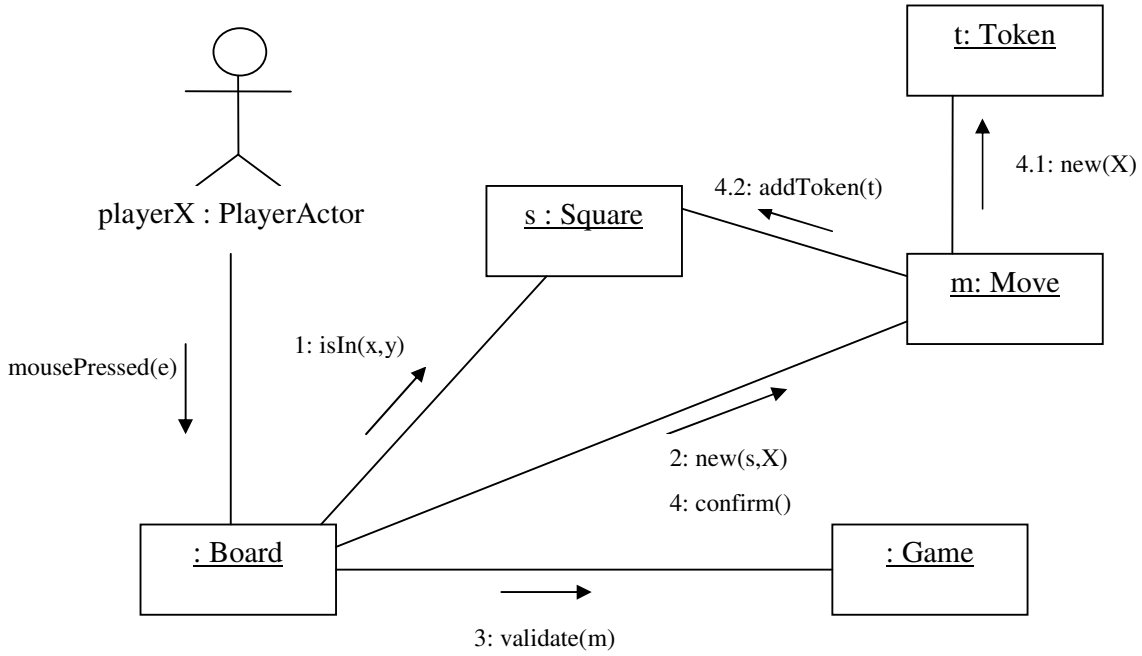


التحليل: الخطوة 3

- ارسم مخطط التعاون لحركة وضع إشارة من قبل لاعب في لعبة Tic-Tac-Toe على الرقعة.
- يحدد اللاعب "Player" الموقع على الرقعة "Board" الذي يريد وضع علامة عليه (وليكن ذلك من خلال النقر بالفأرة).

يقوم "Board" بتحديد المربع المطلوب "Square"، ومن ثم يتأكد من خلال مساعدة الصفوف "Game" و "Move" و "Token" من أن هذا هو فعلاً دور اللاعب، وأن المربع فارغ ويمكن وضع علامة فيه.

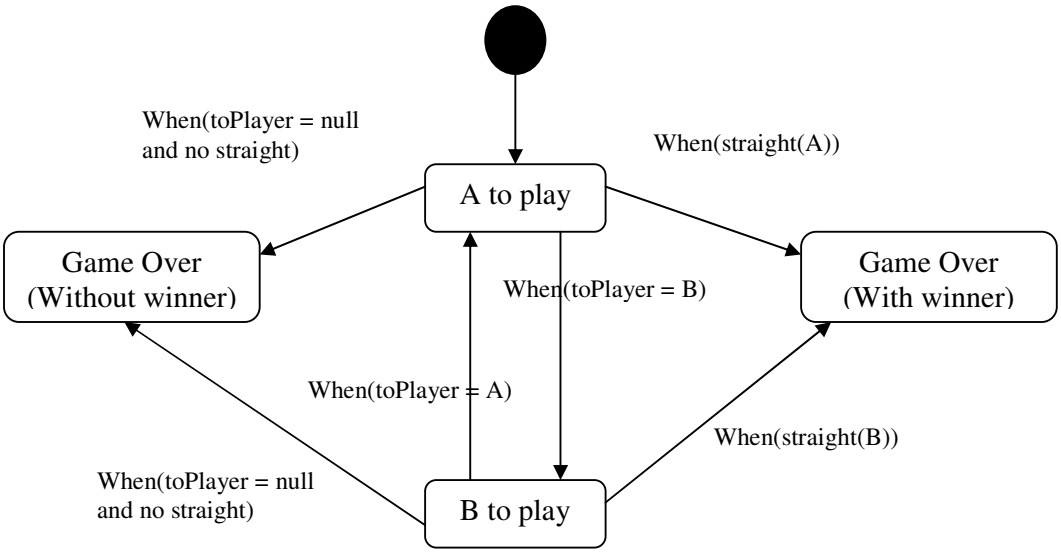
- يبين الشكل التالي مخطط التعاون لحركة وضع علامة ضمن الرقعة:



www.manaraa.com

التحليل: الخطوة 4

- ارسم، مخطط الحالات الذي يصور حالات انتقال اللعب بين اللاعبين والتي تنتهي بفوز أحدهما أو بدون فوز.
- يبين الشكل التالي مخطط حالات الصف Game والأحداث المؤثرة عليه:



نص التمرين3: dining philosophers

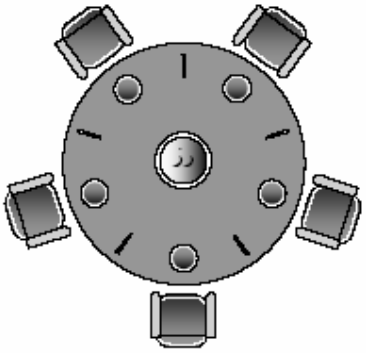
- مسألة الفلاسفة dining philosophers:

لنأخذ خمسة فلاسفة يقضون حياتهم في التفكير وتناول الطعام، حيث يتشارك الفلاسفة في طاولة دائرية يحيط بها خمس كراسي (كرسي لكل فيلسوف)، وفي وسط الطاولة صحن أرز، ويوجد على الطاولة خمسة أعواد.

عندما يفكر الفيلسوف لا يتفاعل مع زملائه، وعندما يشعر بالجوع يحاول التقاط أقرب عودين إليه (العودين على يمينه ويساره)، لا يستطيع الفيلسوف سوى التقاط عود واحد في وقت واحد، كما لا يستطيع أن يلتقط العود الموجود في يد أي من جاريه.

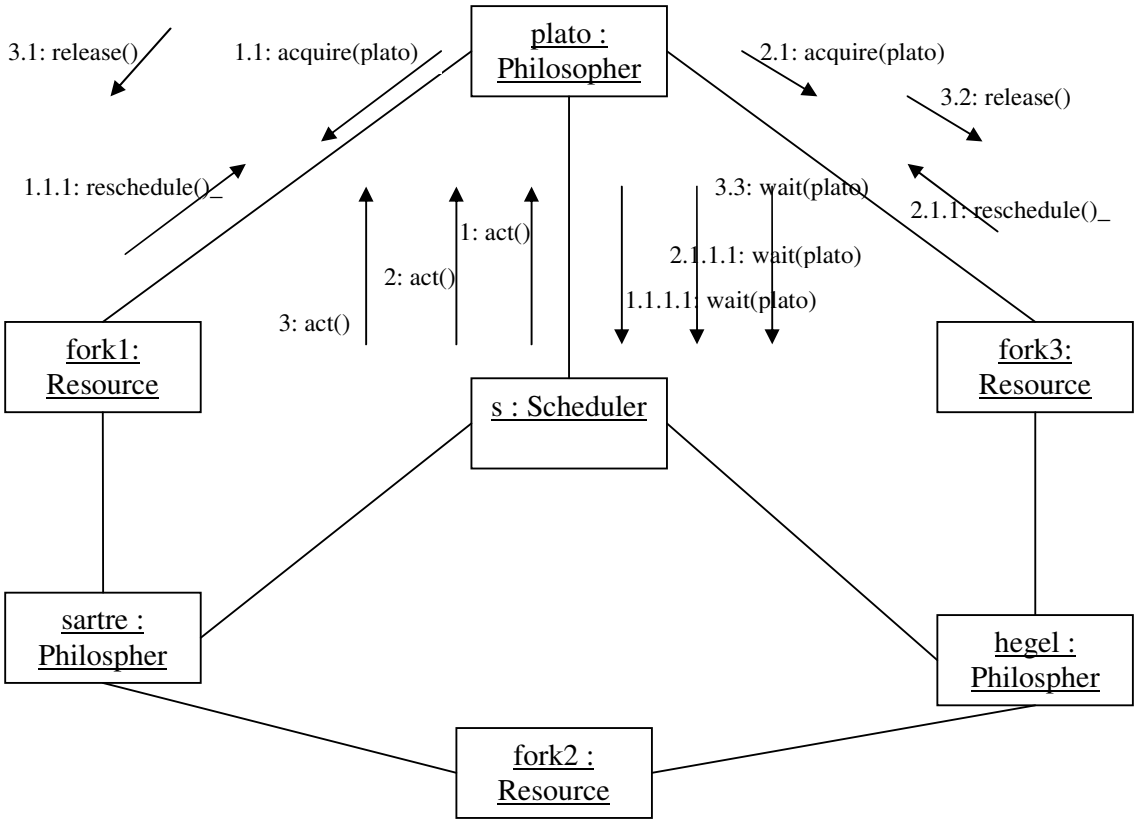
عندما يحصل الفيلسوف على عوديه يستطيع أن يأكل، على أن يتخلى عن عوديه عندما ينتهي.

المنارة للاستشارات
www.manaraa.com



التحليل: الخطوة 1

- ارسم مخطط التعاون لمسألة الفلاسفة.
- يبين الشكل التالي مخطط التعاون لمسألة الفلاسفة:



www.manaraa.com

قراءات اضافية:

- http://www.onjava.com/pub/a/onjava/2005/07/20/businessprocessmodeling.html
- http://www.bpmi.org/
- http://www.bpmn.org/
- Developing Software with UML – Object-oriented analysis and design in practice, Bernd Oestereich, Addison-Wesley Professional; 2 edition (July 15, 2002), ISBN: 020175603X
- Using UML. Software Engineering with Objects and Components (Updated Edition), Perdita Stevens with Rob Pooley, Addison-Wesley ; 1st edition (April 1, 1999), ISBN: 0201360675

المنارة للاستشارات
www.manaraa.com